Algorithms for Intelligent Systems
Series Editors: Jagdish Chand Bansal · Kusum Deep · Atulya K. Nagar

Manjaree Pandit
M. K. Gaur
Prashant Singh Rana
Akhilesh Tiwari *Editors*

Artificial Intelligence and Sustainable Computing

Proceedings of ICSISCET 2021

Springer

Algorithms for Intelligent Systems

This book series publishes research on the analysis and development of algorithms for intelligent systems with their applications to various real world problems. It covers research related to autonomous agents, multi-agent systems, behavioral modeling, reinforcement learning, game theory, mechanism design, machine learning, meta-heuristic search, optimization, planning and scheduling, artificial neural networks, evolutionary computation, swarm intelligence and other algorithms for intelligent systems.

The book series includes recent advancements, modification and applications of the artificial neural networks, evolutionary computation, swarm intelligence, artificial immune systems, fuzzy system, autonomous and multi agent systems, machine learning and other intelligent systems related areas. The material will be beneficial for the graduate students, post-graduate students as well as the researchers who want a broader view of advances in algorithms for intelligent systems. The contents will also be useful to the researchers from other fields who have no knowledge of the power of intelligent systems, e.g. the researchers in the field of bioinformatics, biochemists, mechanical and chemical engineers, economists, musicians and medical practitioners.

The series publishes monographs, edited volumes, advanced textbooks and selected proceedings.

Indexed by zbMATH.

All books published in the series are submitted for consideration in Web of Science.

Manjaree Pandit · M. K. Gaur ·
Prashant Singh Rana · Akhilesh Tiwari
Editors

Artificial Intelligence and Sustainable Computing

Proceedings of ICSISCET 2021

Springer

Editors
Manjaree Pandit
Department of Electrical Engineering
Madhav Institute of Technology
and Science
Gwalior, India

Prashant Singh Rana
Thapar Institute of Engineering
and Technology
Patiala, India

M. K. Gaur
Madhav Institute of Technology
and Science
Gwalior, India

Akhilesh Tiwari
Madhav Institute of Technology
and Science
Gwalior, India

ISSN 2524-7565 ISSN 2524-7573 (electronic)
Algorithms for Intelligent Systems
ISBN 978-981-19-1652-6 ISBN 978-981-19-1653-3 (eBook)
https://doi.org/10.1007/978-981-19-1653-3

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Organizing Committee

Organizing Chairs

Dr. Pratesh Jayaswal, Madhav Institute of Technology and Science, Gwalior, India
Dr. Kusum Kumari Bharti, IIITDM Jabalpur, India

Program Chairs

Dr. Manish Dixit, Madhav Institute of Technology and Science, Gwalior, India
Dr. Harish Sharma, RTU Kota, India

Publicity Chairs

Dr. Manoj Trivedi, Madhav Institute of Technology and Science, Gwalior, India
Dr. Anjali Patil, Madhav Institute of Technology and Science, Gwalior, India
Dr. Mukesh Saraswat, JIIT Noida, India
Dr. Ashish Tripathi, MNIT Jaipur, India

Advisory Committee

Laxmi Srivastava, MITS, Gwalior, India
R.V. Rao, Sardar Vallabhbhai National Institute of Technology, Surat, India
Mukhtiar Singh, Delhi Technological University, New Delhi, India
J. C. Bansal, South Asian University, New Delhi, India
Daizhong Su, Nottingham Trent University Burton Street, Nottingham, UK

Joze Balic, University of Maribor Smetanova, Slovenia
J. Paulo Davim, University of Aveiro, Portugal
Felix T. S. Chan, The Hong Kong Polytechnic University, Hong Kong
Ajith Abraham, Machine Intelligence Research Labs, Washington, USA
P. N. Suganthan, Nanyang Technological University, Singapore
Duc T. Pham, University of Birmingham, Birmingham, UK
I. K. Bhat, Former Vice Chancellor, Manav Rachna University, Delhi, India
S. G. Deshmukh, IIT, Delhi, and Former Director, IIITM, Gwalior, India
Andrew Kusiak, University of Iowa, Iowa City, USA
D. P. Kothari, Former Director I/C, IIT, Delhi, India
Ali Cemal Benim, Duesseldorf University of Applied Sciences, Germany
S. N. Singh, Former Vice Chancellor, MMM University, Gorakhpur, India, and Professor, IIT Kanpur, Kanpur, India
R. A. Gupta, Vice Chancellor, Rajasthan Technical University, Kota, India
A. P. Mittal, Former, Member Secretary, AICTE, New Delhi, India
G. A. Vijayalakshmi Pai, PSG College of Technology, Coimbatore, Tamil Nadu, India
Prof. Aparajita Ojha, PDPM IIITDM, Jabalpur, India
Prof. Shekhar Verma, IIIT, Allahabad, India
Vikas Rastogi, Delhi Technological University (DTU), Delhi, India
C. Patvardhan, Dayalbagh Educational Institute, Dayalbagh, Agra, India
Kapil Sharma, Delhi Technological University (DTU), Delhi, India
D. K. Chaturvedi, Dayalbagh Educational Institute (DEI), Agra, India
Lillie Dewan, NIT, Kurukshetra, India
Sumam Mary Idicula, Cochin University of Science and Technology, Cochin, India
Surender Reddy Salkuti, Woosong University, Republic of Korea, South Korea
Amit Shrivastava, Delhi Technological University (DTU), Delhi, India
A. K. Saxena, Dayalbagh Educational Institute, Dayalbagh, Agra, India
S. C. Choube, UIT, RGPV, Bhopal, India
Shekhar Verma, IIIT, Allahabad, India
Sanjay Agrawal, NITTTR, Bhopal, India
Sandeep Gandi, General Manager (QS), Tata Projects Ltd, India
R. K. Mishra, IIT, BHU, Varanasi, India
K. V. Arya, ABV-IIITM, Gwalior, India
Yogendra Arya, J. C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, India
Rajesh Kumar, Malaviya National Institute of Technology, Jaipur, India
Aditya Trivedi, ABV-IIITM, Gwalior, India
H. M. Suryawanshi, VNIT, Nagpur, India
P. K. Singh, ABV-IIITM, Gwalior, India
Manohar Singh, Central Power Research Institute, Bengaluru, India
M. M. Tripathi, Delhi Technological University (DTU), Delhi

Majid Jameel, Jamia Millia Islamia, New Delhi, India
Surekha Bhanot, BITS, Pilani, India
R. N. Sharma, NIT, Hamirpur, India
Biplab Bhattacharya, IIT-ISM, Dhanbad, India

Technical Program Chairs

S. K. Jain, MITS, Gwalior, India
Hari Mohan Dubey, BIT, Sindri, India
Tripta Thakur, MANIT, Bhopal (Presently Director General National Power Training Institute, Faridabad, India)
Jayshri Vajpai, M B M Engineering College, J N V University, Jodhpur, Rajasthan, India
Amit Singhal, JK Lakshmipat University, Jaipur India
Anmol Ratan Saxena, NIT Delhi, New Delhi, India
Himmat Singh Ahirwar, MITS, Gwalior, India
Adikanda Parida, Regional Institute of Science and Technology, Nirjuli, Arunachal Pradesh, India
Sulochana Wadhwani, MITS, Gwalior, India
D. K. Saini, University of Petroleum and Energy Studies, Dehradun, India
R. Kansal, MITS, Gwalior, India
Arvind Jain, NIT, Agartala, India
R. K. Gupta, MITS, Gwalior, India
V. P. Vishwakarma, GGSIPU, Delhi, India
Urmila Kar, NITTTR, Kolkata, India
Nitin Mallik, The NorthCap University, Gurgaon, India
Taruna Jain, Barkatullah University, Bhopal, India
Laxmi Shrivastav, MITS, Gwalior, India
Pradyumna Chaturvedi, VNIT, Nagpur, India
Shishir Dixit, MITS, Gwalior, India
Vandana Vikas Thakre, MITS, Gwalior, India
Amit Aherwar, MITS, Gwalior, India
C. S. Malvi, MITS, Gwalior, India
Jawar Singh, IIT, Patna, India
Pushpendra Singh, Delhi Technological University (DTU), Delhi
Abhishek Asthana, Sheffield Hallam University, UK
Rolf Crook, University of Leeds, UK
Sudhir Singh Bhadoria, RGPV, Bhopal, India
A. K. Wadhwani, MITS, Gwalior, India
Rajesh Kumar, Delhi Technological University (DTU), Delhi, India
Vinod Katare, General Manager and Superintending Engineer, MPSEB, Madhya Pradesh, India
Kirti Pal Singh, Gautam Buddha University, Noida, India

Trapti Jain, IIT Indore, India.
Y. Kumar, MANIT, Bhopal, India
Sonali Agarwal, IIIT, Allahabad, India
Perminderjit Singh, Punjab Engineering College, Chandigarh, India
Kamal Raj Pardasani, MANIT, Bhopal, India
Arvind Jain, NIT, Agartala, India
Debashis Chatterjee, Jadavpur University, Kolkata, India
Shailaja Kumari, NIT, Warangal, India
P. K. Singhal, MITS, Gwalior, India
Sanjay Tiwari, MITS, Gwalior, India

Session Management Committee

Rajni Ranjan Singh Makwana, MITS, Gwalior, India
Prabhakar Sharma, MITS, Gwalior, India
Atul Chauhan, MITS, Gwalior, India
Mir Shahnawaz Ahmad, MITS, Gwalior, India
Arun Kumar, MITS, Gwalior, India
Rishika Shah, MITS, Gwalior, India
Pushpendra Singh, MITS, Gwalior, India
Aditi Tiwari, MITS, Gwalior, India

Organizing Committee

Vijay Bhuria, MITS, Gwalior, India
Vishal Chaudhary, MITS, Gwalior, India
Saumil Maheshwari, MITS, Gwalior, India
Vikram, MITS, Gwalior, India
Nikhil Paliwal, MITS, Gwalior, India
Poonam Lodhi, MITS, Gwalior, India
Himanshu Mittal, JIIT, Noida, India
Raju Pal, JIIT, Noida, India
Avinash Pandey, JIIT, Noida, India

Preface

This book contains outstanding research papers as the proceedings of the 3rd International Conference on Sustainable and Innovative Solutions for Current Challenges in Engineering and Technology (ICSISCET 2021). ICSISCET 2021 has been organized by Madhav Institute of Technology and Science, Gwalior, India, and technically sponsored by Soft Computing Research Society, India. It was held on November 13–14, 2021, at Madhav Institute of Technology and Science, Gwalior, in virtual mode due to the COVID-19 pandemic. The conference was conceived as a platform for disseminating and exchanging ideas, concepts, and results of the researchers from academia and industry to develop a comprehensive understanding of the challenges of the advancements of sustainable and innovative solutions for current challenges in engineering and technology viewpoints. This book will help in strengthening congenial networking between academia and industry. The conference focused on collective intelligence, sustainable computing and information technology, computational intelligence and machine learning, embedded systems and VLSI design.

We have tried our best to enrich the quality of the ICSISCET 2021 through a stringent and careful peer-review process. ICSISCET 2021 received a significant number of technical contributed articles from distinguished participants from home and abroad. ICSISCET 2021 received 267 research submissions from 40 different countries, viz. Algeria, Austria, Bangladesh, Canada, China, Egypt, Ethiopia, Finland, Germany, Ghana, Greece, India, Iran, Italy, Kenya, Liberia, Malaysia, Mexico, Morocco, Nepal, New Caledonia, Oman, Peru, Poland, Portugal, Romania, Russia, Saudi Arabia, Senegal, Serbia, Slovakia, Spain, Sri Lanka, Turkey, Ukraine, United Arab Emirates, UK, USA, Vietnam and Yemen. After a very stringent peer-reviewing process, only 55 high-quality papers were finally accepted for presentation and the final proceedings.

In fact, this book presents novel contributions in areas of computational intelligence, and it serves as a reference material for advanced research.

Gwalior, India — Manjaree Pandit
Gwalior, India — M. K. Gaur
Patiala, India — Prashant Singh Rana
Gwalior, India — Akhilesh Tiwari

Acknowledgements

The organizing chairs of the **3rd International Conference on "Sustainable and Innovative Solutions for Current Challenges in Engineering and Technology**" gratefully acknowledge technical support from **Soft Computing Research Society, New Delhi, India**.

The **conference Chief Patron Prof. K. K. Aggarwal, Chairman National Board of Accreditation**, New Delhi, India, and **Patron Prof. R. K. Pandit, Director, Madhav Institute of Technology and Science, Gwalior, India**, were the guiding force behind the conference.

Prof. K. K. Aggarwal also kindly accepted to deliver the keynote address, for which the organizers are eternally indebted.

Prof. Xin-She Yang, Middlesex University, The Burroughs, Hendon, London, and **Dr. Swagatam Das**, ISI, Kolkata, India, deserve a special mention for sparing their valuable time for delivering the keynote talks.

Dr. Jagdish Chandra Bansal, Series Editor, Algorithms for Intelligent Systems (AIS); Mr. Aninda Bose, Executive Editor; and Mr. Maniarasan, Project Coordinator, Springer Nature, deserve special thanks for helping the organizers in publishing the proceedings.

The support from **Dr. Jagdish Chandra Bansal, Dr. Mukesh Saraswat** and **Dr. Harish Sharma**, from the **Soft Computing Research Society, New Delhi**, right from paper review stage to the smooth conduction of the conference on November 13 and 14, 2021, is gratefully acknowledged. Thanks are also due to all the **esteemed reviewers** for their time and significant contribution in maintaining the quality of the papers.

The contribution of **internal and external experts** as **session chairs** and the **session support teams** during the two days was crucial for the effective conduction of the conference. They played a key role in conference conduction by giving valuable comments for improving the quality of the paper and by ensuring that all reviewer comments were incorporated into the revised papers, before publication. The organizers are grateful for their support and efforts in conference conduction.

During the COVID times, organizing an international conference in **virtual mode** was quite challenging. The hard work and efforts of the **Conference Core Team** are

sincerely acknowledged. The members of the **Conference Core Team** have worked relentlessly and have left no stone unturned to make the e-conference a reality.

Thanks are also due to media persons, guests, authors and all those who have directly or indirectly contributed in organizing and conducting this conference.

Organizing Team—ICSISCET 2021

Contents

About the Editors

Manjaree Pandit obtained her M.Tech. degree in Electrical Engineering from Maulana Azad College of Technology, Bhopal, (India), in 1989, and Ph.D. degree from Jiwaji University Gwalior (India) in 2001. She is currently working as Dean Academics and Professor in the Department of Electrical Engineering, M.I.T.S., Gwalior, (India). She is a senior member of IEEE, a recognized reviewer of a few IEEE Transactions, Springer and Elsevier journals, and has published more than 65 papers in reputed international journals. She has been recognized as an outstanding reviewer by Elsevier and received Top Peer Reviewer Award from the international reviewer database site Publons (Clarivate Analytics) for being in top 1% of reviewers in Engineering for year 2017–19 and year 2018–2019 and Top Peer Reviewer Award by Publons for being in top 1% of reviewers in cross field in year 2018–19. Her areas of interest are hybrid renewable energy resources integration with the power grid, nature-inspired algorithms, ANN and fuzzy neural network applications to electrical power system. She has successfully completed research projects funded from AICTE, DST and UGC and has published 68 papers in International Journals of repute, guided more than 70 PG dissertations and 05 Ph.D. candidates. Her papers are very well cited in Google Scholar, Scopus, and Web of Science. She has also received ISTE National Research Award for the years 2002 and 2004, Certificate of Merit from Institution of Engineers, India, UGC Research Award for carrying out postdoctoral work during 2009–2011. She has been involved in conducting a number of national conferences, workshops, and short-term courses.

M. K. Gaur is working as Professor and Head in Mechanical Engineering Department, Associate Proctor and In-charge of the Institute Industry Interaction Cell at Madhav Institute of Technology and Science, Gwalior, India. He has made substantial contributions to the fields of solar thermal energy, solar distillation, and heat and mass transfer applications. He possesses eighteen years teaching experience of postgraduate and undergraduate classes. He has authored three books titled *Solar Energy: Problems Solutions and Experiments*, *Optimization of Machining Parameters of Hard Porcelain on CNC Machine*, and *Solar Energy Systems: Thermal Analysis and Its Application.* Eighty of his papers have been published in various International and

National journals and conferences. He is an active reviewer of prestigious journal like *Solar Energy, Desalination, Desalination and Water Treatment*, *Journal of Thermal Engineering*, etc. His broad areas of interest include thermodynamics, heat transfer, solar energy and solar distillation. He is Life Member of the Indian Society for Technical Education (India) IET and Faculty Advisor of Society of Automotive Engineers INDIA MITS chapter. He attended one-week summer school 'UKERC' at University of Sussex, Brighton, UK, in July 2009. He had obtained Ph.D. from Indian Institute of Technology Delhi, New Delhi, and Master of Technology in Engineering Materials from Maulana Azad National Institute of Technology Bhopal in 2002. He secured fifth position in the university in his Bachelor of Engineering in Mechanical Branch from Oriental Institute of Technology and Science, Bhopal, in 1999.

Dr. Prashant Singh Rana is presently working as Associate Professor in the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab. He received his both Ph.D. and M.Tech. from ABV-IIITM, Gwalior. His areas of research are machine learning, deep learning, bioinformatics, and optimization. He has published more than 70 research papers in different journals and conferences. He completed five projects sponsored by DST, ICMR, NVIDIA, and one project is going on. He published 10 patents. He guided seven Ph.D. students and 18 masters students.

Dr. Akhilesh Tiwari has received Ph.D. degree in Information Technology from Rajiv Gandhi Technological University, Bhopal, India. He is currently working as Professor and Head in the Department of Information Technology, Madhav Institute of Technology and Science (MITS), Gwalior, (Madhya Pradesh), India. Previously, he has also worked as Head of CSE and IT department from March 2018 to December 2020, at MITS Gwalior. His areas of current research include knowledge discovery in databases and data mining, terrorist network mining, hesitation mining, and pattern warehousing. He has published more than 70 research papers in the journals and conferences of international repute, which include publications in international publishing houses such as IGI Global USA, Springer, Inderscience, Taylor & Francis (CRC Press), and Elsevier with SCI, SCIE, and SCOPUS indexing. He has published 10 book chapters in edited books and also edited one book titled *Soft-Computing-Based Nonlinear Control Systems Design* for IGI Global USA. He has guided 04 Ph.D. scholars from various universities in India and more than 20 dissertations at master's level, and at present, he is guiding 08 Ph.D. scholars. Currently, he is Chairman of Board of Studies in Information Technology, contributed as Chairman of Research Advisory Committees (RACs), working as Member Secretary in the Academic Council, Member in the IQAC Cell, Member in the Academic Development Cell at Madhav Institute of Technology and Science, Gwalior (Madhya Pradesh), India, and Member in the Board of Studies of Computer Science and Engineering at Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (Madhya Pradesh). He is also acting as a reviewer and member in editorial board/program committee of various international journals and conferences. He has been honored with the "Outstanding Young Person of Gwalior Award" in 2014

by the JCI Gwalior. He has been honored with the award of recognition for significant contribution in the "Implementation and preparation of structure of flexible curriculum based on AICTE model curriculum" at MITS Gwalior. He is having the memberships of various academic/scientific societies including IETE, GAMS (Gwalior Academy of Mathematical Sciences), IACSIT, and IAENG. He is also in the expert panel/selection committees of various agencies/universities/institutions of India, including National Institute of Electronics and Information Technology (NIELIT), Ministry of Electronics and Information Technology, Government of India.

Chapter 1
Demand-Based Land Suitability Prediction Model for Sustainable Agriculture

G. Sumaiya Farzana and N. Prakash

1 Introduction

India is blessed with the 10th largest fertile land in the world. It is the only country that has 60 soil varieties and among 20 agroclimatic regions, 15 major Agro-climatic conditions exist in India. Nearly 80–90% of the Indian population relies on agriculture. The agricultural industry currently contributes about 15.96% of India's GDP, and half of the workforce in India depends on agriculture for its source of income. According to the Global Hungry food index report, India ranks 103 out of 119, the main reason for this discrepancy is agricultural food wastage. United Nations Food and Agriculture Organization (FAO) states that more than 40% of food produced is wasted in India. Consequently, near to 194 million people go hungry all over the country every single day, that is, nearly one in six people. Total food grain produced in India during 2015–16 was projected at 252.23 million tons, which is five times higher than the year 1950–51 [1]. Yet, abundant food production is not sufficient to confirm food security. Not the whole volume of food produced is consumed, a large amount of food is wasted. The survey estimates that every year in India, about 67 million tons of food is wasted, which has been estimated to be around Rs 92,000 crores [2, 3]. In the case of wheat and paddy, the losses range from 4.93% to 5.53%, and in the case of pulses, the losses range from 6.36 to 8.41 [4]. As expected, the perishable crops fruits and vegetables suffered much higher losses which range from 4.58 to 15.88. In 2016, Central Statistical Organization states that the production of horticulture crops in the year 2012–13 was 284,000 crores whereas the estimation of losses was about 31,500 crores. Apart from food waste, the resources used to propagate seeds and grow food also go to waste. 25% of water goes to waste, which is used to produce

G. Sumaiya Farzana (✉) · N. Prakash
B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India
e-mail: sumaiyaharis89@gmail.com

N. Prakash
e-mail: prakash@crescent.education

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_1

food, while millions of Indians do not have access to fresh drinking water, leading to further healthcare crisis. Food waste occurs due to the three following reasons. First is the inappropriate land usage, that is without considering the demand using up of the excess land area for production. Second, it is difficult for perishable products to survive in long-haul transport. The third is the lack of cold storage infrastructure, that is in India rural farmers have less access to cold storage infrastructure. Moreover, agriculture food waste may also cause pollution, as the waste emits as much carbon pollution that is equivalent to 39 million vehicles. It also emits 3.3 billion tons of greenhouse gas which plays a huge impact on climate change.

Hence, it is essential to control the agricultural food waste and to minimize the resources used for cultivation. In this paper, a land suitability prediction model is developed using linear regression techniques. As the datasets area and production depend on each other, linear regression technique is used [5]. The linear regression technique is the commonly used technique for capturing linear patterns in data [6]. The developed land suitability prediction model will help farmers in selecting the suitable land area for cultivation according to demand. It will also minimize the resources used for cultivation and raise the income of farmers.

The following is a summary of the paper. The work done in this area is described in Sect. 2, and the methodology is described in Sect. 3. The system architecture is explained in Sect. 4 of the paper. The results and analyses are presented in Sects. 5 and 6 concludes with the conclusion.

2 Literature Survey

Purnamasari et al. [7] developed a land suitability model to identify the suitable areas for cassava production. The authors have developed a yield prediction model based on appropriate areas that measure with land suitability analysis. From LSA analysis, the priority indicators were identified using the fuzzy expert system with multicriteria decision method and also ecological categories. The study revealed that 42.17% of the land is suitable for cassava production. RuhollahTaghizadeh-Mehrjardi et al. [8] aimed to measure land suitability for two main crops, namely wheat and barley. The authors have calculated land suitability classes for the two crops, and they mapped with traditional approaches and machine learning. The authors have proved that ML-based land suitability maps provide more accuracy when compared to the traditional approach. Mokarram et al. [9] developed a land suitability classification using an ensemble classifier generation technique named as RotBoost, which is a combination of Rotation Forest and AdaBoost. The authors have proved that the developed land suitability classification model provides more accuracy than Rotation Forest and AdaBoost.

Senagi [10] developed a land suitability model for crop production using a parallel random forest classifier. The authors have predicted the land suitability of crop production from soil property information without the interference of a soil scientist expert. Bagherzadeh et al. [11] developed a land suitability model by integrating

parametric learning neural networks and fuzzy models for soybean production. The authors have made their study with 41 land units and revealed that less production of soybean cultivation is because of the fertility properties in soil.

Parthiban [12] compared naive Bayes and j48 algorithms in predicting the land suitability of crops and recommending fertilizers. The authors have proved that j48 is better than naive Bayes in predicting the land suitability of crops. Sarmadian et al. [13] make use of a Support Vector Machine to evaluate land suitability analysis of rainfed wheat. The value of numerous soil and environment indicators is collected to find out the land suitability index. The authors have mentioned that their land suitability model is a suitable alternative to land suitability scenarios.

Elsheikh [14] developed an intelligent system named Agriculture Land Suitability Elevation for measuring the land suitability of different crops in subtropical and tropical regions. The authors ensured that the developed tool was beneficial for decision-makers in determining the land quality. Komolafe et al. [15] used the j48 algorithm in a decision tree to classify the land suitable or not suitable for cultivating cassava. The developed land suitability classification model provides an accuracy of 87.5% for correctly classified instances and 12.5% accuracy for incorrectly classified instances. The authors have stated that their model has a high potential in selecting suitable land for cultivating cassava.

Researchers have done a major contribution in classifying the land suitability of crops. Most of the researcher's work focused on classifying the land suitability of crops. However, none of them have worked on predicting the land suitability of crops by considering the demand. In this paper, a land suitability prediction model is developed using linear regression techniques to predict the land suitability of the crop according to the demand.

3 Regression Models

The regression model is one of the most widely used statistical models among the supervised learning algorithm for identifying linear patterns in data. The key objective of the regression model is to forecast output or predictor variable that is continuous values, based on the given data [16, 17]. Regression technique is also used for discovering the relationship between the variable which have causal relation [18]. The regression technique is classified into two based on the usage of variables, namely single linear regression and multiple linear regression.

3.1 Single Linear Regression

It is a simple linear regression model, in which prediction is performed based on one predictor and one explanatory variable [17]. It can be expressed in (1)

$$P = I + \beta * X + E \tag{1}$$

where:

P is the predictor variable
'I' is the intercept
'X' is the explanatory variable
β is the slope.

3.2 Multiple Linear Regression

In multiple linear regression, predictions are performed based on two or more explanatory variables and one predictor variable [19]. It can be written in (2)

$$P_i = I + \beta_1 * X_{i,1} + \beta_2 * X_{i,2} + \beta_n * X_{i,n} + E_i \tag{2}$$

where I is the intercept.
$\beta_1, \beta_2, \ldots. \beta_n$ are the slope coefficients of the regressors.
X_i $(i = 1, 2 \ldots n)$ are regressors.
E_i is the error term.

To estimate the P values, the slope of the line of equation and intercept values need to be calculated.

The slope coefficient of the equation can be calculated as

$$\text{Slope}(\beta) = n\left(\sum XP\right) - \left(\sum X\right)\left(\sum P\right) / n\left(\sum X^2\right) - \left(\sum X\right)^2 \tag{3}$$

The intercept of the line can be calculated as

$$\text{Intercept}(S) = \left(\sum P\right)\left(\sum X^2\right) - \left(\sum X\right)\left(\sum XP\right) / n\left(\sum X^2\right) - \left(\sum X\right)^2 \tag{4}$$

where n represents the total number of regressors used in this model.

4 System Architecture

The datasets used for predicting the land suitability of crops were acquired from national and horticultural crops, Tamilnadu. As onion is one of the most commonly used vegetables in Tamilnadu, onion is taken as an example to build the forecasting model. The Onion dataset contains 200 rows with two columns, namely area and

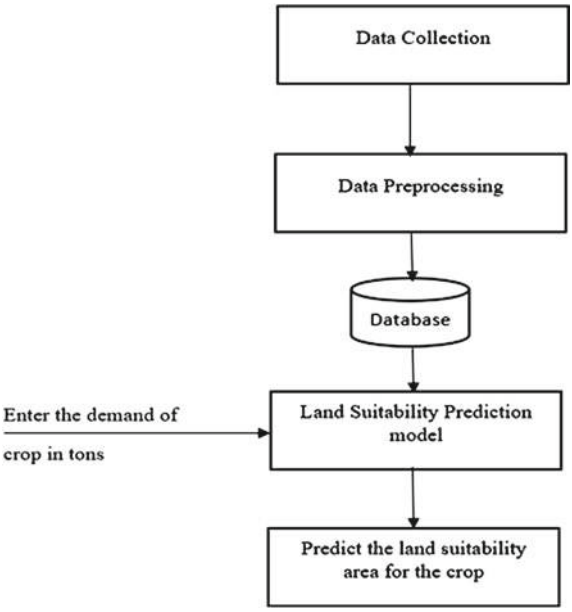


Fig. 1 System architecture for land suitability prediction

production. The area is considered as a predictor variable, and production is considered as an explanatory variable. Area values range from 1 to 1927 ha while production values range from a minimum of 9 tons to a maximum of 21,411 tons. Data is preprocessed to remove the most repeated values, missed values, and the preprocessed data stored in a database. The preprocessed data is applied to the developed land suitability prediction model to predict the land suitability area of the crop shown in Fig. 1. By making use of this land suitability prediction model effectively, farmers can cultivate the crops in the required land area, and it also minimizes the waste in cultivation.

Steps to calculate the land suitability area given in the following equations.

Land suitability prediction model can be written as (5)

$$\hat{A}_i = I + \beta * X_i + E \tag{5}$$

where

A_i denotes the predictor variable (area)
I is the intercept
X denotes the explanatory variable (production)
β is the regression coefficient
E_i is the error term.

The intercept and regression coefficient of land suitability prediction can be calculated by minimizing the sum of squares given in (6).

$$\text{SSE} = \left(A_i - \widehat{A_i}\right)^2 \tag{6}$$

Substituting A_i in (6) to get (7)

$$\text{RSS} = \sum_{n}^{i=1} \left(A_i - (I + \beta X_i)^2\right) \tag{7}$$

Taking the partial derivative with respect to intercept I is given in (8).

$$\begin{aligned} \frac{\partial \text{RSS}}{\partial I} &= \sum_{n}^{i=1} 2(A_i - I - \beta X_i)(-1) \\ &= -2 \sum_{n}^{i=1} (A_i - I - \beta X_i) = 0 \end{aligned} \tag{8}$$

It can also be represented as (9).

$$\sum_{n}^{i=1} e_i = 0 \tag{9}$$

The partial derivative with respect to slope coefficient β is given in (10)

$$\frac{\partial \text{RSS}}{\partial \beta} = \sum_{n}^{i=1} 2(A_i - I - \beta X_i)(-X_i) \tag{10}$$

Differentiating the equation with respect to β to get (11).

$$-2 \sum_{n}^{i=1} X_i (A_i - I - \beta X_i) = 0$$

$$\text{i.e} \sum_{n}^{i=1} X e_i = 0 \tag{11}$$

Expanding (9) and (11) to get (12) and (13)

$$\sum_{i=1}^{n} A_i = nI + \sum_{i=1}^{n} X_i \beta \tag{12}$$

$$\sum_{n}^{i=1} X_i A_i = \sum_{i=1}^{n} X_i I + \sum_{n}^{i=1} X_i^2 \beta \tag{13}$$

To find the intercept dividing (12) by n to get (14).

$$\overline{A} = I + \beta \overline{X} \tag{14}$$

Solving for intercept I give

$$I = \overline{A} - \beta \overline{X}$$

To find the regression coefficient β, multiply (12) by $\sum X_i$ to get (15) and (13) by n to get (16)

$$\sum X_i \sum A_i = n \sum X_i I + \left(\sum X_i\right)^2 \beta \tag{15}$$

$$n \sum X_i A_i = n \sum X_i I + n \sum X_i^2 \beta \tag{16}$$

Subtracting (15) from (16) to get (17)

$$\begin{aligned} n \sum X_i A_i - \sum X_i \sum A_i &= n \sum X_i^2 \beta - \left(\sum X_i\right)^2 \beta \\ &= \beta \left(n \sum X_i^2 - \left(\sum X_i\right)^2 \right) \end{aligned} \tag{17}$$

Solving for regression coefficient β gives

$$\beta = \frac{n \sum X_i A_i - \sum X_i \sum A_i}{n \sum X_i^2 - \left(\sum X_i\right)^2}$$

Hence, by using intercept (I) and regressor coefficient (β) suitable land area (A) can be predicted.

5 Results and Analysis

The proposed land suitability prediction model is implemented in the Spyder environment in python version 3.7. To build an effective forecasting model using linear regression, it is essential to check whether the dependent and independent variables are linearly related. The relationship between the area (predictor variable) and production (explanatory variable) are analyzed through the seaborn function in python, and results are plotted using scatterplot in a graph shown in Fig. 2. From the visual representation of patterns in the data, it is observed that there exists a strong linear relationship between the variables.

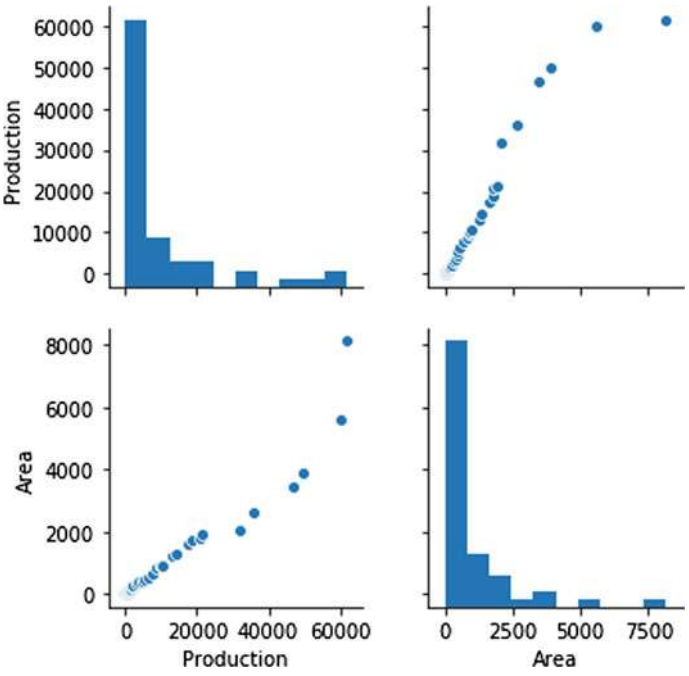


Fig. 2 Scatterplot relationship between the explanatory and predictor variable for the crop onion

Table 1 Correlation analysis

Variable	Production	Area
Production	1.00000	0.991634
Area	0.991634	1.00000

Table 1 shows the correlation between the area and production. The result of the correlation value of 99% indicates that the explanatory variable exhibits a strong relationship with the predictor variable. Figure 3 represents the regression fitting line of the land suitability prediction model. The regression line represents a good fit, and it is proving that line fitted with less error. Hence, by making use of this regression model, the suitable land area can be forecasted with good accuracy. Before building the land suitability prediction model, the efficiency of the model could be evaluated with the sample datasets of the onion by considering the area as a predictor variable and production as the explanatory variable. The land suitability prediction model is evaluated using the R-squared values, standard error, and p-value as the metrics. On analyzing the table, p values of 0.02 specify that the variables are highly significant at a level of 0.05 and provide an R-squared error of 0.996 which implies that the model is 99.6% accurate in predicting the suitable land area shown in Table 2.

Table 3 shows the comparison of the actual land area and predicted land area for the crop onion. From Table 3, it is observed that the actual area is relatively nearer to

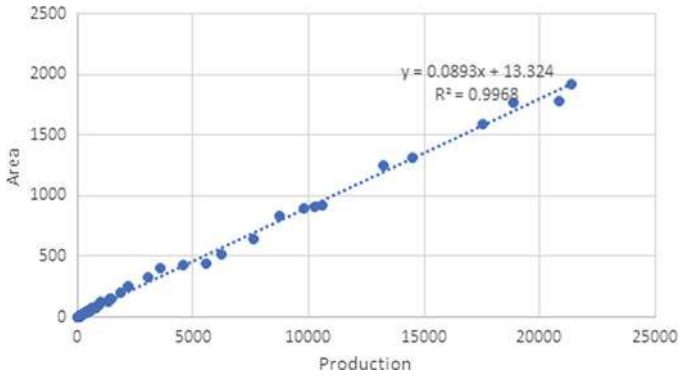


Fig. 3 Regression line

Table 2 Linear regression coefficient

Variable	Estimate	Standard error	T value	*P*-value
Intercept	−134.40268	66.98876	−2.006	0.0413
Area	11.15783	0.09774	114.156	0.02
R-Squared Error		0.9968		
Adjusted R-Squared Error		0.9967		

Table 3 Comparison of actual land area and predicted land area

Actual land area (Ha)	Predicted land area (Ha)
11	16
33	37
35	38
55	60
80	82
100	95
131	130
435	432

the predicted area. From the analysis made with the results of scatterplot, correlation function, R-squared value, and *P*-value, it can be verified that the proposed land suitability prediction model is more effective in forecasting the suitable land area.

6 Conclusion

In this paper, an efficient Demand-based Land suitability prediction model is proposed to forecast the suitable land area according to the demand. The proposed Land suitability prediction model has been evaluated using the sample datasets of the onion by considering the area as a dependent variable and production as the independent variable. As the dependent and independent variable is highly correlated and the p values have high significance, an R-squared value of 99.6% concluded that the proposed land suitability prediction model is more effective and accurate. Hence, the developed land suitability prediction model allows farmers to select the suitable land area and it also minimizes the waste and resources used for the cultivation and thereby it also raises the income of farmers. In the future, the land suitability prediction model can be enhanced by using multiple variables.

References

1. Kadhka S (2017) Reducing food waste vital for India's food security
2. Srivastava R (2020) India grows more food, wastes more, while more go hungry
3. Bharucha J (2018) Tackling the challenges of reducing and managing food waste in Mumbai restaurants. Br Food J 120:639–649
4. Artiuch P, Kornstein S (2012) Sustainable approaches to reducing food waste in India
5. Balaji Prabhu BV, Dakshayini M (2019) Regression-based AGRO forecasting model. In: International conference on emerging technologies in data mining and information security, pp 479–490
6. Balaji Prabhu BV, Dakshayini M (2020) An effective multiple linear regression-based forecasting model for demand-based constructive farming. Int J Web-Based Learn Teach Technol 15:1–18
7. Ayu Purnamasari R, Noguchi R, Ahamed T (2019) Land suitability assessments for yield prediction of cassava using geospatial fuzzy expert systems and remote sensing. Comput Electron Agric 166:105018
8. Taghizadeh-Mehrjardi R, Nabiollahi K, Rasoli L, Kerry R, Scholten T (2020) Land suitability assessment and agricultural production sustainability using machine learning models. Agronomy 10:1–20
9. Mokarram M, Hamzeh S, Aminzadeh F, Zarei AR (2015) Using machine learning for land suitability classification. West Afr J Appl Ecol 23:63–73
10. Senagi K, Jouandeau N, Kamoni P (2017) Using parallel random forest classifier in predicting land suitability for crop production. J Agric Informatics 8:23–32
11. Bagherzadeh A, Ghadiri E, Souhani Darban AR, Gholizadeh A (2016) Land suitability modeling by parametric-based neural networks and fuzzy methods for soybean production in a semi-arid region. Model Earth Syst Environ 2:1–11
12. Parthiban C,Krishnan MB (2016) Expert system for land suitability evaluation using data mining's classification techniques: a comparative study. Int J Comput Trends Technol 33:87–92
13. Sarmadian F, Keshavarzi A, Rooien A, Zahedi G, Javadikia H (2014) Support vector machines based-modeling of land suitability analysis for rainfed agriculture. J Geosci Geomatics 2:2–3
14. Elsheikh R, Mohamed Shariff ARB, Amiri F, Ahmad NB, Balasundram SK, Soom MAM (2013) Agriculture Land Suitability Evaluator (ALSE): a decision and planning support tool for tropical and subtropical crops. Comput Electron Agric 93:98–110

15. Komolafe EO, Awoyelu IO, Ojetade JO (2019) Predictive modeling for land suitability assessment for Cassava cultivation 9:21–31
16. Balaji Prabhu BV, Dakshayini M (2021) Machine learning-based decision support system for effective quality farming. Int J Grid High Perform Comput 13:82–109
17. Balaji Prabhu BV, Dakshayini M (2020) Computational performance analysis of neural network and regression models in forecasting the societal demand for agricultural food harvests. Int J Grid High Perform Comput 12:35–47
18. Douglas C, Montgomery, Elizabeth Peck A, Geoffrey Vining G (2012) Introduction to linear regression analysis, 5th edn
19. Prabhu BVB, Dakshayini M (2018) Demand-prediction model for forecasting AGRI-needs of the society. In: Proceedings of International Conference on Inventive Computing and Informatics, ICICI 2017, pp 430–435

Chapter 2
Power Generation Forecasting of Wind Farms Using Machine Learning Algorithms

Tejas Bhardwaj, Sumit Mehenge, and B. Sri Revathi

1 Introduction

Most pollution in the world is caused by generating electricity, and renewable energy sources like wind energy are rapidly emerging to overcome the pollution [1, 2]. By predicting the seasonal fluctuations of winds, wind energy can produce electrical power satisfactorily and can be planned to deliver uninterrupted energy supply [3]. Power generation prediction helps in system planning, effective operation, and reduced cost. Several wind speed and wind power forecasting have been proposed and implemented in literature as shown in Fig. 1.

Forecasting approaches to predict wind energy are classified as four major categories: physical approaches, statistical methods, artificial intelligence-based models, and hybrid models [4, 5]. Pie et al. [6] proposed multi-objective moth flame optimization algorithm and achieved highly accurate forecasts using an improved wavelet neural network (WNN). Sahra et al. [7] suggested a hybrid three-step forecasting model and performance was evaluated using MLP neural network, GMDH, RBF, and SVR. Mucun et al. [8] describes an improved multi-distribution ensemble (MDE) with multiple adopted ensemble members for probabilistic wind power forecast application. AI-based methods are highly preferred because of their ability to deal with nonlinear problems for WPP [9] using Long Short-Term Memory models and multi-objective grey wolf optimizer algorithm in [10] to optimize extreme learning machine model.

T. Bhardwaj (✉) · S. Mehenge · B. Sri Revathi
Vellore Institute of Technology, Chennai, India
e-mail: tejasbhardwaj007@gmail.com

S. Mehenge
e-mail: sumitmehenge@gmail.com

B. Sri Revathi
e-mail: srirevathi.b@vit.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_2

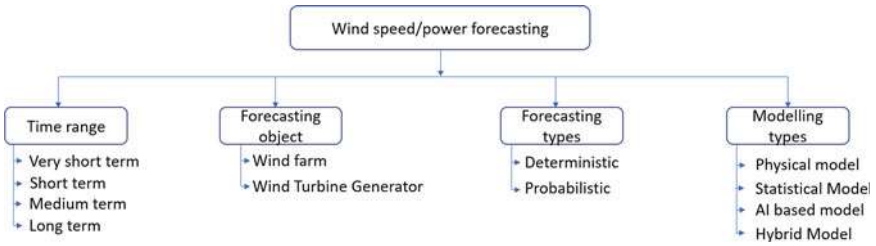


Fig. 1 Wind speed/power forecasting classification

Various studies are performed on wind power forecasting using variety of analysis methods and on various horizons; out of which, statistical methods had provided feasible results [11]. Methods like numerical wind prediction, ensemble forecasting, statistical, and ML have been used in the past, and ML-based methods have proven to be the best. ML provides learning ability to machines without being explicitly programmed; help to forecast results, to report the expected output, and forecast with respect to its past records. Based on the past records, ML algorithms describe the expected outcomes or forecasts with precision [12].

Accurate wind power forecasts reduce the risk of unreliability and allow for better grid planning and integration of wind into power systems since wind is of the stochastic nature. It also helps dealing with challenges of supply balancing and demand and power pooling, and enables better dispatch, improved scheduling, and unit commitment [13]. With highly accurate forecasting results, wind power a fluctuating source can be successfully integrated and balanced by adjusting generation capacity of other plants. Proposed work helps in power pooling as the utilities could forecast the power generation using ML algorithms.

2 Predicting Power Generation from Wind Energy

2.1 Overview of the Proposed Work

This research work using ML algorithms aims to forecast the power generated by wind farms. From the existing literature, it is proven that short-time prediction gives better accuracy [14]. Hence, the prediction was made hourly to give the better accuracy as each wind farm gives different power output according to the wind direction, wind speed, and zonal and meridional wind components [15]. Two ML models are proposed for training and testing different ML algorithms to obtain the best performing algorithm. Data from seven wind mills is taken with a three-year horizon and applied five ML models for comparison. The best algorithm found out of comparison is fine-tuned to give results with higher accuracy. The algorithms are shown in Fig. 2.

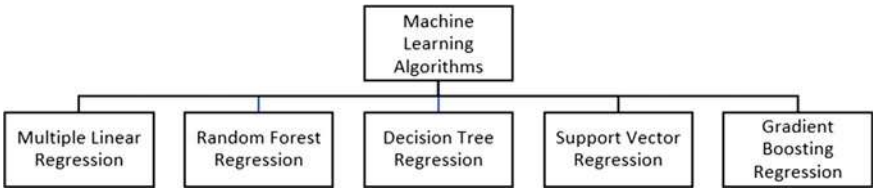


Fig. 2 Different machine learning algorithms used for wind power forecasting

Table 1 Parameters of the wind farms dataset

Parameters	Unit
Time stamp	Hours, day, month, year
Wind speed	m/s
Wind direction	Degree
Zonal wind components	m/s
Meridional wind components	m/s
Power generated	MW

A GUI application is created which uses the best performing fine-tuned algorithm so that the ML algorithms can be employed in other wind farms.

2.2 Description of Model and Dataset

The ML model was created such that the 80% of the data was used to train new data and the rest 20% of the data was used to test the model [16]. The data was obtained from National Institute of Wind Energy (NIWE). The dataset contained recordings of seven wind farms for a three-year horizon having hourly recordings and had 18,757 entries. The parameters of the dataset are given in Table 1.

3 Proposed Models and Methodology

3.1 Data Preprocessing

In data preprocessing, the data is modified or encoded or bring into a state where machine can easily start analyzing [17]. In this paper, in the data preprocessing stage median was taken into consideration for those missing values as taking median values is more reliable than mean or mode in data science.

Existing algorithms like Autoregressive Integrated Moving average (ARIMA) [18], Vector autoregressive (VAR) model [19, 20], ANNs [21], support vector

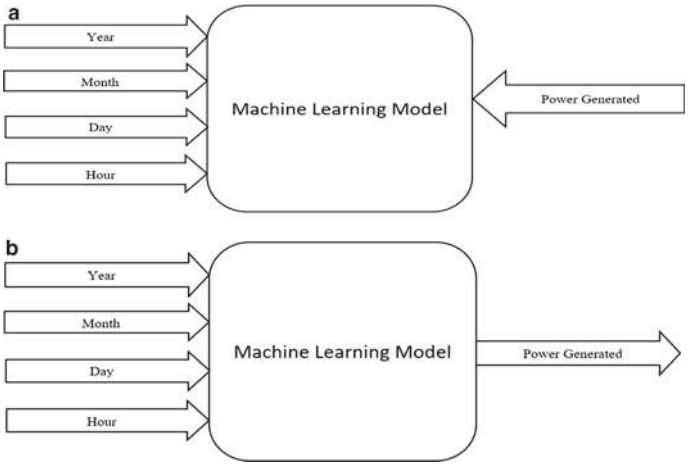


Fig. 3 **a** Training proposed model 1, **b** testing proposed model 1

machines [22], and Kernel Nearest Neural Networks (KNN) [23] were discovered. Five different algorithms based on regression: Multiple Linear Regression (MLR), Random Forest Regression (RF), Decision Tree Regression (DTR), Support Vector Regression (SVR), and Gradient Boosting Regression (GBR) [24–26] are considered for comparison in the presented work. Two different ML models have been proposed, and they have different input parameters.

Training set of proposed model 1 contains input parameters with time stamp giving date, and time of the hourly wind power measurements [27]. The proposed ML model was trained with 80% of data and then tested on 20% data. This model is perfect for predicting power for planning of power outages and assisting power pooling. The independent variables defined here are time stamp and dependent variables being the power generated. The Scientific Python Development environment (SPYDER) was used for the proposed work. Figure 3a and b shows the training and testing of the proposed ML model 1.

3.2 Proposed Model 2

The proposed model 2 was trained with input parameters with timestamp, zonal and meridional wind component, wind speed, and wind direction [28]. The ML model was trained with 80% of data and then tested on 20% data. The independent variables defined here are time stamp, wind speed, wind direction, zonal and meridional wind components, and dependent variables being the power generated. This model can be used before the installation of wind power plants in an unfamiliar site to determine the wind potential at the unfamiliar site so one can find wind efficient energy areas

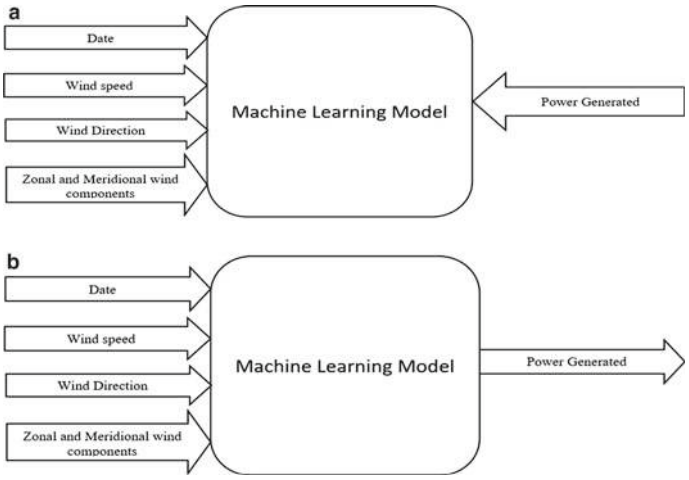


Fig. 4 **a** Training proposed model 2, **b** testing proposed model 2

to set up new farms as this model could be applied to a place different from model-trained places. Figure 4a and b shows the training and testing of the proposed ML model 2.

4 Results and Discussions

After training and testing the ML models with the wind farms dataset, goodness of fit was calculated using the R-squared test which determines the accuracy of the results between actual and predicted power values. Figure 5 shows the wind power forecast with actual and predicted power values for proposed model 1 using decision tree. The data is split at each node of the tree and fits the data very well; therefore, predictions of decision tree algorithm are good.

Figure 6 shows the wind power forecast with actual and predicted power values for proposed model 1 using gradient boosting. As gradient boosting is the combination of multiple decision trees created by improving error which came for previous trees and learning it is not able to produce good performance and under fits the data producing very poor accuracy.

Figure 7 shows the wind power forecast with actual and predicted power values for the proposed model 1 using MLR. As the data is not linear, it is difficult to fit on linear line and is a classic example of under fitting which gives very poor accuracy.

Figure 8 shows the wind power forecast with actual and predicted power values for the proposed model 1 using Random Forest. As Random Forest is the combination of multiple decision trees created using random data, predictions got improved further more than decision tree's prediction and fits the data with good accuracy.

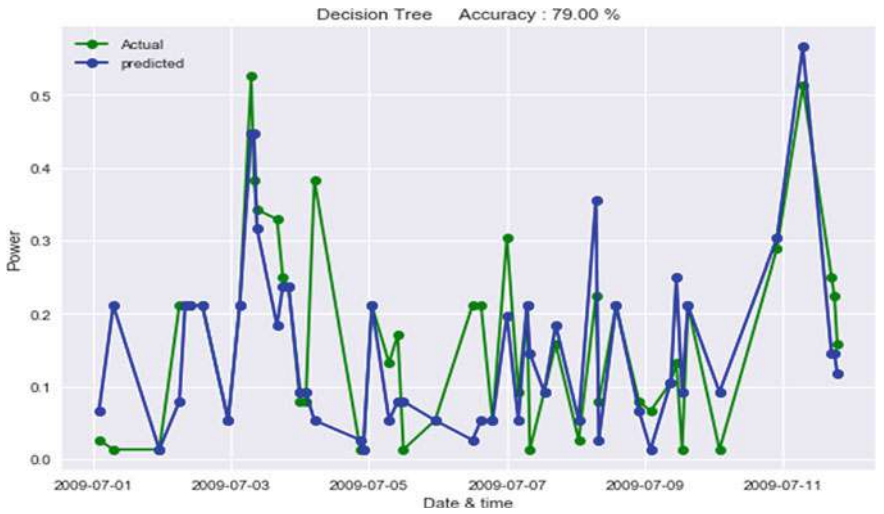


Fig. 5 Wind power forecast results for proposed model 1 using decision tree

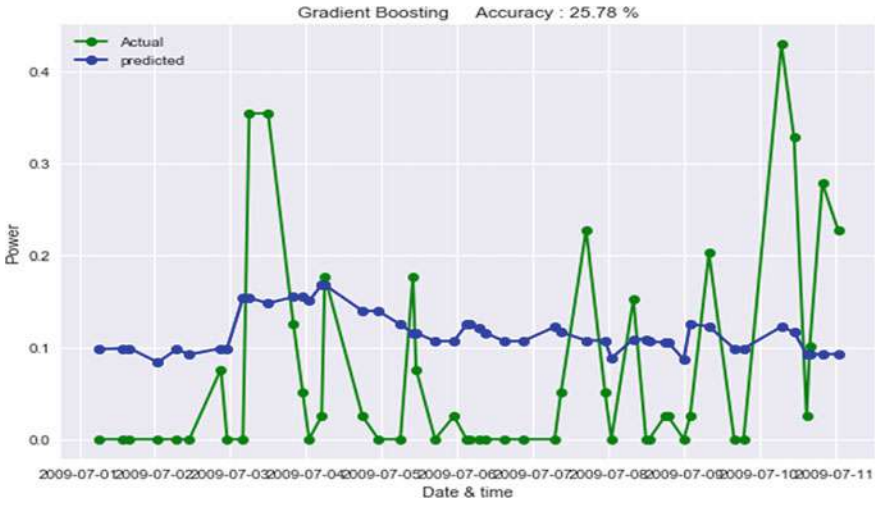


Fig. 6 Wind power forecast results for proposed model 1 using gradient boosting

Figure 9 shows the wind power forecast with actual and predicted power values for the proposed model 1 using SVM. It is difficult for SVM to divide data into classes using vector and under fits the data giving low accuracy; therefore, SVM gives poor result.

Table 2 shows the wind power forecasting accuracy of the different ML algorithms using model 1. The bold values in Table 2 indicate the best performing algorithm's accuracy for the proposed model 1 results.

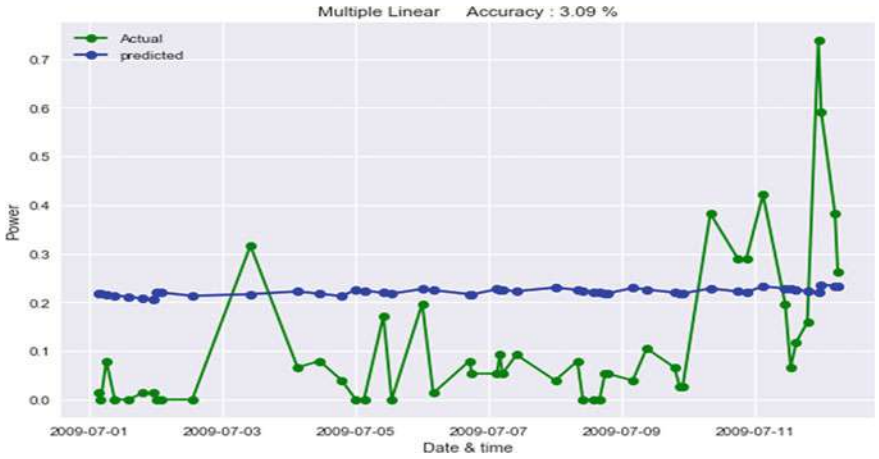


Fig. 7 Wind power forecast results for proposed model 1 using MLR

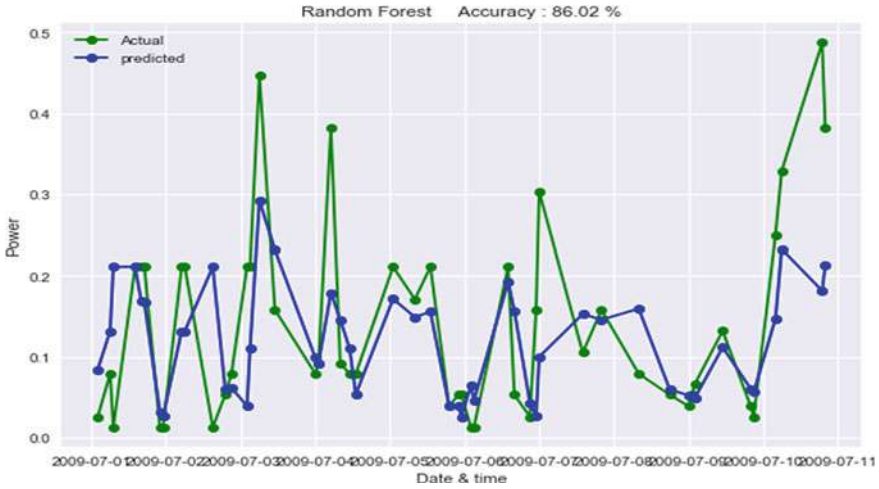


Fig. 8 Wind power forecast with actual and predicted power values for proposed model 1 using Random Forest

The results of this ML model showcase that ML algorithms could be used for to long-term forecasting of wind power, yield accurate result, and could also assist in power pooling.

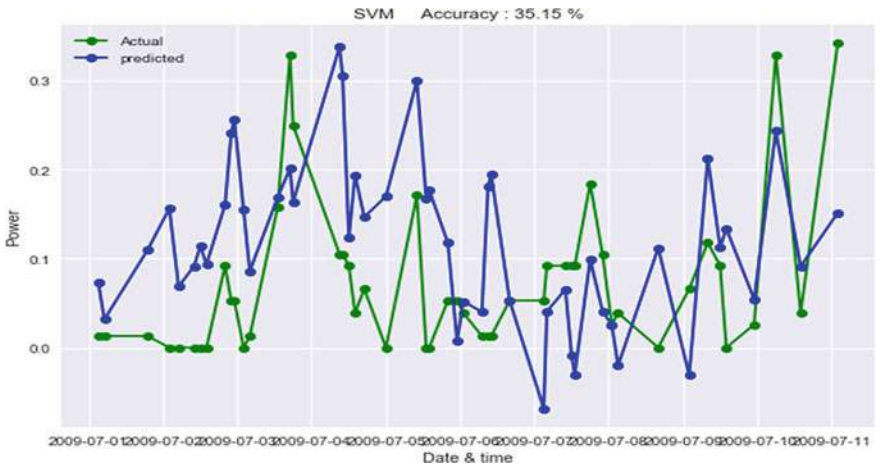


Fig. 9 Wind power forecast results for the proposed model 1 using SVM

Table 2 Wind power forecasting accuracy of the different machine learning algorithms using the proposed model 1

Algorithm	Minimum accuracy (%)	Maximum accuracy (%)
Decision Tree Regression	70.28	79.00
Gradient Boosting Regressor	15.48	25.78
Multiple Linear Regression	1.44	3.09
Random Forest	**78.63**	**86.02**
Support Vector Machine (SVM)	25.48	35.15

4.1 Proposed Model 2 Results

Figure 10 shows the wind power forecast with actual and predicted power values for the proposed model 2 using decision tree. The data is split at each node of the tree and fits the data very well therefore predictions of decision tree algorithm are good.

Figure 11 shows the wind power forecast with actual and predicted power values for the proposed model 2 using gradient boosting. As gradient boosting is the combination of multiple decision trees created by improving error which came for previous trees and learning it is not able to produce good performance and under fits the data.

Figure 12 shows the wind power forecast with actual and predicted power values for the proposed model 2 using MLR. As the data is not linear, it is difficult to fit on linear line and is a classic example of under fitting which gives very poor accuracy.

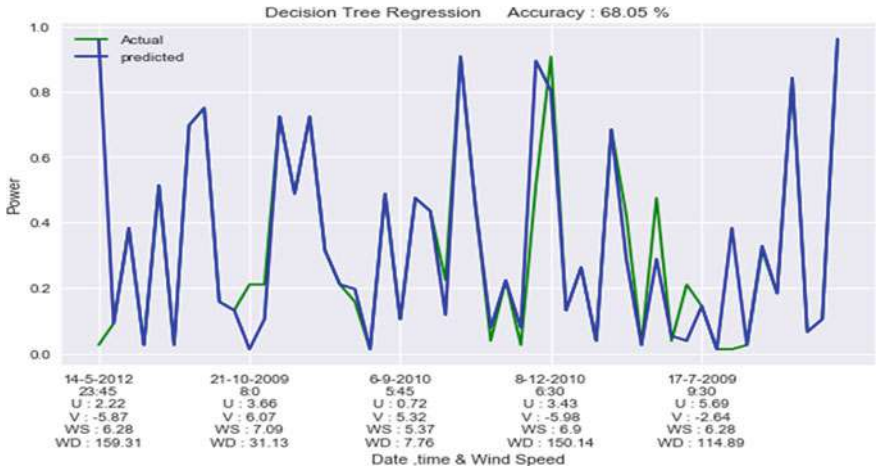


Fig. 10 Wind power forecast results for the proposed model 2 using decision tree

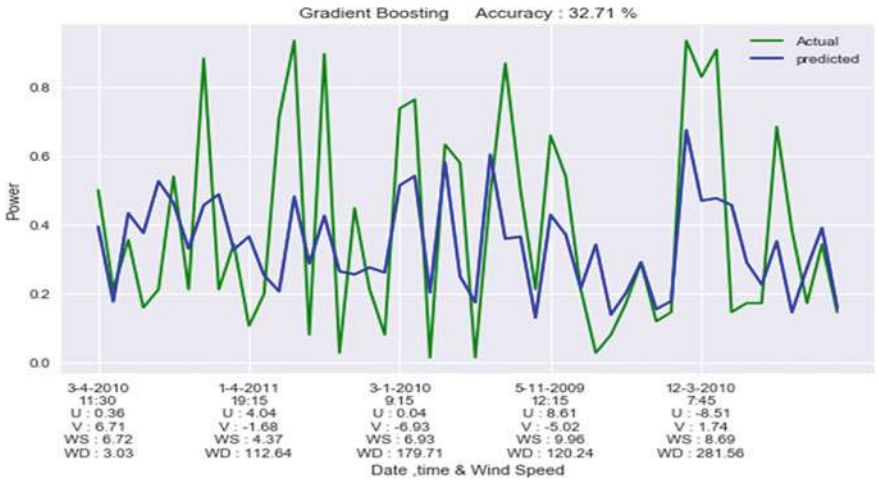


Fig. 11 Wind power forecast with actual and predicted power values for the proposed model 2 using gradient boosting

Figure 13 shows the wind power forecast with actual and predicted power values for the proposed model 2 using Random Forest. As Random Forest is the combination of multiple decision trees created using random data, predictions got improved further more than decision tree's prediction and fits the data with good accuracy.

Figure 14 shows the wind power forecast with actual and predicted power values for the proposed model 2 using SVM. It is difficult for SVM to divide data into classes using vector and under fits the data giving low accuracy; therefore, SVM gives poor result.

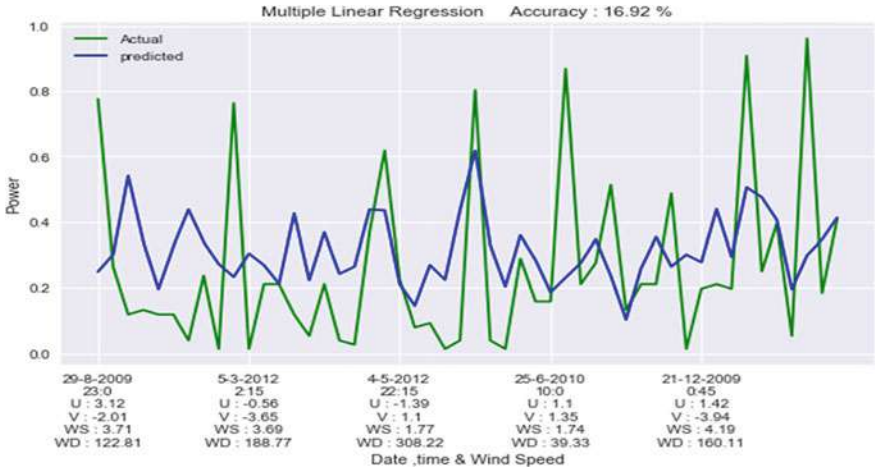


Fig. 12 Wind power forecast results for proposed model 2 using multiple linear regression

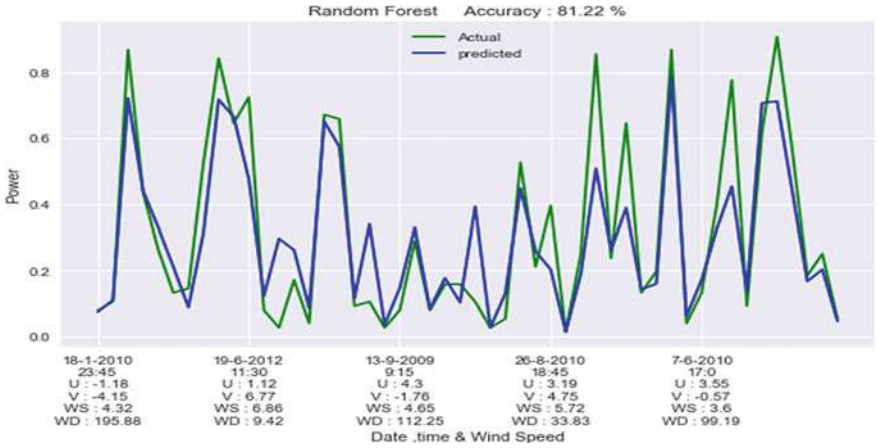


Fig. 13 Wind power forecast results for the proposed model 2 using Random Forest

Table 3 shows the wind power forecasting accuracy of the different ML algorithms using the proposed model 2. The results of this case show case this model can be used beforehand the installation of wind power plants in an unfamiliar site to determine the wind power potential with high efficiency. Random Forest was the algorithm with the best performance; so, further optimizations for better performance were done by fine-tuning methods. Hyper-parameter tuning was used as a fine-tuning method where using grid search CV optimization the hyper-parameters of Random Forest algorithms which are height of the trees and N estimator are calculated and modified for performance improvement.

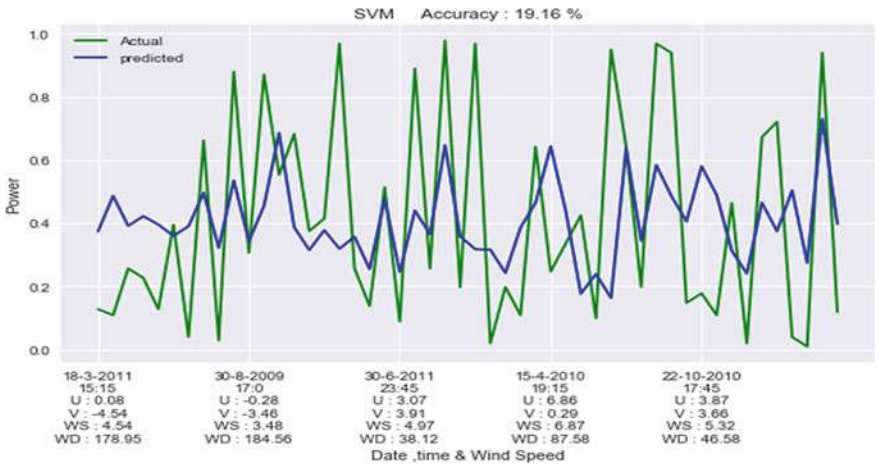


Fig. 14 Wind power forecast results for the proposed model 2 using SVM

Table 3 Forecasting accuracy of different ML algorithms using the proposed model 2

Algorithm	Minimum accuracy (%)	Maximum accuracy (%)
Decision Tree Regression	60.75	68.05
Gradient Boosting Regressor	20.78	32.71
Multiple Linear Regression	8.17	16.92
Random Forest	**73.65**	**81.22**
Support Vector Machine (SVM)	12.48	19.16

Figures 15 and 16 show tuned wind power forecast with actual and predicted power values of RF algorithm for the proposed model 1 and tuned wind power forecast with actual and predicted power values of RF algorithm for the proposed model 2, respectively. After observing the performance of all algorithms, RF algorithm performed the best; thus, hyper-parameter tuning was used as a fine-tuning method. From the results, 2–3% improvement in accuracy is observed in both the proposed model 1 and model 2. Figure 17 shows the output window of the GUI application after prediction is done displaying the predicting power values with the desired timestamp which is the user input.

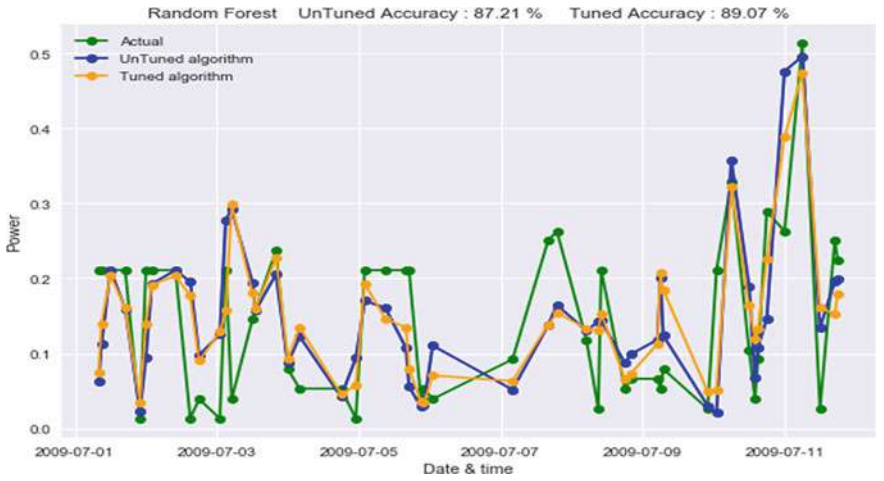


Fig. 15 Tuned wind power forecast results of RF algorithm for the proposed model 1

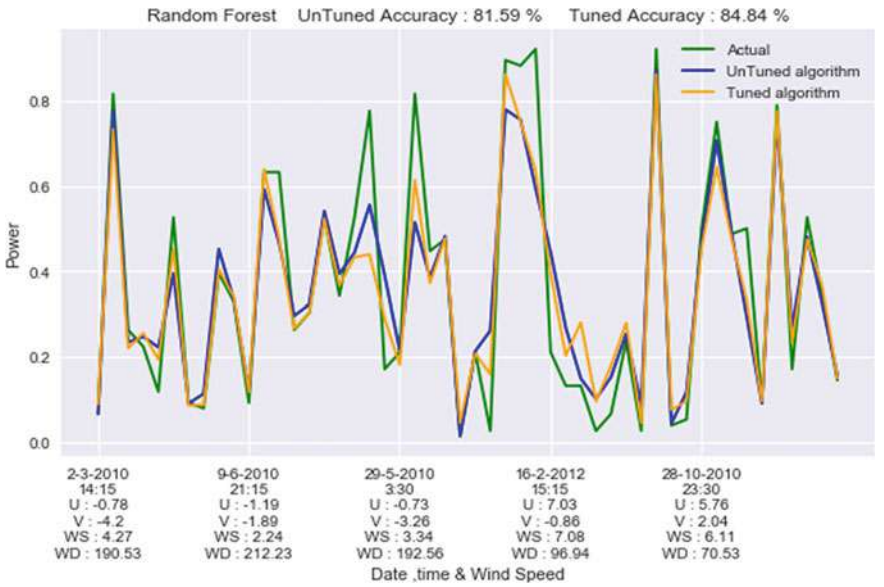


Fig. 16 Tuned wind power forecast results of RF algorithm for the proposed model 2

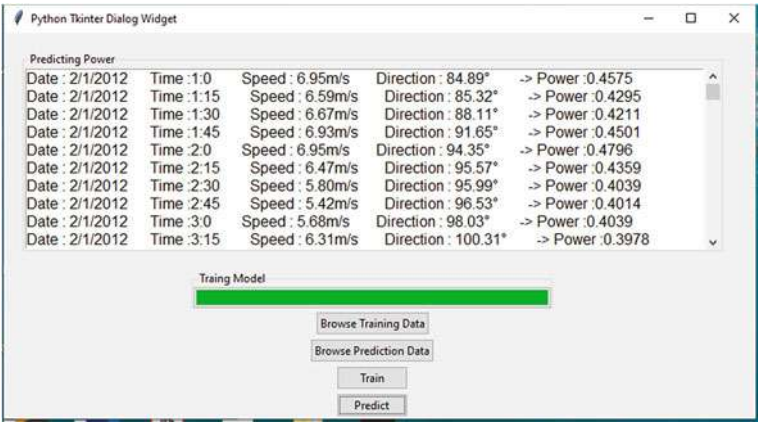


Fig. 17 Output window of the GUI application after prediction is executed

5 Conclusion

In this work, long-term wind power forecasting was accomplished based on wind dataset consisting of various parameters for a three-year horizon using five different types of ML algorithms. Out of the five algorithms, the best performing algorithm was identified and fine-tuned for performance improvements using hyper-parameter tuning. The results showed that for the proposed ML model 1, RF is the best performing ML algorithm with maximum R2 value of 89.07% and, for the proposed ML model 2, RF is the best performing ML algorithm having maximum R2 value of 84.84%. From the analysis and results, the proposed method can be used before installation for finalizing the site selection. A GUI application was developed which could forecast wind power based on historic data fed to it.

References

1. Wang HZ, Wang GB, Li GQ, Peng JC, Liu YT (2016) Deep belief network based deterministic and probabilistic wind speed forecasting approach. Appl Energy 182:80–93
2. Qin Wang Q, Martinez-Anido CB, Wu H, Florita AR, Hodge B-M (2016) Quantifying the economic and grid reliability impacts of improved wind power forecasting. IEEE Trans Sust Enrgy 7(4):1525–1537
3. Lima JM, Guetter AK, Freitas SR, Panetta J, de Mattos JGZ (2017) A meteorological–statistic model for short-term wind power forecasting. J Control Autom Electr Syst (Springer 2017)
4. Wang Y, Zou R, Liu F, Zhang L, Liu Q (2021) A review of wind speed and wind power forecasting with deep neural networks. Appl Energy 304

5. Hanifi S, Liu X, Lin Z, Lotfian S (2020) A critical review of wind power forecasting methods—past, present and future. Energies 13(15)
6. Du P, Wang J, Yang W, Niu T (2019) A novel hybrid model for short-term wind power forecasting. Appl Soft Comput J 80:93–106
7. Khazaei S, Ehsan M, Soleymani S, Mohammadnezhad-Shourkaei H (2022) A high-accuracy hybrid method for short-term wind power forecasting. Energy 238, Part C
8. Sun M, Feng C, Zhang J (2020) Multi-distribution ensemble probabilistic wind power forecasting. Renew Energy 148:135–149
9. Yu R, Gao J, M, Yu, W, Lu, T, Xu, Zhao M, Zhang J, Zhang R, Zhang Z (2019) LSTM-EFG for wind power forecasting based on sequential correlation features. Futur Gener Comput Syst 93:33–42
10. Hao Y, Tian C (2019) A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting. Appl Energy 238
11. Stetco A, Dinmohammadi F, Zhao X, Robu V, Flynn D, Barnes M, Keane J, Nenadic G (2019) Machine learning methods for wind turbine condition monitoring: a review. Renew Energy 133:620–635
12. Shi X, Lei X, Huang Q, Huang S, Ren K, Hu Y (2018) Hourly day-ahead wind power prediction using the hybrid model of variational model decomposition and long short-term memory. Energies 11(11)
13. Scarabaggio P, Grammatico S, Carli R, Dotoli M (2021) Distributed demand side management with stochastic wind power forecasting. IEEE Trans Cntrl Syst Tech 1063–6536
14. de Alencar DB, de Mattos Affonso C, de Oliveira RCL, Rodríguez JLM, Leite JC, Filho JCR (1976) Different models for forecasting wind power generation: case study. Energies 2017, 10
15. Pearre NS, Swan LG (2018) Statistical approach for improved wind speed forecasting for wind power production. Sustain Energy Technol Assess 27:180–191
16. Sharifzadeha M, Sikinioti-Locka A, Shaha N (2019) Machine-learning methods for integrated renewable power generation a comparative study of artificial neural networks, support vector regression, and gaussian process regression. Renew Sustain Energy Rev 108:513–538
17. Liu H, Mi X, Li Y (2018) Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network. Energy Convers Manag 156:498–514
18. Ekström J, Koivisto M, Mellin I, Millar RJ, Lehtonen M (2018) A statistical modeling methodology for long-term wind generation and power ramp simulations in new generation locations. Energies 11:2442
19. Dowell J, Pinson P (2016) Very-short-term probabilistic wind power forecasts by sparse vector autoregression. IEEE Trans Smart Grid 7(2):763–770
20. Meng A, Ge J, Yin H, Chen S (2016) Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. Energy Convers Manag 114:75–88
21. Qin Y, Li K, Liang Z, Lee B, Zhang F, Y, Gu, Zhang L, F, Wu, Rodriguez D (2019) Hybrid forecasting model based on long short term memory network and deep learning neural network for wind signal. Appl Energy 236:262–272
22. Sun G, Jiang C, Cheng P, Y, Xu L, Wang YF, He Y (2018) Short-term wind power forecasts by a synthetical similar time series data mining method. Renew Energy 115:575–584
23. Demollia H, Dokuzb AS, Ecemisb A, Gokcek M (2019) Wind power forecasting based on daily wind speed data using machine learning algorithms. Energy Convers Manag 198:111823
24. Li C, Lin S, Xu F, Liu D, Liu J (2018) Short-term wind power prediction based on data mining technology and improved support vector machine method: a case study in Northwest China. J Cleaner Prod 205
25. Peng T, Zhou J, Zhang C, Zheng Y (2017) Multi-step ahead wind speed forecasting using a hybrid model based on two stage decomposition technique and AdaBoost-extreme learning machine. Energy Convers Manage 153:589–602
26. Malakhov A, Goncharov F, Grynazina E (2019) Testing machine learning approaches for wind plants power output. IEEE REEPE

27. Yana J, Ouyangb T (2018) Advanced wind power prediction based on data-driven error correction. Energy Convers Manage 180:302–311
28. Kariniotakis G (2017) Renewable energy forecasting. Woodhead Publishing

Chapter 3
Music Recommendation System Based on Emotion Detection

Deepa Parasar, Ishveen Sahi, Shivangi Jain, and Arjun Thampuran

1 Introduction

Many people regard music as an escape mechanism from their overwhelming emotions. Each individual has preferred genres of music. These preferences are highly dependent on the state of mind of a person.

Music players these days provide a wide range of functionality which include fast forward, shuffle, music categorization, download, voice-based search, stream music on smart home devices, and so on. These functions satisfy the requirements of the users to a certain extent. However, they have to go through the time-consuming process of manually scanning music playlists to discover songs that correlate with their current mood and behavior. Often users lose interest in listening to music because of this tedious process.

Moreover, numerous new music tracks and albums are released every day in different languages and diverse cultures. Users find it challenging to stay updated to the latest releases of albums because of the same reason. Hence, a feature that can resolve the problems stated above would be a valuable addition to the design of the existing music players.

Deep learning technologies have progressed significantly in the detection of human emotions by making tasks easier and more efficient for users. Various techniques that make use of facial expressions or voice modulation are incorporated to detect the user's current frame of mind.

Since preferred genres of music are highly subjective, it would be convenient for the users if music player applications could recommend and sort a playlist for them as per their mood.

D. Parasar (✉) · I. Sahi · S. Jain · A. Thampuran
Amity School of Engineering and Technology, Amity University Maharashtra, Maharashtra, India
e-mail: dparasar@mum.amity.edu

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_3

2 Related Work

Numerous notable works have been done in the field of human emotion detection.

Dolly Reney and Dr. Neeta Tripathi proposed an emotion detection model using the Viola–Jones face detection algorithm and KNN classifier [1]. They considered the user's voice along with facial data for emotion detection. Viola–Jones face detection algorithm was used to create the database for the face detection, and the KNN classifier algorithm was used to find the face and emotion of the person. They achieved an overall average of 95% accuracy on different lengths of the database. However, their model was not effective in detecting tilted or turned faces and was highly sensitive to the uneven lightening of the input images. It required high memory to train the model, making it computationally expensive. Moreover, with a humungous amount of data, the prediction rate drastically decreased.

A correlation coefficient-based approach was applied by Chang et al. [2]. It was used to determine features of the music that evoke emotions. These features were used to train two Support Vector Machines (SVMs) to classify music that induced happiness, sadness, anger, or peacefulness. General features that were used in audio signal recognition and classification were Linear Prediction Cepstral Coefficient (LPCC) and Mel Frequency Cepstral Coefficient (MFCC). A feature selection algorithm called Sequential Floating Forward Selection (SFFS) was taken into account to classify discriminative features. The Pearson product-moment correlation coefficient was applied. They attained an average hit rate of 68.21% through this model. However, it performed poorly in the presence of high noise, i.e., when the target classes were overlapping with the features. In cases where the number of features for each data point exceeded the number of training data samples, the SVMs underperformed.

Debishree Dagar, Abir Hudait, H. K. Tripathy, and M. N. Das designed a facial expression recognition system using Gabor feature extraction and neural network [3]. They took live video stream as input and a Multi-Layer Perceptron (MLP) neural network was used to detect the face frame from the video. The main advantage of using Gabor filters was that they provided optimal simultaneous resolution in both space and frequency domains with the downside of having chances of high redundancy in the captured features. Due to some features of the frontal dynamics of some people, the principal components of various emotions were overlapping which gave them an ambiguous or wrong output.

Additionally, Mithun Madathil analyzed the collaborative filtering of Spotify API [4]. He mentioned that Spotify uses the K-nearest neighbor technique (KNN) to calculate the distance between different users with similar interest in music. He developed a taste profile for each user with individual taste in music, used the KNN algorithm for distance calculation, and recommended content based on this information. He also incorporated a feedback system that records the frequency of pressing the 'like' button when users find songs matching their tastes. The future service for the recommendation is based on the history of clicking the 'like' record. The limitation of this approach was the cold start problem, which occurred when a new user entered the system with no ratings to recommend. In addition, it gave poor personalization for

users with unusual taste because either the ratings were few or such users did not have many neighbors.

Rajesh K M and Naveenkumar M developed an emotion detection system using the Fisherface algorithm [5]. It performed two classes of analyses to achieve recognition, i.e., Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), respectively. The limitation of this model was that it worked only with a frontal view of the face.

Youyi Cai et al. created a video-based emotion recognition system using Convolution Neural Network (CNN) and Bi-directional Recurrent Neutral Networks (BRNN) [6]. CNN modeled spatial dependencies within every image while the BRNN model was used for temporal dependencies. This BRNN model was trained using all available input information in the past and future of a specific frame. The drawback of this model was that the recognition rate for neutral and anxious emotions was way less than baseline. It also failed to detect angry emotion accurately. Most of the anxious and angry samples were erroneously detected as worried or disgust.

Shiqing Zhang and Shiliang Zhang made a hybrid deep model consisting of CNN and 3D-CNN in combination with a Deep Belief Network (DBN) model [7]. Their model contained two separate input streams, i.e., the audio network processing audio signals with a CNN model and the visual network processing visual data with a 3D-CNN model. The outputs of these two fully connected layers were then fused in a fusion network built with a DBN model. The major limitation of this model was its expensive computational cost due to massive network parameters. In addition, they used the BAUM-1s dataset wherein spontaneous emotions are more difficult to be recognized.

Tomas Matlovic et al. developed a method of emotion detection using an Electroencephalogram (EEG) sensor [8]. Emotiv Epoc was used to detect the participant's brain activity. They achieved 53% accuracy in emotion detection using this method. They made use of the DEAP dataset and applied Discrete Wave Transform on raw EEG data for pre-processing. To achieve accuracy in the detection of angry and fearful emotions, they had to oversample the data. The oversampling of data can result in overfitting and eventually increase the computational cost.

A facial detection model using Bayesian Network and K2 algorithm was created by Miyakoshi and Kato [9]. This model also considered partial occlusion of the face while detection. As the Bayesian classifier infers from the dependencies among the target attribute and explanatory variables, it can detect emotions without filling gaps of occluded features. K2 algorithm was used to estimate real casual relations among facial features. The major limitation of this model was its computational cost due to full Bayesian learning.

Liyanage C. DE SILVA et al. developed a facial emotion recognition system using multimodal information [10]. This model took 3 types of input, i.e., only audio clip, only video clip, and audio and video clips together. A weighting matrix was already set which had different weights based on audio and video dominance. The inputs were compared with this matrix to detect the emotion. This system failed to classify some emotions for different speakers from diverse cultural backgrounds. Also, the accuracy of positive emotion detection was more than negative emotion detection.

3 Methodology

3.1 General Architecture of the Model

The general architecture of the model is as shown in Fig. 1. The model was trained on the FER2013 dataset, after which it was loaded with its corresponding weights and the Haar Cascade algorithm was implemented to detect the face with higher accuracy. Haar Cascade algorithm uses edge detection features that can capture specific facial features like eyes, lips, eyebrows, and so on. There are two categories of images in its database, i.e., positive images and negative images. The former consists of an object which is absent in the latter.

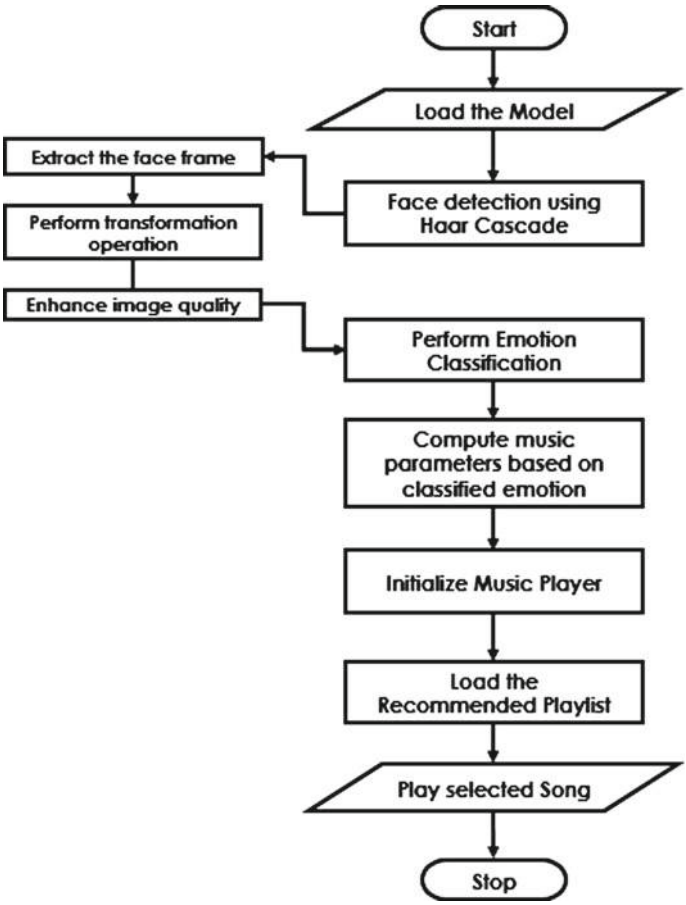


Fig. 1 Proposed methodology

After face detection, the frame will be cropped as per the size of all the images in the dataset, which will further be used for training the model. There are many reasons due to which prediction for a certain emotion can be inaccurate. One such reason is poor illumination. To overcome this caveat, it is necessary to enhance the quality of the image. In addition, the model expects a gray-scale formatted image to match the format of the images already existing in the dataset.

A convolution layer was used to take the 3D tensor as input. Hence, it is required to expand the dimension of the frame from 2 to 3D. Emotion is classified in real time, and once it is detected, a dialogue box prompts that takes basic user information like name, age, favorite genres, and favorite artists. All the information provided by the user is appended to the database.

Here, we used the MongoDB database. A mongod or mongo daemon service runs in the background and listens on port 27017. The program retrieves favorite artists' URI (Uniform Resource Indicator) from a dictionary that comprises the name of the artist and their corresponding URI. Following this, a request is made to the Spotify API web server with certain parameters.

Few operations are performed on the received response from the Spotify server, which would be mentioned thoroughly and clearly in Sect. 3.3. It might take a while to generate a user-specific playlist. In the meantime, a music player that is created by using Tkinter and pygame libraries gets initialized and starts to play songs. Once the playlist is generated, the application terminates and a window with the list of recommended songs pops up.

3.2 Training the Emotion Detection Model

FER2013 dataset comprises 35,000 images and 7 emotion labels, i.e., happy, sad, neutral, surprise, anger, disgust, and fear. Since fear and disgust emotions were not considered, all those tuples were eliminated from the dataset before training the model. Then, the dataset was split into train set and test set in a ratio of 3:1.

Considering the model is trained on a Convolutional Neural Network, the convolution layer expects a 3D tensor as input. Hence, there was a need to expand the dimension of all the images of the train set to 3 dimension. The model is defined by specifying different layers and their appropriate parameters.

It was compiled using the categorical cross-entropy loss function and Stochastic Gradient Descent (SGD) optimizer. The number of epochs, number of batches, and batch size were set after which the model was trained.

Once the training process was completed, the model was saved into a JSON (JavaScript Object Notation) formatted file as it is not feasible and efficient to train the model every single time. The weights were saved as well in a '.h5' file format.

The framework of the model is shown below in Fig. 2.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 22, 22, 32)	832
max_pooling2d_6 (MaxPooling2	(None, 11, 11, 32)	0
dropout_2 (Dropout)	(None, 11, 11, 32)	0
conv2d_7 (Conv2D)	(None, 4, 4, 64)	51264
max_pooling2d_7 (MaxPooling2	(None, 2, 2, 64)	0
dropout_3 (Dropout)	(None, 2, 2, 64)	0
flatten_3 (Flatten)	(None, 256)	0
dense_6 (Dense)	(None, 64)	16448
dense_7 (Dense)	(None, 7)	455

Fig. 2 Framework of the model

3.3 Music Recommendation System Implementation

As mentioned in Sect. 3.1, once the user emotion is detected, a dialogue box prompts that takes in basic user information. It is fundamental for a user to have a Spotify account to establish a connection with the Spotify API web server. If a Spotify account of that user exists, then the next step is to get the client id and client secret id. Client id serves as a unique identifier to our application while client secret id is the pass that is used to access and retrieve response from the Spotify API web server. Emotion Detection training system is shown in Fig. 3.

Authentication of the user is a mandatory step before requesting data from the Spotify API web server. If the authentication is successful, then the program takes the user's favorite artists and accesses their URI from the dictionary. It sends a request to the server to access the information about the user's favorite choice of artists and similar artists to that.

Artists who have their popularity score above 50 get selected for the recommendations. Following which a request is made to the server to retrieve the top tracks of all the filtered artists. Lastly, the audio features of every track are returned by the server according to which a recommendation list is generated. Audio features of a track include acoustics, danceability, energy, instrumentals, liveness, loudness, speechiness, valence, tempo, etc.

We considered features like acoustics, danceability, valence, tempo, and energy to generate a playlist as per the user's emotion. As mentioned earlier, it may take some time to curate a playlist for the user. In that time, a music player pops up and starts playing some songs to entertain the user. Once the recommendation list is ready, the music player application closes and the recommendation list pops up.

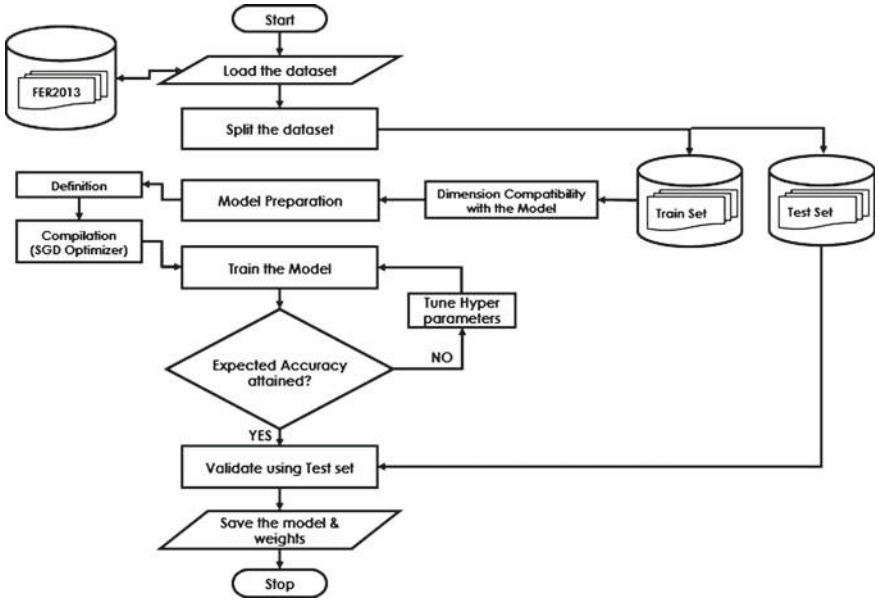


Fig. 3 Emotion detection training

4 Implementation Details

Here, Fig. 4 shows the form that pops up after emotion detection. This form takes basic user information like name, age, favorite artist, and favorite genres. This information is stored in the database for future purposes. Figure 5 shows the GUI of the music player that is created by using Tkinter and pygame libraries. The music player gets initialized and starts to play songs while the recommendation playlist is being generated.

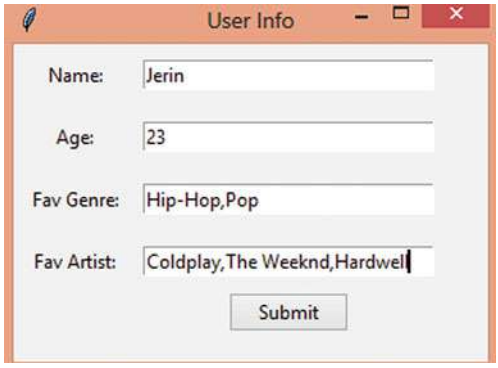


Fig. 4 Form taking user information

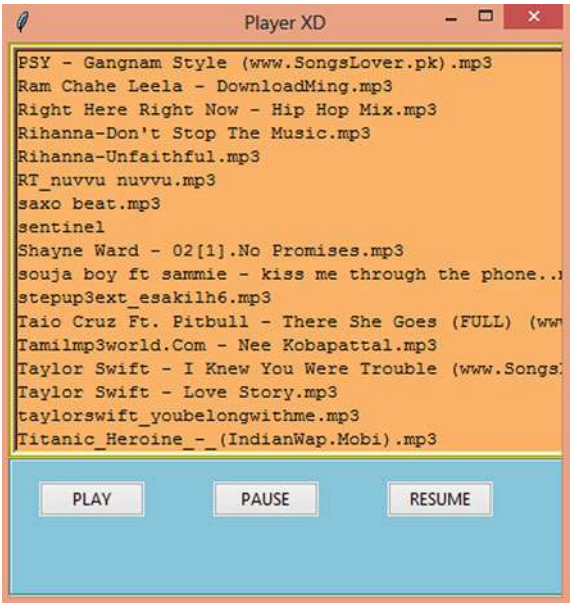


Fig. 5 GUI design of the music player

The following description show the results of emotion detection for the five emotions considered and the recommended playlists:

For neutral emotion as shown in Fig. 6, the danceability factor and the valence threshold was kept greater than 0.3, while the threshold for tempo was set greater than 7.0. Recommended playlist is shown in Fig. 7.

For happy emotion as shown in Fig. 8, the output was most accurate when the danceability factor was set greater than 0.4, valence greater than 0.6 and energy greater than 0.7. The threshold value for valence was chosen high for the recommendations of happy emotion because high valence music tracks are positive and

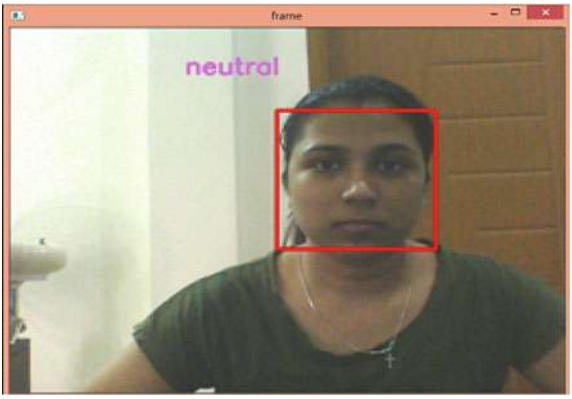


Fig. 6 Emotion detection for neutral

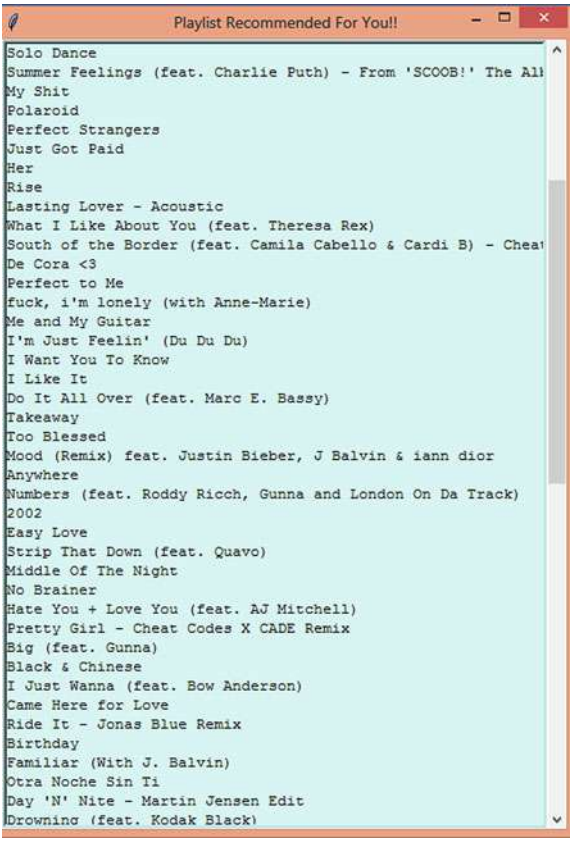


Fig. 7 Playlist recommendation for neutral

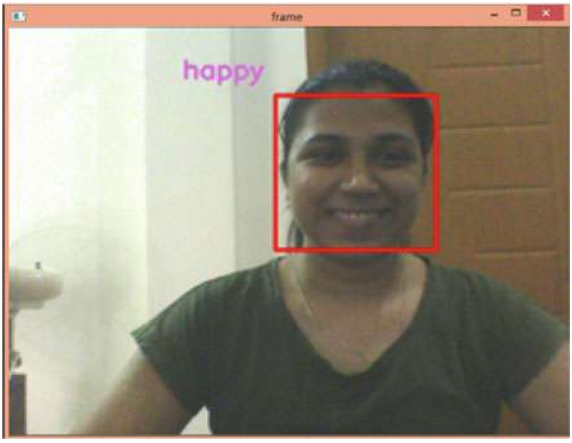


Fig. 8 Emotion detection for happy

Fig. 9 Playlist recommendation for happy

cheerful. Energy represents the intensity or speed of a track. During happy emotion, people tend to listen to tracks which are fast, loud, or noisy. Recommended playlist is shown in Fig. 9.

For the recommendation of sad emotion as shown in Fig. 10, the threshold for acousticness was set greater than or equal to 0.4 and valence was set less than 0.5. This is because people prefer listening to melodious and more musical instrumental songs during sad times. Also, they wish to listen to less positive music, and hence, the threshold of valence is kept low. Recommended playlist is shown in Fig. 11.

For surprise emotion as shown in Fig. 12, danceability was set to greater than 0.7, acousticness less than 0.3, valence greater than 0.6 and tempo less than 85. All these thresholds were optimized repeatedly until the most accurate output was achieved. Recommended playlist for surprise emotion is shown in Fig. 13.

Lastly, for angry emotion as shown in Fig. 14, the threshold of valence was set less than 0.6 and tempo was set to greater than 80. The tempo was set high for this emotion because people prefer listening to heavy music while they are angry and tempo represents Beats per Minute (BPM). Recommended playlist is shown in Fig. 15.

Fig. 10 Emotion detection for sad

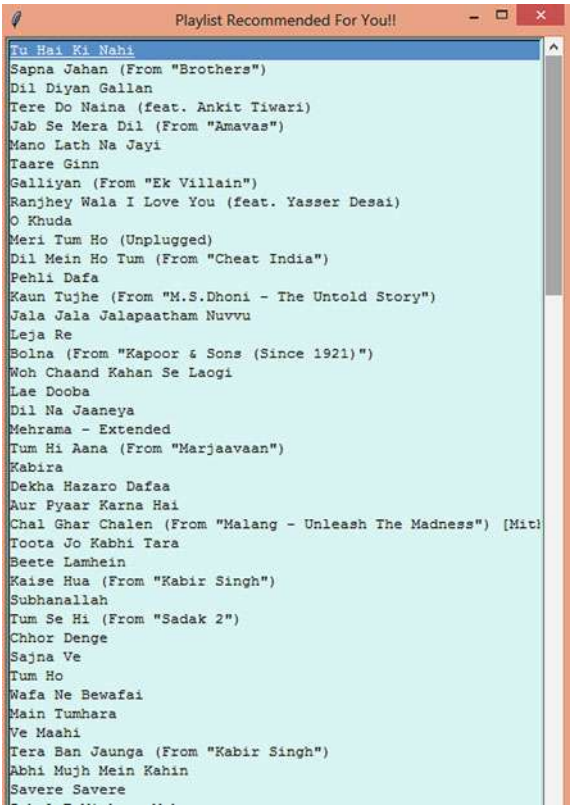


Fig. 11 Playlist recommendation for sad

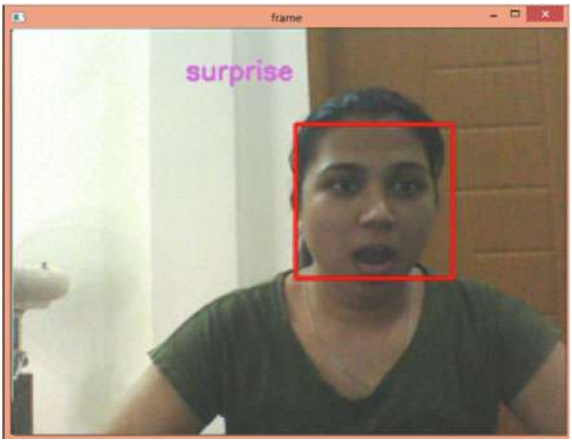


Fig. 12 Emotion detection for surprise

Fig. 13 Playlist recommendation for surprise

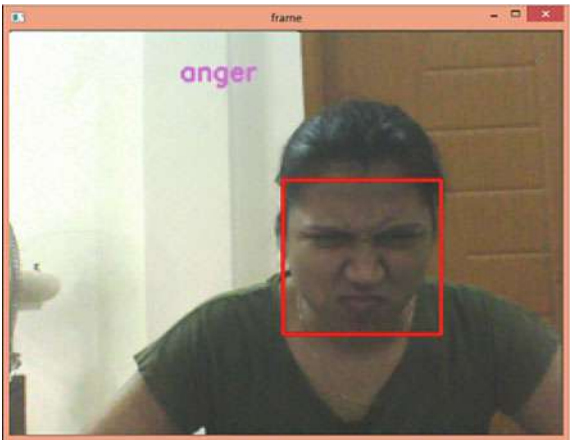


Fig. 14 Emotion detection for angry

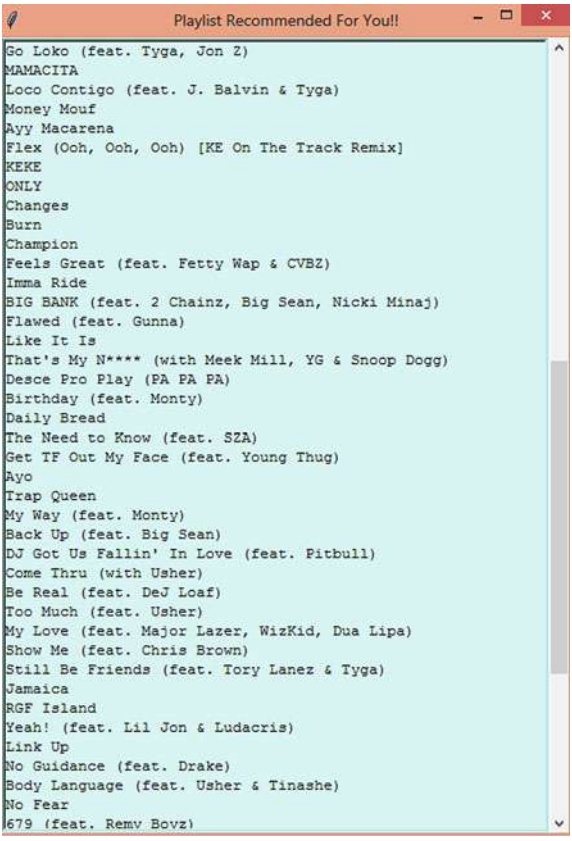


Fig. 15 Playlist recommendation for angry

5 Conclusion

The motive behind developing this project was to create a convenient music application. The sole aim was to reduce the time taken by users to search for appropriate songs that were in accordance with their frame of mind and to keep them updated with new music tracks. The proposed model takes a real-time captured image of the user, his favorite genre, and artists as inputs, detects the emotion of the subject, and outputs a recommended playlist of songs that match the user's detected emotion. The purpose of accepting users' favorite genres and artists was that no matter what emotion a user is feeling, he would always prefer to listen to his favorite artists. The emotion detection module is responsible to bring forth the songs which suit the user's current mood. The accuracy of music prediction completely differs from person to person.

The proposed methodology generates a playlist by using various audio features of the songs, while the other author [4] as mentioned in Sect. 2 used KNN classification and collaborative filtering. His model depended entirely on user feedback or rating of songs which gave a poor start to new users of the system and to the users who had fewer neighbors with similar likings around them. Additionally, the song recommendation could be faulty for unpopular songs since they have fewer feedbacks from previous users. On the contrary, our system presents a complete list of songs based on the audio features to provide the users with a wide variety of music that resonates with their emotions. Thus, the user experience of our system is well received from its first use.

Another author used a combination of MLP and Gabor filters for emotion detection [3]. This led to high chances of redundancy in capturing frontal face features which resulted in incorrect outputs. Instead, we incorporated CNN to train the model and achieved a categorical accuracy of 87.22%.

Our emotion detection model could be further enhanced by increasing the number of images in the dataset (currently FER2013 dataset comprises 35,000 images). The playlist generation could also be advanced by fine-tuning the ranges of audio features like danceability, tempo, acoustics, etc. Overall, this system saves user's time by curating a customized playlist for them. It might have its limitations; nevertheless, it still provides a better user experience by generating a customized playlist and that too at ease of a click.

References

1. Reney DF, Tripaathi NS (2015) An efficient method to face and emotion detection. In: 5th international conference on communication systems and network technologies. IEEE, Gwalior
2. Chang C et al (2010) A music recommendation system with consideration of personal emotion. In: International computer symposium. IEEE, Tainan
3. Dagar D et al (2016) Automatic emotion detection model from facial expression. In: International conference on advanced communication control and computing technologies. IEEE, Ramanathapuram

4. Madathil MF (2017) Music recommendation system spotify—collaborative filtering. In: High performance and automatic computing. https://hpac.cs.umu.se/teaching/sem-mus-17/Reports/Madathil.pdf. Last accessed 2021/30/10
5. Rajesh MF, Naveenkumar MS (2016) A robust method for face recognition and face emotion detection system using support vector machines. In: International conference on electrical, electronics, communication, computer and optimization techniques. IEEE, Mysuru
6. Cai Y et al (2016) Video based emotion recognition using CNN and BRNN. In: Tan T et al (eds) Pattern recognition. CCPR 2016. Communications in computer and information science, vol 663. Springer, Singapore
7. Zhang S et al (2018) Learning affective features with deep model for audio-visual emotion recognition. IEEE Trans Circuits Syst Video Technol 28(10):3030–3043
8. Matlovic T et al (2016) Emotion detection using facial expressions recognition and EEG. In: 11th international workshop on semantic and social media adaptation and personalization. IEEE, Thessaloniki
9. Miyakoshi YF, Kato SS (2011) Facial emotion detection considering partial occlusion of face using bayesian network. In: IEEE symposium on computers & informatics. IEEE, Kuala Lumpur
10. Silva LF, Miyasato TS, Nakatsu RT (1997) Facial emotion recognition using multimodal information. In: International conference on information, communications and signal processing. IEEE, Singapore

Chapter 4
Service Analytics on ITSM Processes Using Time Series

Ioannis Karamitsos, Omar Murad, and Sanjay Modak

1 Introduction

Customer service analytics entails collecting and analysing various customer data and metrics to derive actionable information. Recent developments in data analytics have led to a renewed interest in customer experience. Any contact with the customer service team provides an exciting opportunity to advance our knowledge for the customer collecting data from different sources such as support tickets, emails and social media messages. Data analytics methods play an essential role in the customer experience journey, and it is fast becoming a key instrument for understanding the customer needs and their satisfaction from the product or service offered by the organisation.

Service analytics is an important component in the customer service, and using information technology service management (ITSM) tools, it was possible to forecast prospective customers based on previous customer behaviour.

ITSM tools assist infrastructure and operations (I&O) organisations in managing IT service consumption, the infrastructure that supports the IT services and the IT organisation's responsibilities for delivering business value through these services. These are the most frequently utilised by IT service desks and IT service delivery operations to support incident, request, problem, change, service level, knowledge and configuration management tasks and workflows.

IT service delivery to the customers uses the best practices and standards defined by the ITSM framework [1]. Below are some of the examples of such commonly known standards:

- Information technology infrastructure library (ITIL) [2]
- Control objectives for information and related technologies (COBITs) [3]
- Business process framework (eTom) [4]

I. Karamitsos (✉) · O. Murad · S. Modak
Rochester Institute of Technology, Dubai, UAE
e-mail: ixkcad1@rit.edu

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_4

- Microsoft operations framework (MOF) [5]

There are multiple processes under ITSM [6], and each has a set of guidelines and procedures. Listed below are some of the most common ITSM processes:

- Incident management [7]
- Problem management [7]
- Request fulfilment [7]
- Change management [8]
- Service asset and configuration management [8]
- Release and deployment management [8]
- Information security management [9]
- Service level management [9]

During the daily operation of business functions and services, many tickets are raised from the customers, increasing the incident and request fulfilment records logged in the ITSM system.

This study aims to analyse the data of both incident and request fulfilment processes and to understand the demand level on these two processes throughout the year. By utilising time series methods, we will forecast the expected demand from customers for the services offered and supported by the IT department in a given period using time series analysis.

The main contribution of this paper is as follows:

- Analysing the tickets logged in the ITSM tool for incident and request fulfilment processes.
- The identification of patterns and trends of customers demand on IT managed services
- To evaluate and select a model for forecasting the demand expected in a given period

This remainder of the paper has been organised in the following way. Section 2 begins by laying out the related work on different ITSM processes. Section 3 is concerned with the time series modelling methodology used for this study. Section 4 presents the findings of the study, focussing on the comparison between different forecasting models for the ITSM process. Section 5 analyses the results of selected time series model with real data set implementing forecasting models. Finally, in Sect. 6, the conclusion gives a brief summary of the findings, and areas for further work are identified.

2 Related Work

The field of IT service management covers a wide range of processes, procedures and practices.

Ferreira and Da Silva [10] present the data mining for assessing if an implemented business process is designed following the ITIL framework. The authors apply data mining on the logged tickets for reported issues received by the company, which is considered as incidents to understand and retrieve the behaviour of the business process compared to that of the ITIL incident process.

Goby et al. [11] present the application of data analytics for automating the assignment of helpdesk tickets, allowing options for early detection of backlog and pending incidents and improving the sharing of knowledge within helpdesk members. To achieve this objective, the authors employ topic modelling and predictive analytics applied to a comprehensive data set consisting of logged support tickets for a global automotive supplier.

Delias et al. [12] provide an approach identifying and discovering behavioural trends and patterns in the request fulfilment management process. The authors present real data set containing the logged records of both incidents and problems to identify and provide outcomes facilitating the overall improvement of the request management process. The case study is for an IT system managed by Volvo IT to handle reported issues faced by users and logged as incidents. The focus of the paper is to allocate the areas in the process needing improvement by delivering efficient and analytics-based recommendations for supporting data-driven decision-making.

Clavon [13] presents and compares the use of different methods for predictive analysis and univariate modelling to anticipate the expected load-on-demand management process for the next six weeks in advance. The data collected for the work spans the months of January 2016 and December 2016 and includes information about the date tickets was submitted, the date they were resolved, the business hours during they were resolved. The study was conducted for City of Atlanta department with a total number of 11,760 tickets.

Gerke and Tamm [14] present an approach for the continual improvement of ITIL-based processes using data mining techniques and the ITIL frameworks as a guiding principle and model. By utilising data mining, the authors extract the behaviour and the key performance indicators (KPIs) of existing IT processes and compare them with the defined metrics in the ITIL processes.

Gupta et al. [15] cover the IT service management scope dealing with end-users' interactions with the IT service provider through their tickets in the system. The focus of the work is the assessment and the reduction of input from users in the logged system records. The proposed approach is by assessing the effect of the requests for users' input and determining the influence resulting in the resolution time faced by the end user. According to the authors, the approach identifies two types of requests for users' input:

- Real: Asking for information from the user to process the request
- Tactical: The user input is not required but requested to pause the SLA timer

The authors propose a solution design that will ask for the user input before submitting the ticket providing information that potentially the technician working on the ticket may request, which eventually reduces the real type of user input requests.

As for the tactical type of user input requests, the authors cover it by applying a detection system to distinguish this type of user input requests.

Lin et al. [16] state that the backend devices of an IT infrastructure contribute to generating a significant amount of warning events and incident records, making it challenging to interpret helpful information to improve operational performance. The present approach depends on utilising machine learning for unsupervised clustering algorithms to cluster warning events and incident records based on their text. A key benefit resulting from the approach is the removal of manual classification for warning events and incident records, which are costly and consume time and effort.

3 Time Series Modelling

The analytics modelling includes multiple and various steps and activities. Figure 1 is depicted the times series process.

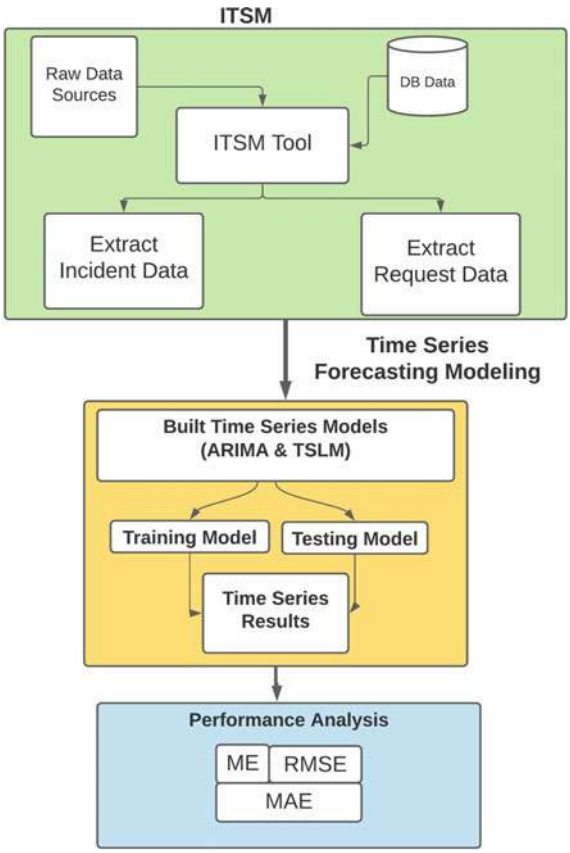


Fig. 1 Time series process flow

The work starts from the ITSM tool to generate the data for both incident and request fulfilment processes. Once the data have been exported, the next and most critical activity is converting the data to time series format. The next part is for time series modelling. It includes using different techniques to build time series models and compare the results and accuracy of each model. The last part is the performance analysis in terms of mean error (ME), root mean squared error (RMSE) and mean absolute error (MAE) metrics. Each of the used time series models is described briefly below:

- **Auto ARIMA**: An automatic procedure that provides for many variations on the algorithm to obtain an autoregressive integrated moving average (ARIMA) model [17].
- **ARIMA**: Combines differencing with autoregression and a moving average model to produce a non-seasonal ARIMA (p, d, q) model where p is the number of lag observations, d denotes the number of times that the raw observations are differenced; also known as the degree of initial differencing, and q denotes the size of the moving average window [17].
- **TSLM**: Fits a linear regression model to time series data and provides additional facilities for handling time series [17].

4 Performance Analysis

Performance measures are of particular value when we come to select the forecasting model for our study. For building and verifying the performance of the forecasting models, each time series data set is split into training and testing sets. The training set is the available data we have to forecast. The testing data set is already realised, but we hold out to test the performance and quality of forecasting models.

When splitting the time series, it is extremely important to ensure that the data points included in the training partition have a time stamp occurring before that of the data points in the testing partition to allow for proper training and validation of the model.

For incident time series and request time series, the training set has all the records from 22/07/2018 till 31/03/2020, whilst the testing set contains all the records from 01/04/2020 till 29/07/2020. Using different forecasting models and techniques is a critical factor in helping to identify which of the models provide the highest accuracy rate.

From the literature review, we were able to identify potential forecasting models suitable for use with the incident and request time series data. These models are as follows:

- ARIMA c (3, 0, 4) and TSLM model for incident data
- ARIMA c (5, 0, 4) and TSLM model for request data

Comparing the forecasted results against the actual results through plots made it easier to filter out the models that are not suitable to work with, but it is not enough to

identify the best forecasting model to use. For this study, the forecast measures were used to evaluate the proposed models. The forecast measures are the mean error (ME), the root mean squared error (RMSE) and mean absolute error (MAE). For that purpose, we need to look into other parameters that include different accuracy measures, and below are the ones mainly used:

- **Mean Error (ME)**: The simple measure of forecast accuracy is the mean or average of the forecasted errors. The mean error is a valuable way of detecting systematic bias in a forecast. Starting at forecast origin t, with $(t + m)$ observations available, generates (m) forecasts. The mean error (ME) is given by:

$$\mathrm{ME} = \sum_{t=1}^{m} (Y_{t+1} - F_{t+1})/m \tag{1}$$

  where the forecast error at the time $(t + 1)$ may be denoted by $Y_{t+1} - F_{t+1}$. The term Y_{t+1} is the actual value at the time $(t + 1)$, and the term F_{t+1} is the forecast value in the same time. As the positive and negative forecast errors tend to offset one another, the mean error is probably to be small. Thus, the ME metric is not always a useful performance measure.
- **Root Mean Squared Error (RMSE)**: This is the root of the average squared distance of the actual and forecasted. The RMSE is given by:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \tag{2}$$

  where mean square error (MSE) is given by: $\mathrm{MSE} = \sum_{t=1}^{m} (Y_{t+1} - F_{t+1})^2/m$. The term Y_{t+1} is the actual value at the time $(t + 1)$, and the term F_{t+1} is the forecast value in the same time.
- **Mean Absolute Error (MAE)**: This measures the absolute error rate of the forecasts. The mean absolute error (MAE) is given by:

$$\mathrm{MAE} = \sum_{t=1}^{m} |Y_{t+1} - F_{t+1}|/m \tag{3}$$

  where the term Y_{t+1} is the actual value at the time $(t + 1)$ and the term F_{t+1} is the forecast value in the same time.

On the selected forecasting models, we compute their accuracy and determine which model generates the best results.

In Table 1, the ARIMA c (3, 0, 4) model shows that the RMSE value is 9.92 on the training partition and 8.78 on the testing partition. The MAE value is 7.52 on the training set and 6.72 on the testing set. The deviation observed in errors (RMSE and MAE) between the training and testing sets is considerably low, making this model a possible candidate. The RMSE value is 6.62 on the training partition and 13.55 on the testing partition. On the other hand, for the TSLM model, the value of MAE is 4.45 on the training set and 9.79 on the testing set. The deviation seen in

Table 1 Performance analysis for the incident time series

Incident time series				
Models		ME	RMSE	MAE
ARIMA c (3,0,4)	Training set	0.01338605	9.921567	7.521685
	Testing set	0.56593183	8.784029	6.718977
TSLM	Training set	8.513036e–17	6.619009	4.488374
	Testing set	4.070144e + 00	13.546951	9.791207

errors (RMSE and MAE) for the testing set is almost twice that of the training set, but the errors (RMSE and MAE) observed on the training set of TSLM are of lower values than that of the ARIMA c (3, 0, 4) model. Comparing the two forecasting models, TSLM delivers the smallest RMSE value on the training data set. Therefore, the TSLM model also can be considered as a possible choice.

In Table 2, it can be seen for the ARIMA c (5, 0, 4) model that the value of RMSE is 32.51 on the training partition and 34.41 on the testing partition. The MAE value is 22.93 on the training set and 26.91 on the testing set. The deviation observed in errors (RMSE and MAE) between the training and testing sets is considerably low, making this model a possible candidate. As for the TSLM model, the RMSE value is 31.84 on the training partition and 43.74 on the testing partition. The value of MAE is 19.68 on the training set and 33.99 on the testing set. It is expected that the RMSE of the training data set generally decreases for both models. Comparing the two forecasting models, TSLM delivers the smallest RMSE value on the training data set. The deviation seen in errors (RMSE and MAE) for the testing set is almost 1.5 times that of the training set, but the errors (RMSE and MAE) observed on the training set are of lower values than that of the ARIMA c (5, 0, 4) model.

As can be seen from the above Tables 1 and 2, the values of RMSE are larger than MAE values that mean the more significant difference between them, the greater the variance in the individual errors for the incident and request time series. Similarly, the values of RMSE for the testing data set are higher than the training data set due to the use of all the data (training and testing) preceding each observation.

Table 2 Performance analysis for the request time series

Request time series				
Models		ME	RMSE	MAE
ARIMA c (5,0,4)	Training set	0.08231241	32.51304	22.93364
	Testing set	−23.41561705	34.40971	26.90801
TSLM	Training set	1.000000	31.83656	19.68006
	Testing set	−2.285124	43.73594	33.99587

5 Experiments with Real Data

The final approach to determine the best model will be by comparing the future forecasts from each (ARIMA and TSLM) and evaluating them according to how they match the real pattern observed in the original incident and request time series data. The actual data represent the existing demand created from 1 August 2019 to 31 July 2020, as depicted in Fig. 2. The forecasted results represent the expected demand to be created from 1 August 2020 to 31 July 2021.

5.1 Actual Data—Incident Time Series

In the following Figs. 3 and 4, we evaluate the future forecast using the selected forecasted models. For the ARIMA c (3, 0, 4) model, Fig. 3 presents the future forecast. It is clear from the figure that the ARIMA c (3, 0, 4) model future forecasts do not follow the pattern observed in the plot of the real data. The model provides a uniform distribution of the forecasted objects, which is not the case when dealing with incident tickets based on day to day operations. For that the ARIMA c (3, 0, 4) model is not the best choice to use on the incident time series data.

The same is depicted in Fig. 4 using the incident TSLM model.

It is clear from the plots that the incident TSLM model future forecasts pattern is similar to that observed in the plot (Fig. 2) of the real data. The model provides a changing distribution of the forecasted objects for each day matching the case when dealing with incident tickets based on day to day operations. For that the TSLM model is the most suitable to use on the incident time series data.

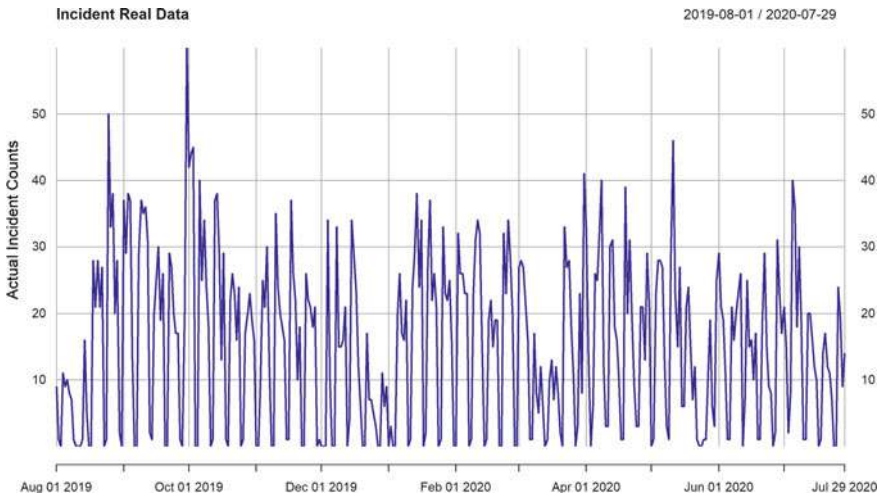


Fig. 2 Incident real data

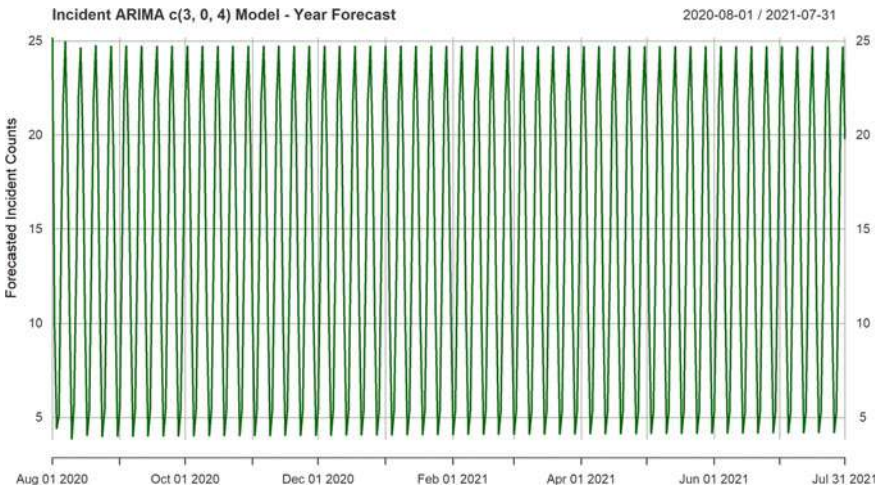


Fig. 3 ARIMA c (3, 0, 4) future forecast

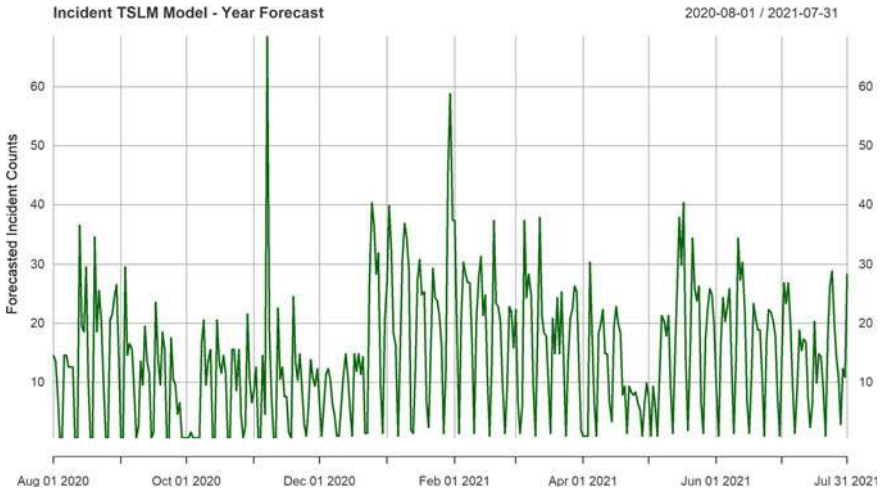


Fig. 4 Incident TSLM future forecast

5.2 Actual Data Request Time Series

The actual data represent the existing request demand created from 1 August 2019 to 31 July 2020, as depicted in Fig. 5.

To evaluate the selected forecasted models for request real data, the future forecasting methods were used as depicted in the Figs. 6 and 7. For the ARIMA c (5, 0, 4) model, Fig. 6 presents the future forecast.

Figure 7 presents the future forecast for the request TSLM model.

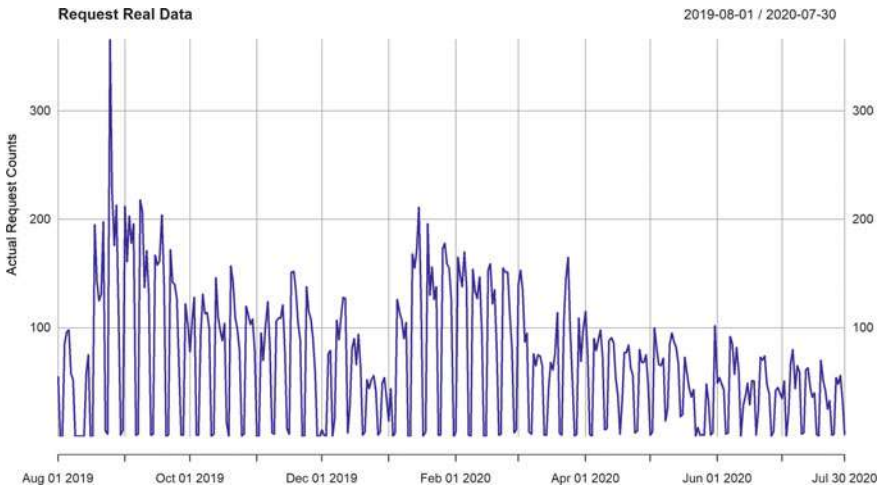


Fig. 5 Request real data

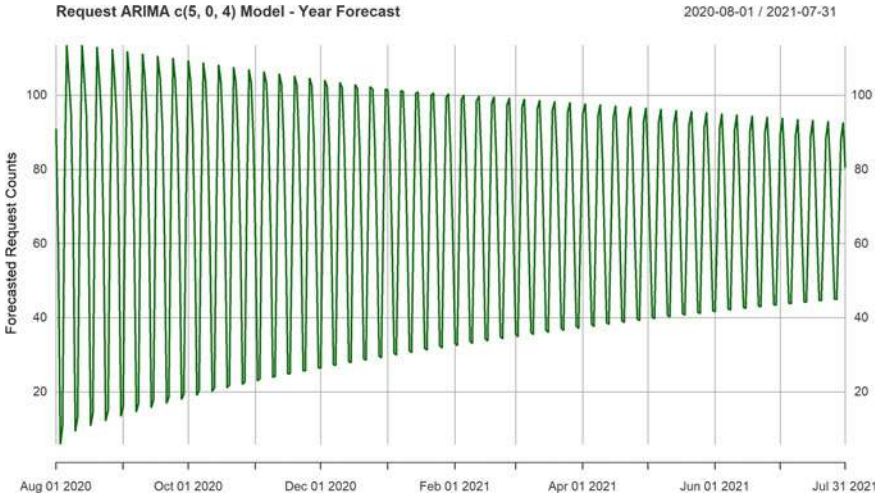


Fig. 6 ARIMA c (5, 0, 4) future forecast

Overall, these plots indicate that the request TSLM model future forecasts pattern is similar to that observed in the real data plot. The model provides a changing distribution of the forecasted objects for each day matching the case when dealing with request tickets based on day to day operations. This study has shown that the TSLM model is the most suitable to use on the request time series data.

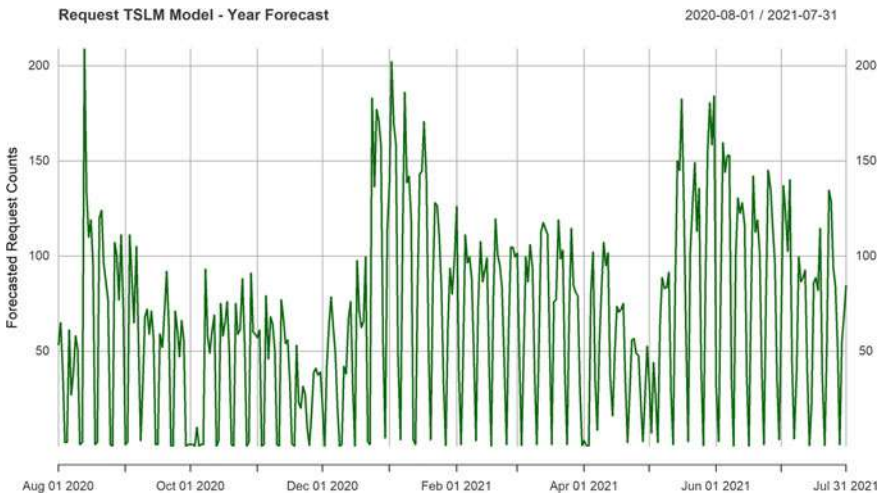


Fig. 7 Request TSLM future forecast

6 Conclusion and Future Work

This study extends our knowledge of forecasting models to build and evaluate the load expected on the incident management process and the request fulfilment process. Both of these two processes are part of the different IT service management processes but are more focussed on the delivery of IT services to the customers. ARIMA and TSLM are the two forecasting models used in this study. By comparing the forecasted results with the actual data from the testing partition, it becomes easier to filter out the models not suitable to work with each time series data set. The findings observed in this filter method are in alignment with visualisations findings which showed the time series behaviour. The accuracy parameters such as root mean squared error (RMSE) and mean absolute error (MAE) provide a comparison of the error rate for the remaining identified potential models (ARIMA and TSLM). For this study, the TSLM forecasting model is the most suitable for the IT service analytics.

Further research might explore the performance using support vector machine (SVM), neural networks in supplement to the methods described in this paper. It would be interesting to assess the analysis of the given IT service processes using different forecasting models.

References

1. Hertvik J (2020) IT risk management framework & process for ITSM environments, http://www.bmc.com/blogs/risk-management-framework-process-itsm-environments. Last accessed 2020/10/14

2. AXELOS ITIL. http://www.axelos.com. Last accessed 2020/11/05
3. Cobit 5: a business framework for the governance and management of enterprise IT, Illinois: ISACA (2019)
4. TM Forum. Business Process Framework (eTOM). http://www.tmforum.org. Last accessed 2020/11/18
5. Microsoft Operations Framework (MOF). http://www.docs.microsoft.com. Last accessed 2020/12/10
6. Chand S (2020) What are 26 ITIL processes and how they work? http://www.edureka.co/blog/itil-processes. Last accessed 2020/10/13
7. Cartlidge A (2013) Operational support and analysis ITIL intermediate capability handbook. TSO, London, United Kingdom
8. Anderson D (2013) Release, control and validation ITIL intermediate capability handbook. TSO, London, United Kingdom
9. Wright T (2014) Planning, protection and optimization ITIL intermediate capability handbook. TSO, London, United Kingdom
10. Ferreira DR, Silva M (2008) Using process mining for ITIL assessment: a case study with incident management. Technical University of Lisbon, Portugal
11. Goby N, Brandt T, Feuerriegel S, Neumann D (2016) Business intelligence for business processes: the case of IT incident management. Association for Information Systems
12. Delias P, Doumpos M, Matsatsinis N (2015) Business process analytics: a dedicated methodology through a case study. EURO J Decis Process 3(3):357–374
13. Clavon DM Jr (2017) A comparison of univariate modeling and forecasting techniques for predicting ITIL demand management workload up to six weeks ahead. The George Washington University, Washington, WA
14. Gerke K, Tamm G (2009) Continuous quality improvement of IT processes based of reference models and process mining. In: Proceedings of the fifteenth Americas conference on information systems, San Francisco, California
15. Gupta M, Asadullah A, Padmanabhuni S, Serebrenik A (2018) Reducing user input requests to improve IT support ticket resolution process. Empirical Softw Eng 23(3):1664–1703
16. Lin D, Raghu R, Ramamurthy V, Yu J, Radhakrishnan R, Fernandez J (2014) Unveiling clusters of events for alert and incident management in large-scale enterprise IT. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 1630–1639
17. Hyndman RJ, Athanasopoulos G (2018) Forecasting: principles and practice. Monash University, Australia

Chapter 5
Comparative Analysis of Color-Based Segmentation Methods Used for Smartphone Camera Captured Fingerphotos

Diwakar Agarwal and Deepak Mangal

1 Introduction

In biometric-based human identification, since long time, the fingerprint has secured the top position among various biometric traits due to its characteristic of uniqueness, acceptability, and permanence [1, 2]. Nowadays, massively, the fingerprint technology is deploying in law enforcement agencies, forensic, migration and border control, and several private and government organizations [2, 3]. However, touch-based fingerprint recognition system has suffered from various problems, e.g., presence of dirt and oil residue on the sensor plate, latent impression of the previous users, constraint acquisition environment, and signal distortion caused by the pressure applied on the sensor. These factors limit the large-scale deployability of touch-based automated fingerprint recognition systems. In view of overcoming these limitations, the touchless fingerprint recognition was first introduced by Song et al. [4] in 2004. Since then, number of works have been reported by the researchers on touchless to touch-based fingerprint matching algorithms.

The advantage of less-constraint touchless acquisition process paves a new path for using the touchless fingerprint technology in smartphone authentication. The usage in smartphones also becomes easier due to low-cost high-resolution camera smartphones widely available in the market. However, the fingerprint authentication process using mobile phone is more challenging in comparison to the sensor-based authentication process. The large degree of freedom of a mobile camera causes variation in image focus, rolling, and pitching of the finger. In addition to this, the lighting of a camera lens and the distance between lens and finger affects the contrast between

D. Agarwal (✉)
Department of Electronics and Communication Engineering, GLA University, Mathura, India
e-mail: diwakar.agarwal@gla.ac.in

D. Mangal
Department of Computer Engineering and Applications, GLA University, Mathura, India
e-mail: deepak.mangal@gla.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_5

ridges and valleys. Moreover, the complex background with varying illumination also poses a difficulty in touchless fingerprint recognition [5, 6]. These problems make the pre-processing as a crucial step in touchless fingerprint recognition. It has been observed that the pre-processing of an input fingerphoto is necessary before feature extraction and fingerprint matching [7]. The pre-processing includes finger segmentation or detection, RoI extraction, orientation correction, and fingerprint enhancement. The segmentation is holding the first step in the pre-processing with an objective to eliminate the background in a fingerphoto image. The approaches based on sharpness [5, 8, 9], shape [10], and color [11–17] are available for segmentation in the literature.

This paper aims to review and put the comparative analysis of color-based segmentation methods which mainly used three benchmark color models, namely YCbCr, CMYK, and CIELAB. The reason behind choosing the color-based approach is its robustness because it was investigated that the skin color of a finger remains consistent even if the background and illumination vary [17]. In this paper, the methods explored by Wang et al. [15], Malhotra et al. [17], and Wasnik et al. [16] are implemented on ISPFDv1 dataset [18] which contain fingerphotos captured in an uncontrolled environment. Although, in recent years, several works [6, 19] have presented the survey on pre-processing workflow of the touchless fingerprint recognition system, they are lacking in providing the distinguishing comparison between the methods used in various pre-processing steps. The main contribution of this paper is the quantitative comparison between three color-based segmentation methods in an unconstraint environment.

2 Literature Survey

Hiew et al. [11] put an emphasis on using the chrominance component of YCbCr color model for segmenting the finger and background. The probable fingerprint region was determined in CbCr plane in the absence of the luminous component. Birajadar et al. [14] segmented by using CbCr component of YCbCr and hue component of HSV color model. The binary mask with sharp finger boundary was generated by applying thresholding approach followed by binary morphological operations. Wang et al. [15] method is based on the study that shows fine details in a fingerprint are more observed under blue light illumination [20]. After converting RGB input image into YCbCr color space, Otsu's thresholding was applied on the Cb channel to generate a binary mask. Ravi and Sivanath [21] further investigated that an extension of the Cr component with HSV and nRGB color space provides promising result in finger detection. Malhotra et al. [17] introduced the use of magenta channel in CMYK color model for detecting the skin color. The Otsu's thresholding was applied on the magenta channel to generate a binary mask. Later, series of morphological operations was performed to reduce skin and non-skin error. Wasnik et al. [16] proposed the use of CIELAB color model for segmenting the finger from a fingerphoto. At first, RGB input image was converted into L*a*b color space. Then, k-means clustering was

applied on the chromaticity layer 'a*' and 'b*' to group the pixels into two regions, i.e., background and foreground. The color models other than above mentioned have also been adopted by some researchers. Lee et al. [9] partitioned skin and non-skin pixels in the rb plane of nRGB color model with the help of two-dimensional color histogram which is referred as lookup table (LT). This method provides remarkable results at the cost of computational complexity occurred due to modeling the color distribution of fingerprint images. Piuri and Scotti [22] merged the result of two different segmentations to get the final segmented finger. The first segmentation is based on the color model used similar in Lee et al. [9] and refined by applying morphological operations, while second one is based on ridge frequency estimation map. Raghavendra et al. [23] proposed scheme is the two-step segmentation process. In the first step, an input image was filtered in the spatial domain, and in the second step, the mean shift segmentation (MSS) was applied to get maximum segmented region.

3 Color-Based Segmentation

The skin color of humans is different due to different amount and kind of melanin present in the skin [20]. However, due to the absence of melanin in the palm, the palm color is not significantly different, and it is mainly affected by the absorption of oxygenated hemoglobin [9]. Therefore, even in an uncontrolled background environment, the fingers of all humans exhibit similar reflection rates [9]. Thus, the surface color of the finger appears as a useful component for the segmentation. The basic approach behind segmentation based on different color models is to use the color information that resides in corresponding color channels.

3.1 Segmentation Using YCbCr Color Model

Wang et al. [15] utilized YCbCr model to segment the finger in fingerphotos captured under the blue light illumination. In this paper, the Wang's method is implemented by following the steps summarized below as.

1. At first, a RGB input image is resized to 256×256 and transformed into YCbCr space according to

$$\begin{bmatrix} Y \\ Cb \\ Cr \\ 1 \end{bmatrix} = \begin{bmatrix} 0.29 & 0.58 & 0.11 & 0 \\ -0.16 & -0.33 & 0.50 & 0.50 \\ 0.50 & -0.41 & -0.08 & 0.50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \\ 1 \end{bmatrix} \tag{1}$$

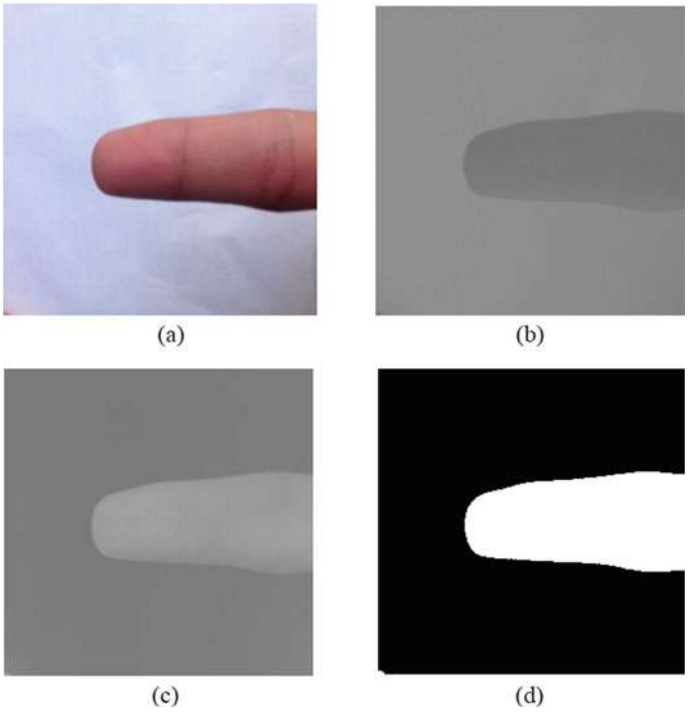


Fig. 1 Results after implementing the Wang's method **a** input RGB image with white outdoor scenario **b** Cb image **c** Cr image **d** binary mask

2. Then, a binary mask is obtained after applying the Otsu's method of global thresholding on the Cr component which is related to the skin tone. The main reason of not considering the Cb component in the implementation is that the difference between background and finger in the Cr image (see Fig. 1c) is more noticeable in comparison to the Cb image as shown in Fig. 1b.

3.2 Segmentation Using CMYK Color Model

Malhotra et al. [17] utilized the magenta channel of CMYK color model as it preserves most of the skin color. In this paper, the Malhotra's paper is implemented with the following steps.

1. At first, a 256 × 256 RGB input image is transformed into CMYK color space.
2. A binary mask (see Fig. 2c) is obtained after applying the Otsu's global thresholding on the magenta image as shown in Fig. 2b.
3. The morphological opening operation is then performed twice on the binary mask in cascaded form by considering the square structuring element of width

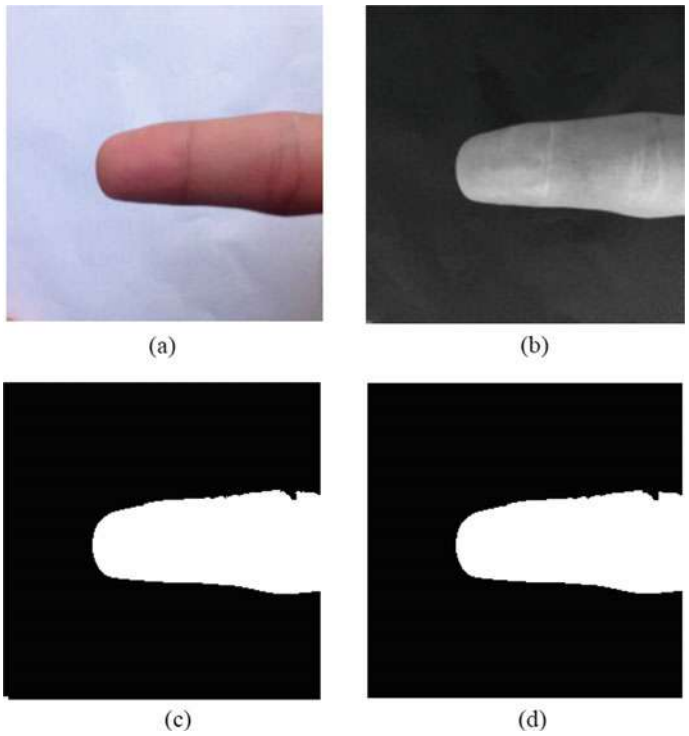


Fig. 2 Results after implementing the Malhotra's method **a** input image same as Fig. 1a **b** magenta image **c** binary mask after Otsu's thresholding **d** final segmented image after applying steps 3 and 4

three pixels. This operation eliminates false positive error (non-skin background seemed as skin foreground).

4. At last, the largest connected component is determined to eliminate the false negative error (skin foreground is seemed as non-skin background). The finger region is appeared as the large binary object as shown in Fig. 2d.

3.3 Segmentation Using CIELAB Color Model

Wasnik et al. [16] utilized CIELAB or CIE L*a*b color model for finger detection. In this method, k-means clustering was performed on the pixels holding 'a*' and 'b*' values to group the pixels into skin and non-skin regions. In this paper, the Wasnik's method is implemented with the following steps.

1. At first, a 256 × 256 RGB input image is transformed into CIE L*a*b color space.
2. Then, colors are classified in 'a*b*' space by applying the k-means clustering. The most important parameter in k-means clustering is the number of clusters.

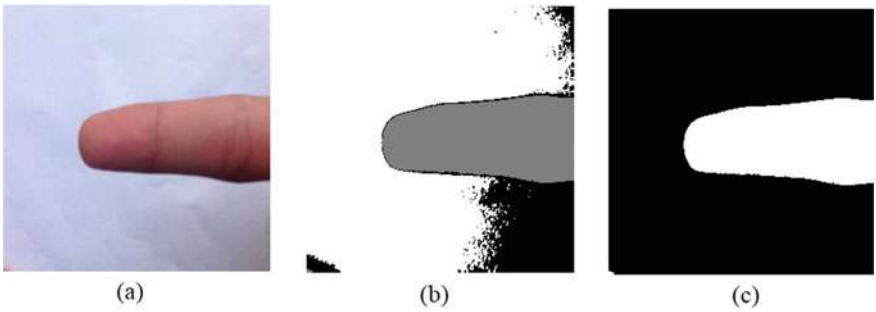


Fig. 3 **a** Results after implementing the Wasnik's method (**a**) input image same as Fig. 1a **b** image labeled by cluster indices **c** binary mask

Here, the number of clusters is empirically chosen as three. The clustering process is repeated three times to avoid local minima. Each pixel is assigned a cluster index either '1,' '2,' or '3' depending upon the distance from the clusters center. In Fig. 3b, the image is shown in which every pixel is labeled by the cluster index each represented by different shades.

3. At last, a binary mask is obtained by highlighting all pixels of the second cluster as shown in Fig. 3c. This cluster is chosen experimentally.

4 Experimental Results and Discussion

This section provides detail of the used database and also a performance comparison based on the amount of segmented proportion of the skin and non-skin regions between implemented segmentation methods.

4.1 Database

The ISPFDv1 dataset consists of 5120 images including 8 samples of each right index and right middle fingers of each 64 subjects. The total images are grouped into three sets corresponding to varying background, varying illumination, and live-scan. Table 1 provides more details of the database.

4.2 Comparative Results

The result of implemented finger segmentation algorithms is compared with the help of four performance metrics, namely correct (C), skin error (SE), non-skin error (NSE), and skin correct (SC). The ground truth which is necessary for evaluation is

Table 1 Details of ISPFDv1 dataset

Set	Sub-set	Number of images	Size	Resolution
Varying background	Natural indoor	64 subjects × 2 fingers × 8 samples = 1024	3264 × 2448	72 dpi
	White indoor	64 subjects × 2 fingers × 8 samples = 1024		
Set	Sub-set	Number of images	Size	Resolution
Varying illumination	Natural outdoor	64 subjects × 2 fingers × 8 samples = 1024	3264 × 2448	72 dpi
	White outdoor	64 subjects × 2 fingers × 8 samples = 1024		
Live-scan		1024	544 × 352	**96 pi**

obtained by marking skin and non-skin regions in images of varying background and varying illumination sets manually by hand. The C is defined as the percentage of total image pixels identified correctly as skin and non-skin pixels. The SE is defined as the proportion of skin pixels identified as non-skin pixels. The NSE is defined as the proportion of non-skin pixels identified as skin pixels. The S is defined as the proportion of correctly identified skin pixels out of total identified skin pixels. Tables 2, 3, 4, and 5 provide the values of performance metrics computed after implementing the above three mentioned methods in four background scenarios, namely natural indoor, natural outdoor, white indoor, and white outdoor. The graphical analysis of the same is also shown in Fig. 4.

Table 2 Comparison of three methods in natural indoor scenario

Metric (%)	Wang et al. [15] YCbCr	Malhotra et al. [17] CMYK	Wasnik et al. [16] CIELAB
C	**88.04**	61.67	58.16
SE	**3.27**	7.14	11.72
NSE	**8.68**	31.17	30.1
SC	**66.26**	31.83	29.96

Table 3 Comparison of three methods in natural outdoor scenario

Metric (%)	Wang et al. [15] YCbCr	Malhotra et al. [17] CMYK	Wasnik et al. [16] CIELAB
C	**87.82**	86.57	54
SE	**4.79**	7.31	13.44
NSE	7.37	**6.11**	32.54
SC	70.34	**70.69**	21.92

Table 4 Comparison of three methods in white indoor scenario

Metric (%)	Wang et al. [15] YCbCr	Malhotra et al. [17] CMYK	Wasnik et al. [16] CIELAB
C	**82.86**	80.39	66.85
SE	**6.39**	6.45	9.49
NSE	**10.74**	13.14	23.65
SC	**56.36**	50.8	42.39

Table 5 Comparison of three methods in white outdoor scenario

Metric (%)	Wang et al. [15] YCbCr	Malhotra et al. [17] CMYK	Wasnik et al. [16] CIELAB
C	**88.46**	81.03	68.47
SE	**4.36**	5.93	8.76
NSE	**7.17**	13.02	22.75
SC	**68.67**	52.9	50.48

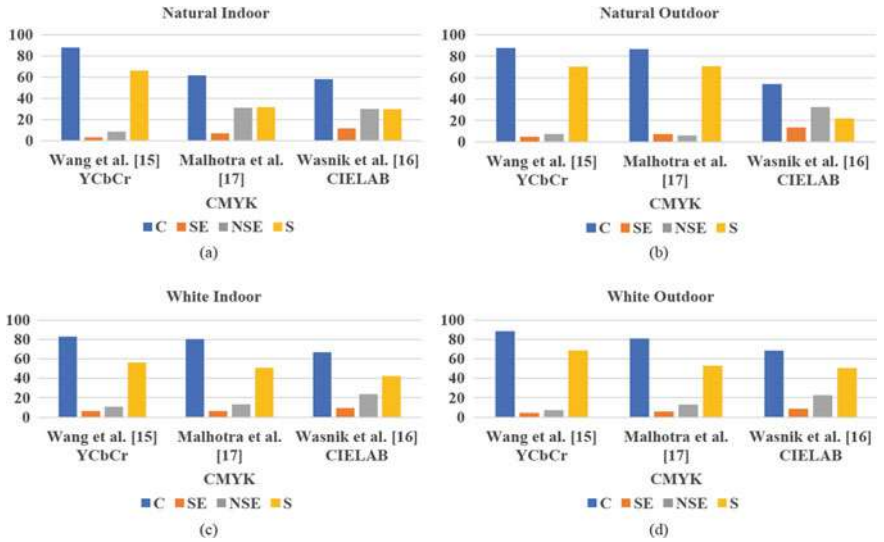


Fig. 4 Graphical representation of the comparison in natural indoor, natural outdoor, white indoor, and white outdoor scenarios

4.3 Discussion

It is desirable to be high C and SC and low SE and NSE for a good segmentation algorithm. It is observed from the experimental results that the Wang's method achieves desirable performance (see values highlighted in bold in Tables 2, 3, and

5) in natural indoor, white indoor, and white outdoor scenarios in comparison to other two methods. The Wang's method achieves high C and low SE also in natural outdoor, but achieves high NSE and low SC in comparison to the Malhotra's method. However, the difference is not so high. The NSE obtained in the Wang's method is only 1.26% higher, and the SC is only 0.35% lower than the Malhotra's method. The reason behind the performance of the Wang's method is the utilization of the Cr component which contains color information more than the chrominance component present in CMYK and CIELAB color models. The light red and pinkish red color which is generally found on the palm of human beings can be easily detected by the Cr component. It is also observed that the Wasnik's method shows less performance. This method is dependent upon the number of clusters requires in k-means clustering as an input parameter. The number of clusters should be equivalent to the number of colors seen in an input image otherwise, either number of segmentable regions will be less or unnecessary more segmentations will be taking place.

5 Conclusion

In this paper, a comprehensive review of the color-based finger segmentation methods is presented. Three segmentation methods using YCbCr, CMYK, and CIELAB color models are implemented on the ISPFDv1 dataset that contains fingerphotos captured in varying illumination and background. The performance is evaluated on the basis of number of pixels correctly identified in skin and non-skin regions. The methods are compared and observed that the finger segmentation based on YCbCr color model is well suited for complex background. This paper will aid the researchers in selecting the correct finger detection approach to be used in touchless fingerprint recognition.

References

1. Jain AK, Flynn P, Ross AA (2008) Handbook of biometrics, 1st edn. Springer, Boston
2. Maltoni D, Maio D, Jain AK, Prabhakar S (2009) Fingerprint analysis and representation. In: Handbook of fingerprint recognition, 2nd edn. Springer, London pp 97–166
3. Labati RD, Scotti F (2011) Fingerprint. In: Encyclopedia of cryptography and security. Springer US, Boston pp 460–465
4. Song Y, Lee C, Kim J (2004) A new scheme for touchless fingerprint recognition system. In: Proceedings of 2004 international symposium on intelligent signal processing and communication systems. IEEE, pp 524–527
5. Lee D, Choi K, Choi H, Kim J (2008) Recognizable image selection for fingerprint recognition with a mobile-device camera. IEEE Trans Syst Man Cybern Part B: Cybern 38(1):233–243
6. Khalil MS, Wan FK (2012) A review of fingerprint pre-processing using a mobile phone. In: 2012 international conference on wavelet analysis and pattern recognition. IEEE, pp 152–157
7. Hiew BY, Teoh AB, Ngo DC (2006) Preprocessing of fingerprint images captured with a digital camera. In: 9th international conference on control, automation, robotics and vision, pp 1–6

8. Lee D, Jang W, Park D, Kim SJ, Kim J (2005) A real-time image selection algorithm: fingerprint recognition using mobile devices with embedded camera. In: Fourth workshop on automatic identification advanced technologies (AutoID). IEEE, New York, pp 166–170
9. Lee C, Lee S, Kim J, Kim SJ (2005) Preprocessing of a fingerprint image captured with a mobile camera. In: Advances in biometrics. Springer, Berlin, pp 348–355
10. Jonietz C, Monari E, Widak H, Qu C (2015) Towards mobile and touchless fingerprint verification. In: 12th international conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, New York, pp 1–6
11. Hiew BY, Teoh ABJ, Ngo DCL (2006) Automatic digital camera based fingerprint image preprocessing. In: International conference on Computer Graphics, Imaging and Visualisation (CGIV). IEEE, New York, pp 182–189
12. Cheddad A, Condell J, Curran K, Kevitt PM (2009) A new colour space for skin tone de-tection. In: 16th International Conference on Image Processing (ICIP). IEEE, New York, pp 497–500
13. Sisodia DS, Vandana T, Choudhary M (2017) A conglomerate technique for finger print recognition using phone camera captured images. In: International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). IEEE, New York, pp 2740–2746
14. Birajadar P, Gupta S, Shirvalkar P, Patidar V, Sharma U, Naik A, Gadre V (2016) Touch-less fingerphoto feature extraction, analysis and matching using monogenic wavelets. In: International Conference on Signal and Information Processing (IConSIP). IEEE, New York, pp 1–6
15. Wang K, Cui H, Cao Y, Xing X, Zhang R (2016) A preprocessing algorithm for touchless fingerprint images. In: Biometric recognition. Springer International Publishing, Cham, pp 224–234
16. Wasnik P, Raghavendra R, Stokkenes M, Raja K, Busch C (2018) Improved fingerphoto verification system using multi-scale second order local structures. In: International conference of the Biometrics Special Interest Group (BIOSIG). IEEE, New York, pp 1–5
17. Malhotra A, Sankaran A, Mittal A, Vatsa M, Singh R (2017) Fingerphoto authentication using smartphone camera captured under varying environmental conditions. In: Marsico M, Nappi M, Proença H (eds) Human recognition in unconstrained environments, Chapter 6. Academic Press, New York, pp 119–144
18. Sankaran A, Malhotra A, Mittal A, Vatsa M, Singh R (2015) On smartphone camera based fingerphoto authentication. In: Proceedings of IEEE international conference on biometrics: theory, applications and systems. IEEE, pp 1–7
19. Priesnitz J, Rathgeb C, Buchmann N, Busch C, Margraf M (2021) An overview of touchless 2D fingerprint recognition. EURASIP J Image Video Process 2021(1):1–28
20. Angelopoulou E (2001) Understanding the color of human skin. In: Human vision and electronic imaging VI, vol 4299. International Society for Optics and Photonics, pp 243–251
21. Ravi H, Sivanath SK (2013) A novel method for touch-less finger print authentication. In: International Conference on Technologies for Homeland Security (HST). IEEE, New York, pp 147–153
22. Piuri V, Scotti F (2008) Fingerprint biometrics via low-cost sensors and webcams. In: 2008 IEEE second international conference on biometrics: theory, applications and systems. IEEE, pp 1–6
23. Raghavendra R, Raja KB, Surbiryala J, Busch C (2014) A low-cost multimodal biometric sensor to capture finger vein and fingerprint. Int Jt Conf Biom 1–7 (IEEE)

Chapter 6
Prediction of Heart Disease Through KNN, Random Forest, and Decision Tree Classifier Using K-Fold Cross-Validation

Meenu Bhagat and Brijesh Bakariya

1 Introduction

In living organisms, the heart plays an important function. Prediction and diagnosis of cardiovascular diseases need greater precision, perfection, and accuracy because even a minor error will result in fatigue or death. There are multiple death cases related to the heart problem, and the number is growing rapidly day by day. To deal with the issue, a prediction system is needed. Today, the majority of patients die only because their disease is discovered at an advanced stage due to instrument in accuracy, so there is a need to learn about more effective algorithms for disease prediction.

2 Literature Survey

Machine learning is a highly effective technology that is built on training and testing. It is a subset of artificial intelligence (AI), which is a wide field of research. Gavhane et al. [1] studied the multilayer perceptron model for prediction of heart disease and accuracy of the algorithm using CAD technology, and he found that if more people use the prediction method to predict their diseases, disease consciousness will rise, and the death rate of heart patients will decrease. Kohli et al. [2] studied diabetes prediction using SVM, heart disease prediction using logistic regression, and prediction of breast cancer using AdaBoot classifier and found that SVM has an accuracy

M. Bhagat (✉) · B. Bakariya
Department of Computer Science & Engineering, I.K. Gujral Punjab Technical University, Punjab, India
e-mail: meenubhagat@yahoo.com

B. Bakariya
e-mail: dr.brijeshbakariya@ptu.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_6

of 85.71%; logistic regression has an accuracy of 87.1%, and AdaBoot classifier has an accuracy of 98.57%. Train test split [3] is a common technique in which the original dataset is divided into two parts: train set and the test set. In cross-validation, a dataset D is evenly partitioned into n disjoint subsets in a standard (n-fold) cross-validation process. The first n folds of a dataset are used to train the classifier, and the remaining $(n - 1)$ folds are used to evaluate the classifier. U. Gupta and Deepak Gupta proposed an algorithm using RHN-TSVR on artificial datasets and real-world datasets with a different level of noise, and the model performed better is prediction ability when compared with other models [4]. They also worked on a Lagrangian twin-bounded support vector machine based on the L2-norm [5] and a fuzzy least square twin-bounded support vector machine based on kernel target alignment [6]. Kohavi [7] compared number of accuracy estimation techniques and discovered that cross-validation outperforms the others, and that further, stratification improves efficiency by reducing bias and variance.

The following is the format of this paper: The first Sect. 1 provides an overview of heart disease and related research. The data preprocessing steps are discussed in Sect. 2. Section 3 presents a brief overview of the machine learning algorithms used, whereas Sect. 4 focuses on the findings of the analysis. The conclusion is presented in Sect. 5.

3 Data Preprocessing

3.1 Data Collection

We have used an existing Cleveland heart disease dataset available at UCI [8]. There are 76 attributes and 303 instances in the dataset. Just 14 of the 76 attributes are tested (Table 1) despite the fact that they are critical to substantiate the efficiency of various algorithms. Before training the machine learning models, we discovered that we need to transform categorical variables into dummy variables and scale all of the values. We build dummy columns for categorical variables using the 'get dummies' method. Figure 1 is showing the general steps to be followed.

3.2 Data Balancing

Data balancing is essential for reliable results since we can see from the data balancing graph that both target groups are identical. The target groups are depicted in Fig. 2, with '0' representing patients with heart disease and '1' representing patients without heart disease.

Table 1 Heart disease dataset

Sr. no	Attribute name	Representative symbol	Details	Sr. no	Attribute name	Representative symbol	Description
1	Age	Age	Age of patient	8	Maximum heart rate	Thalch	Maximum heart rate
2	Sex	Sex	0 = female; 1 = male	9	Exercise-induced angina	Exang	Exercise-induced angina (0—no; 1—yes)
3	Chest pain	Cp	4 types of chest pain (1—typical angina; 2—atypical angina; 3—non-anginal pain; 4—asymptomatic)	10	ST depression	Old peak	ST depression induced by exercise relative to rest
4	Rest blood pressure	Trestbps	Resting systolic blood pressure (in mm Hg on admission to the hospital)	11	Slope	Slope	slope of the peak exercise ST segment (1—upsloping; 2—flat; 3—down sloping)
5	Serum cholesterol	Chol	Serum cholesterol in mg/dl	12	Number. of vessels	Ca	No. of major vessels (0–3) colored by fluoroscopy
6	Fasting blood sugar	Fbs	Blood sugar fasting > 120 mg/dl (0—false; 1—true)	13	Thalassemia	Thal	Defect types; 3—normal; 6—fixed defect; 7—reversible defect
7	Rest electrocardiograph	Restecg	0—normal; 1—having ST-T wave abnormality; 2—left ventricular hypertrophy	14	Target	Target	Diagnosis of heart disease status (0—No; 1—Yes);

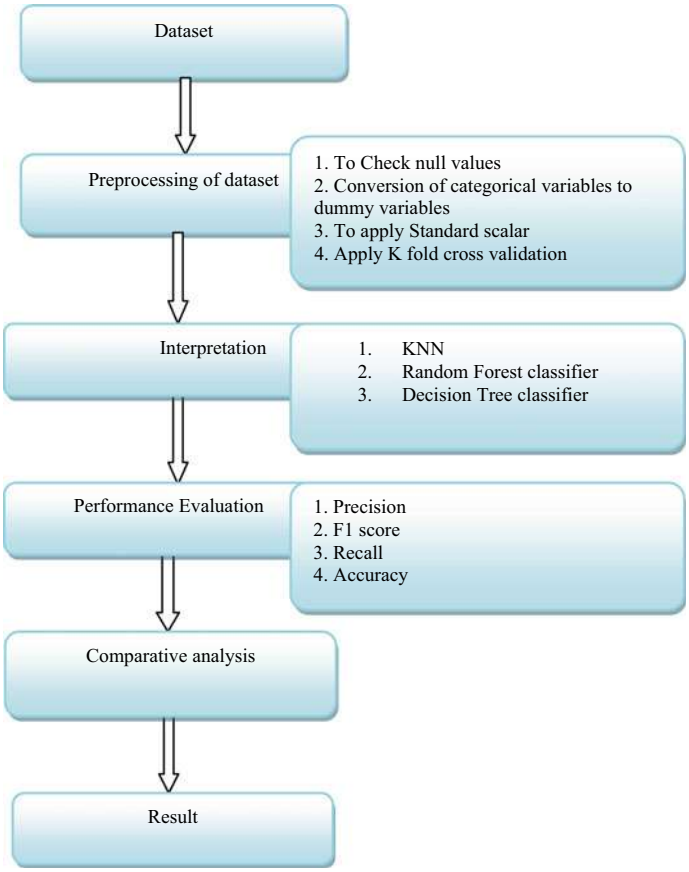


Fig. 1 General process

4 Algorithms

The K-nearest neighbors algorithm is a supervised classification approach. It classifies objects depending on how close they are to one another. It is a case-based learning scenario. The distance between an attribute and its neighbors is calculated using the Euclidean distance. It works with a set of named points to figure out how to mark another point.

A decision tree algorithm is used to predict response to data using classification or regression. When the features are grouped, classification is used, and regression is used when the data are continuous. The decision tree algorithm calculates the entropy of each characteristic first. The dataset is then sorted into groups based on the variables or predictors that have the most information gain or the least entropy.

Random forest is an extensively used supervised machine learning technique that can be applied to both regression and classification problems; however, it is more

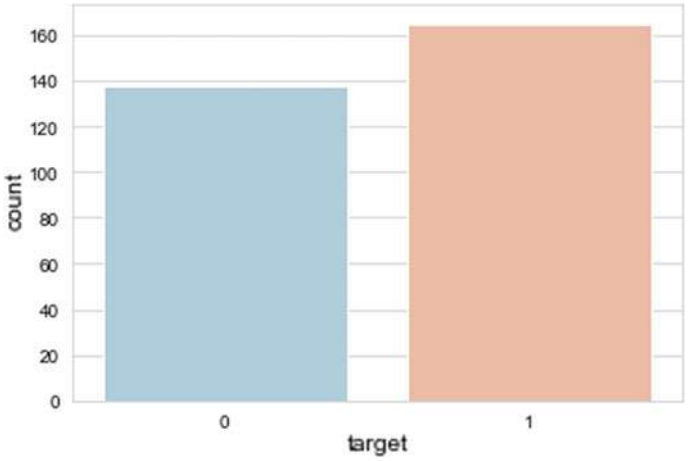


Fig. 2 Target group distribution

effective in the latter. The random forest classifier has achieved 96.1% and 97% accuracy using the Cleveland dataset and people dataset, respectively [9]. In [10], random forest has achieved an f-measure of 0.86. In [11], random forest is used to predict coronary heart disease, and it obtains an accuracy of 97.7%.

5 Results

The histogram shown in Fig. 3 displays the range of dataset attributes as well as the code that was used to generate it. The aim of this study is to determine if a patient may develop heart disease or not. On the UCI repository, this study was conducted on supervised machine learning classification techniques using k-nearest neighbor, decision tree, random forest classifier. The Jupyter Notebook was used to perform a number of experiments involving various classifier algorithms.

Model has been tested for different K values (Fig. 4). It has been analyzed that performance of the model is highest at $K = 12$, where K is number of neighbors.

The model has been tested for different cross-validation values ranging from 8 to 13 for $K = 12$ (Fig. 5), and it has been observed that model is giving best results at cv $= 10$ and $K = 12$ using K-nearest neighbor.

Figure 6 is showing the values of accuracies using decision tree classifier in different folds at cv $= 10$ and max_depth $= 3$.

Figure 7 is showing the values of accuracies in different folds using random tree classifier at cv $= 10$ and number of estimators $= 10$.

We discovered that the accuracy of the KNN is more than decision tree and random forest classifier algorithms for the same values of cross-validation. The results show that KNN is the best among them, with an accuracy of 85.06%, as shown in Fig. 8.

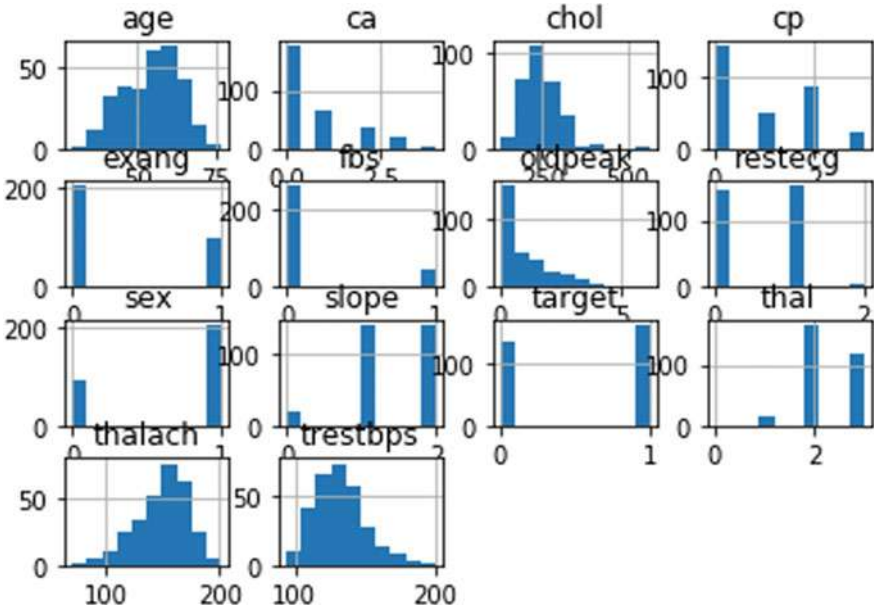


Fig. 3 Histogram

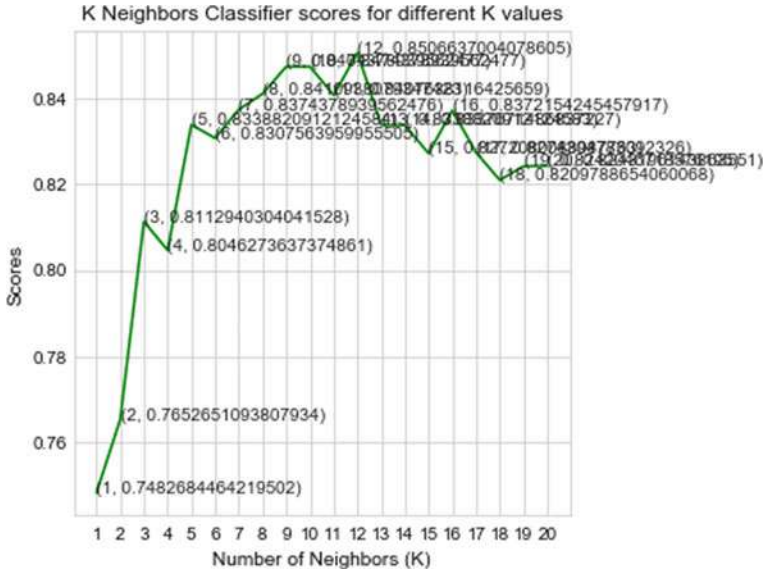


Fig. 4 Values of accuracies corresponding to different KNN values

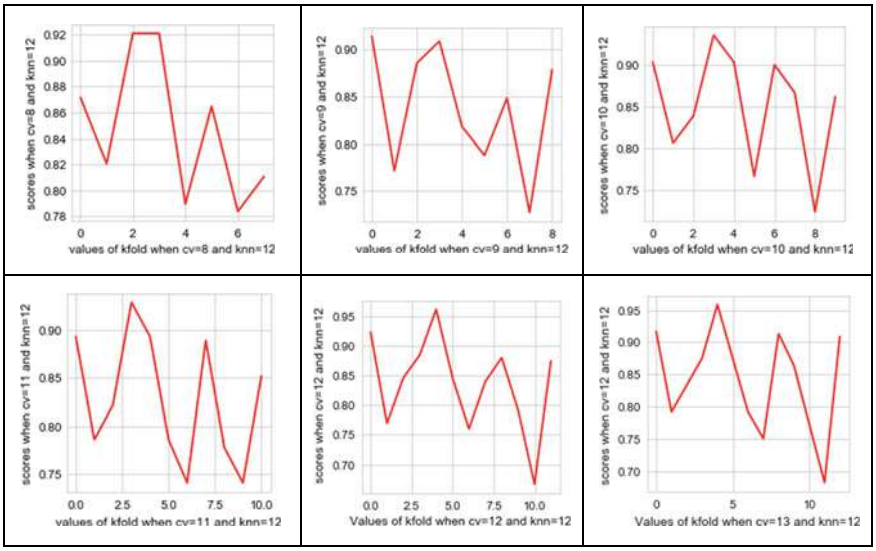


Fig. 5 Values of accuracies when cv = 8 to 13 and $K = 12$ in KNN

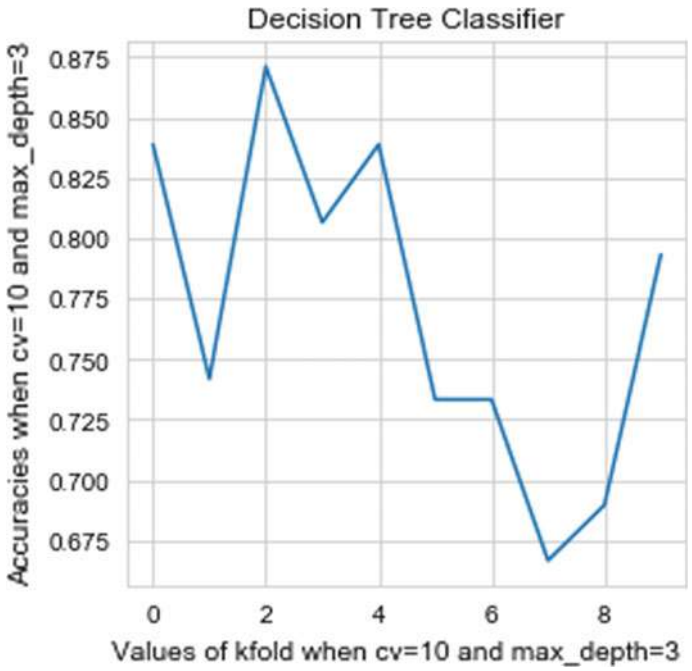


Fig. 6 Values of accuracies in decision tree classifier when cv = 10 and max_depth = 3

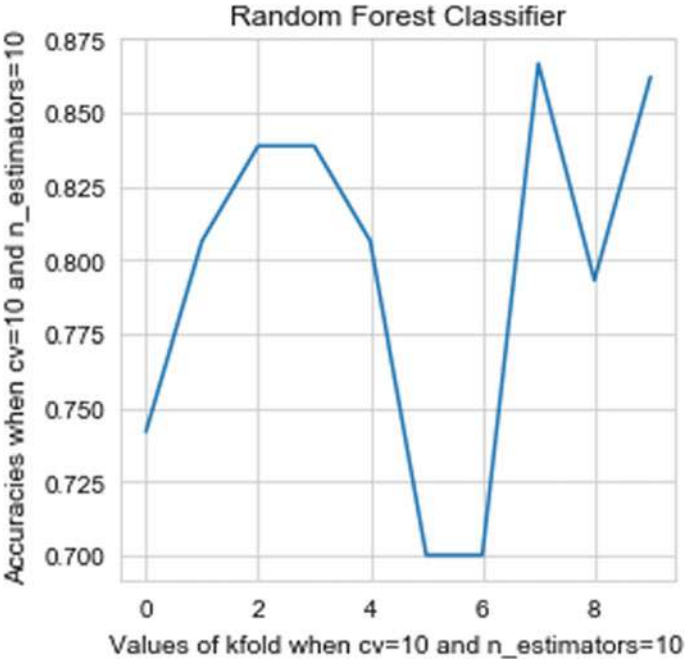


Fig. 7 Values of accuracies in random forest classifier when cv = 10 and number of estimators = 10

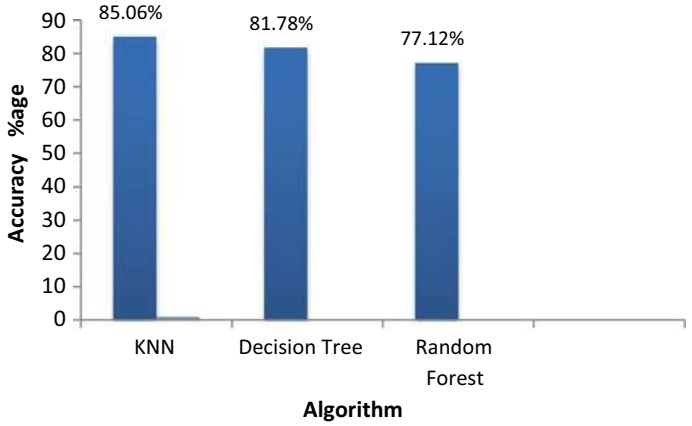


Fig. 8 Values of accuracies when cv = 10 using KNN, decision tree classifier, random forest classifier

6 Conclusions

One of the most important and crucial organs in the human body is the heart, and heart disease prediction is also a major concern for human beings. We focused on the heart disease dataset in this article. This research can also be applied to the prediction of other diseases. The accuracy of machine learning algorithms is determined by the dataset that was used for both training and research. We notice that KNN gives best accuracy results when we compare it with random forest classifier and decision tree while using K-fold cross-validation. In the future, more machine learning approaches will be used for the best analysis of heart diseases and for earlier disease prediction so that the rate of death cases can be reduced by disease knowledge.

References

1. Gavhane A, Kokkula G, Panday I, Devadkar K (2018) Prediction of heart disease using Machine Learning. In: Proceedings of the 2nd international conference on Electronics, Communication and Aerospace Technology (ICECA)
2. Kohli PS, Arora S (2018) Application of Machine Learning in diseases prediction. In: 4th International Conference on Computing Communication and Automation (ICCCA)
3. Zeng X, Martinez TR (2000) Distribution-balanced stratified cross-validation for accuracy estimation. J Exp Theor Artif Intell 12(1):1–12. https://doi.org/10.1080/095281300146272
4. Gupta U, Gupta D (2021) On regularization based twin support vector regression with huber loss. Neural Process Lett 53:459–515. https://doi.org/10.1007/s11063-020-10380-y
5. Gupta U, Gupta, D (2018) Lagrangian twin-bounded support vector machine based on L2-norm. In: Recent developments in machine learning and data analytics, vol 740. AISC. Springer, Singapore, pp 431–444
6. Gupta U, Gupta D, Prasad M (2018) Kernel target alignment based fuzzy least square twin bounded support vector machine. In: 2018 IEEE symposium series on computational intelligence (SSCI), IEEE
7. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp 1137–1143
8. https://archive.ics.uci.edu/ml/datasets/Heart+Disease. Accessed 02 May 2021
9. Xu S, Zhang Z, Wang D, Hu J, Duan X, Zhu T (2017) Cardiovascular risk prediction method based on CFS subset evaluation and random forest classification framework. In: 2017 IEEE 2nd international conference on big data analysis (ICBDA), IEEE, pp 228–232
10. Shahin A, Moudani W, Chakik F, Khalil M et al (2014) Data mining in healthcare information systems: case studies in Northern Lebanon. ISBN: 978-1-4799-3166-8 ©2014 IEEE
11. Mahboob T, Irfan R, Ghaffar B et al (2017) Evaluating ensemble prediction of coronary heart disease using receiver operating characteristics. 978-1-5090-4815-1/17/$31.00 ©2017 IEEE

Chapter 7
Distance Matrix Generation for Dynamic Vehicle Routing Optimization in Transport Fleets Management

Radosław Belka and Mateusz Godlewski

1 Introduction

Implementing effective solutions for automatic and quasi-optimal goods redistributions is one of the critical necessities signaled by the transportation market. The challenge belongs to both Vehicle Routing Problem (VRP) [1, 2] and WorkForce Management (WFM) [3, 4] issues. Many logistics problems are still not solved fully automatically and the effectiveness of tasks is related to the competence of the planners. Manually generated solutions are often based on a simple heuristic and typically use clustering of delivery points or the Largest-Truck First approach. The obtained results are usually correct and satisfying in the case of static requests and delivery points. Still, they may be far from optimal solutions, particularly in the case of frequent changes in the delivery points distribution. Meanwhile, it is easy to notice that in the case of a transport fleet of about 50–80 vehicles, consuming about 50,000 L of fuel, even a few percent of improvement in cost reduction is associated with significant financial gains. Therefore, improving management efficiency by implementing automated WFM problem optimization systems is still a significant challenge. In particular, the issue of WFM can be related to the proper and quasi-optimal assignment of resources (trucks) to the request (orders). The problem is finding an optimal set of routes for a fleet of vehicles to traverse to deliver to a given group of customers. Moreover, an acceptable solution should be found quickly, often within a few minutes. This problem can therefore be treated as a variant of the VRP problem. There are many different variants of the problem and methods for solving them [1, 2, 5]. In [6], a meta-analysis was carried out, where a total of 327 computational models were considered, and the following metaheuristic methods were mentioned: the Genetic Algorithm, Simulated Annealing (SA), Tabu Search (TS) Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO). Often

R. Belka (✉) · M. Godlewski
Kielce University of Technology, Al. 1000-lecia P.P.7, 25-314 Kielce, Poland
e-mail: r.belka@tu.kielce.pl

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_7

there are two-stage solutions, in which the first stage is based on a simple heuristic and gives an approximate solution, while the second one allows you to improve the obtained results. The challenge encountered belongs to both the Capacitated [7] and Split Delivery [8] variants of the VRP problem. Moreover, the variant considers the existence of different types of vehicles with different capacitance (heterogeneous or mixed fleet VRP).

One of the key data structures used in the VRP optimization is a special table, known as distance matrix [9], representing the generalized transport costs between each possible pair of delivery points. In the simplest and basic variant, the representation of this cost is the distance (in km). However, it can be equated with alternative quantities such as travel time, fuel consumption, and other more generalized parameters. The distance matrix (DMx) must present the actual distances between the delivery points, generating from their current positions. The DMx is a square matrix of values where the value DMx $[i, j]$ denotes the distance from destination number i (DestNo $= i$) to destination number j (DestNo $= j$). For N destinations, a matrix with dimensions $N \times N$ will be created. Due to the possibility of one-way routes, the matrix is not symmetrical. To create the DMx, a list of destinations containing the destination identifier and geolocation coordinates is necessary. Location coordinates as latitude and longitude are crucial for obtaining the cost values. The role of the DMx in transport issues was discussed, e.g., in work [10]. In [11], several different models for estimating the distance values based on the location of the destinations were presented. The transformation trend from the static DMx toward individual and dynamically personalized subset was shown in [12].

The paper [6] and the authors' experience indicate the growing importance of the Dynamic VRP variant, which considers dynamic request updates during the service provision. However, the concept of a request should be understood as the quantity and type of goods and the place of their destination. It means that the number and geolocation of destinations may change continuously, for example, in companies that operate portable toilets. In such a case, it may be necessary to re-plan routes, taking into account changes in the spatial distribution of the delivery points. Reasonable updating of the distance matrix may be a crucial factor influencing the effectiveness of the optimization algorithms in the variant of dynamically changing requests.

2 Methods and Data Sets

For the research and analysis, sets of delivery points were used, provided by one of the companies operating in the transport industry. In total, information on 357 potential delivery points was obtained, constituting a static list of destinations, within which a subset of 202 points (+ depot) corresponding to the current needs was selected. Geolocation data was obtained from the co-contractor of the TARGET project—ABC-Track, one of the GPS monitoring companies. Downloading the list of destinations was carried out dynamically thanks to a specially written script in

Python, using the HTTP protocol and GET queries. The developed API was integrated with the getPoint service of the ABC-Track database. The data was returned in JavaScript Object Notation (JSON) format and transferred to a specially created class of DataFrame objects of the Pandas library. The data included address information (street, town, postcode, etc.) and geographic coordinates (Latitude, Longitude), which were then used to generate a temporary distance matrix.

The distance matrix (DMx) was obtained based on a dynamic list of destinations obtained from the ABC-Track service, and then updated using the Mapping API services [13–16] offered by the free Open Source Routing Machine (OSRM) website [17]. A properly written script allowed aggregate queries, significantly reducing the update time.

The impact of updating DMx value on the optimization results was analyzed using the greedy and simple evolutionary 2-opt algorithm [18], both implemented in the Mathcad software. The set of orders, consisting of 1103 packages, was developed based on data obtained from an external shipping company, thanks to which the actual volumes, mass, and delivery points were known. The set of vehicles was generated based on a representative group of vehicles divided into six types with different capacities.

3 Results

3.1 Calculating the Temporary Distance Values (HavDMx)

It was found that updating distance values from the OSRM service can often be too long and therefore unacceptable. In the case of sending individual inquiries, the time to complete the distance matrix was ~5 h for considered 203 points. In the case of sending aggregated questions, it was possible to shorten this time to about 20 min, with this time being approximately linearly dependent on N. In the event of unfavorable connectivity or problems with the service overflow, this time may be extended. Therefore, an alternative method of generating provisional and temporal values should be proposed.

The primary step was to upgrade the list of destinations including their geographical coordinates. At this moment, the respective distance values are not yet known, but temporary values the related distances can firstly be estimated based on the quasi-Euclidean metric. To find the shortest route d_{hav} between two points located on a sphere with radius R, the "haversine" formula can be used (1).

$$d_{\text{hav}} = 2 \cdot R \cdot \arcsin\left(\sqrt{\text{hav}(\varphi 1 - \varphi 2) + \cos(\varphi 1) \cdot \cos(\varphi 2) \cdot \text{hav}(\lambda 1 - \lambda 2)}\right) \quad (1)$$

where
$\varphi 1$, $\varphi 2$—latitudes of destination points.
$\lambda 1$, $\lambda 2$—longitudes of destination points.

R = ~6371 km.

Used above the haversine function has the following form:

$$\text{hav}(\Theta) = \sin^2\left(\frac{\Theta}{2}\right) = \frac{1 - \cos(\Theta)}{2} \tag{2}$$

Figure 1 shows the fragment of the code responsible for generating the temporary HavDMx matrix.

3.2 Correcting the Distance Values

The values of the HavDMx matrix are highly underestimated as the routes connecting two points are usually not a straight line. The relative error can be calculated from the following formula (3).

$$\text{Err} = \frac{d_{\text{real}} - d_{\text{hav}}}{d_{\text{hav}}} \tag{3}$$

where

d_{real}—the actual distance.

d_{hav}—theoretical distance calculated from the dependence (1).

Fortunately, it is possible to obtain information about mutual distances thanks to external Distance Matrix APIs. The Distances Matrix API service allows the calculation of the distance between given points on the map. This service takes the coordinates of the given points and returns the travel distance in kilometers. One of the most popular is the Google API [16]. Nevertheless, the limitation of this service is the number of API requests that can be made within the free account—it is possible to obtain a travel distance between 40,000 pairs of points on the map, which allows creating a DMx with a maximum size of 200 × 200. In addition, the service is paid, and the current cost is estimated at around 5$ per 1 thousand queries. For about 200 destinations, the cost of the service is estimated at just over 200 USD. However, an attractive solution is free of charge service offered by the Open Source Routing Machine (OSRM) [17].

Using the OSRM service, obtaining much more actual mutual distances (denoted as RealDMx table) was possible. They were considered best-known values and identified as the d_{real} value in formula (3). Thanks to this, the relation between relative error and temporary (haversine) distance value was studied. Figure 2 shows the scatter plot relation, where the error values are presented on the horizontal axis and the theoretical distance value on the vertical axis. There was found, the range of the error value was from 0.02 up to >2.5. In other words, some values were underestimated more than three times. The distribution of error values is characterized by an evident skewness (mean value 0.296, median 0.260, standard deviation 0.1886 and

interquartile range 0.1832–0.3677). Preliminary analyzes indicate that estimating the actual distance values requires increasing the temporary, initial values by about 25–30%, which is confirmed by the works of other authors [19–22]. It was also found that only ~0.4% of the results refer to errors above 100%. On the other hand, the scatter plot suggests that for shorter distances (0–10 km), the corrections may be more significant than for the longer distances.

```
# CONSTANTS
BASE_URL = "https://system.abctrack.pl/abcTrack/action/api"
# unique key required for authorization
UNIQUE_KEY = "/key/6*************************8"
# approximate radius of earth in kilometers
R = 6373.0

# INTERDISTANCES MATRIX

def calculate_distance_between(lat1: float, lon1: float, lat2: float, lon2: float) -> float:
    """Returns distance between two geographical coordinates."""
    lat1 = radians(lat1)
    lon1 = radians(lon1)
    lat2 = radians(lat2)
    lon2 = radians(lon2)
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    distance = R * c
    return distance

def get_interdistances_matrix() -> pd.DataFrame:
    """Returns matrix of interdistances between destination points."""

    url = BASE_URL + UNIQUE_KEY + "/method/getPoints"
    results = requests.get(url=url).json().get("data").get("points")

    points = []
    for data in results:
        points.append(str(data["pointId"]))

    matrix = pd.DataFrame(index=points, columns=points)

    # filling interdistance matrix
    for point in results:
        p1 = str(point["pointId"])
        lat1 = float(point["pointLatitude"])
        lon1 = float(point["pointLongitude"])
        for point_ in results:
            p2 = str(point_["pointId"])
            lat2 = float(point_["pointLatitude"])
            lon2 = float(point_["pointLongitude"])
            matrix.at[p1, p2] = calculate_distance_between(
                lat1=lat1,
                lon1=lon1,
                lat2=lat2,
                lon2=lon2
            )

    return matrix
```

Fig. 1 Method of calculating the temporary distance values matrix—Python implementation

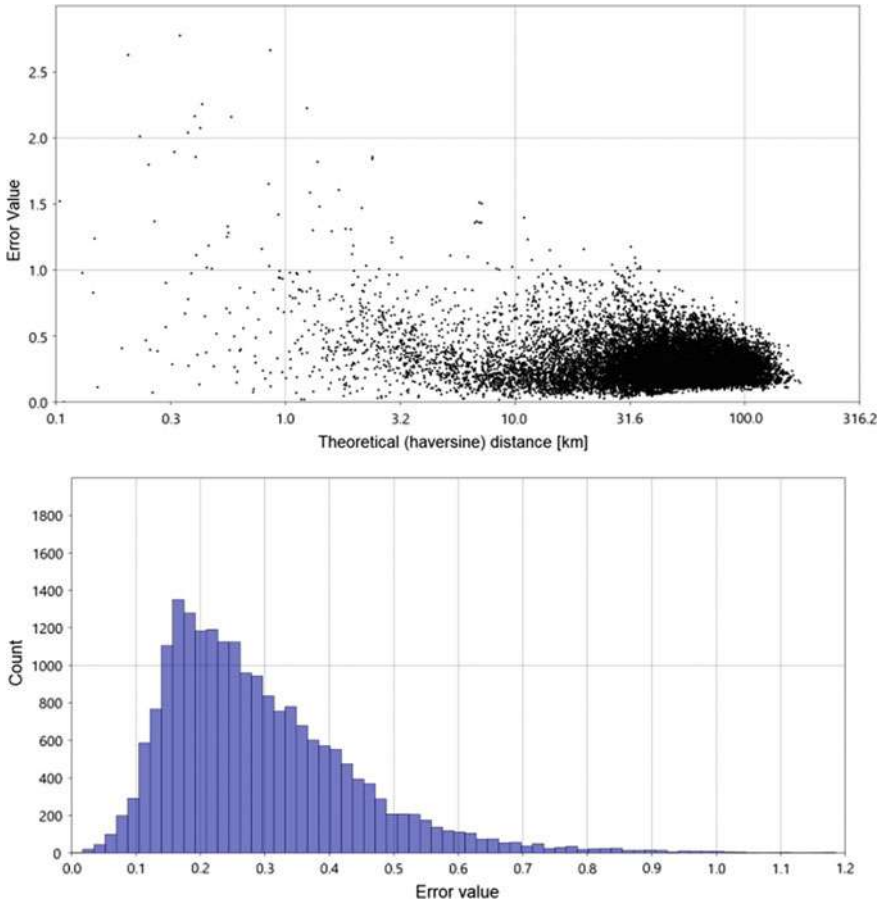


Fig. 2 Scatter plot of error-distance relation (up) and relative error distribution (down)

For the above reason, it was proposed to use a correction function, dependent on d_{hav}, which will play the role of an independent variable. On the other hand, the value of Err from formula (3) was proposed as the dependent variable. Next, it was hypothesized that for larger distances, the relationship $\text{Err}(d_{\text{hav}})$ may be linear or, in a particular case, constant. On the other hand, for smaller values, a non-linear term was introduced, represented by an exponential component. The final form of the function is as follows (4).

$$\text{Err}(d_{\text{hav}}) = a + b \cdot d_{\text{hav}} + c \cdot e^{-d \cdot d_{\text{hav}}} \tag{4}$$

The four (a–d) coefficients were estimated by the method of minimizing the mean square error. Approximation procedure has been made using a non-linear Generalized Reduced Gradient method [23], implemented in the Solver tool of Microsoft Excel

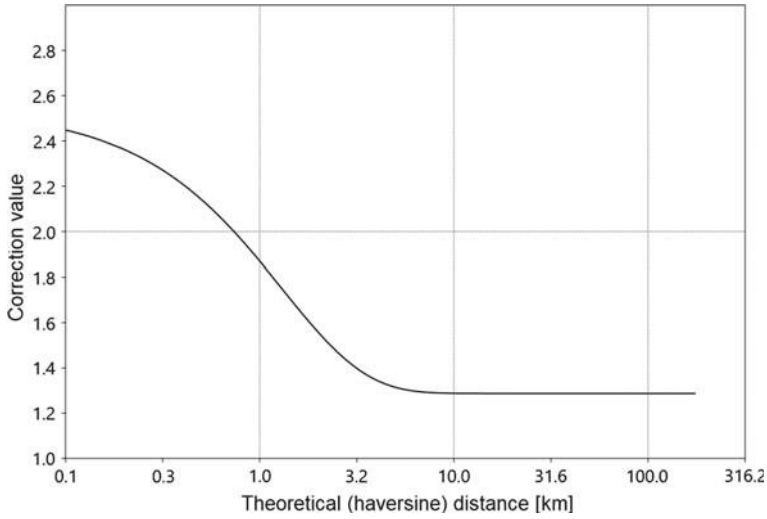


Fig. 3 Proposed distribution of correction function versus original haversine distance

software. As soon as the values of the *a–d* coefficients were calculated, it was possible to generate a correction function, as shown in Fig. 3.

As already suggested, longer distances should be adjusted by increasing their value by approximately 28.7%. On the other hand, for shorter distances (<10 km), the value of the correction coefficient increases significantly and should be greater than 2 for distances of several hundred meters. Determination of the corrected form of the distance matrix (CorrDMx) can then be done with a simple formula:

$$\text{CorrDMx}[i, j] = (1 + \text{Err}(\text{HavDMx}[i, j]) \cdot \text{HavDMx}[i, j]) \tag{5}$$

It was found that the arithmetic mean of the relative error between corresponding values of the RealDMx and CorrDMx matrices was <0.00001. The standard deviation (SD) was 0.1226 and was smaller than the deviation between RealDMx and HavDMx, where it was 0.1887. It was due to the introduction of the exponential component in the approximation function, as a result, the small values of the distance matrix were more significantly corrected. As mentioned before, the obtained results, especially for large distance values, are largely consistent with the proposed values recommended in the works [19–22]. On the other hand, this study suggests larger correction values for short distances (<3 km).

3.3 Dynamic DMx Update

According to the proposed concept, the DMx used in the optimization process can be generated based on the relationship 1, taking into account experimentally proposed correction factors—see Eq. (5) and Fig. 3. However, the obtained corrected matrix is still characterized by a fairly large discrepancy in relation to the empirical data—the average error was estimated at approx. ~12%.

For the generation and updating of the DMx for the purposes of the optimization algorithm, the following flow chart was proposed:

1. Get the actual List of Destination.
2. Calculate the HavDMx based on the haversine distance (1).
3. Correct the haversine distance according to formula (4)—create CorrDMx.
4. Make an additional list containing distances sorted in increasing order.
5. Send inquiries updating the distance values to the OSRM service, properly aggregated previously.
6. (Optional) Simultaneously send update queries to the database of routes included in the ABC-Track system. If the route is in the base (the distance is known), update the distance value.

The idea of generating a dynamic Distance Matrix is presented in Fig. 4.

As indicated—step 6 is optional and depends on the completeness of the databases in the GPS-based vehicle traffic monitoring system. Such systems have become popular in recent years and are widely used in the process of managing vehicle fleets. One of such systems is the above mentioned ABC-Track system. The system's resources were available thanks to the getHistory service, which provides information about the historical journeys of individual vehicles, which was used to obtain the history of the journey of a particular vehicle. In order to obtain historical data, the sent query had to be parameterized with three attributes:

- vehicleId—unique identifier of the truck. The identifiers of each vehicle were obtained using the website getVehiclesExtended.

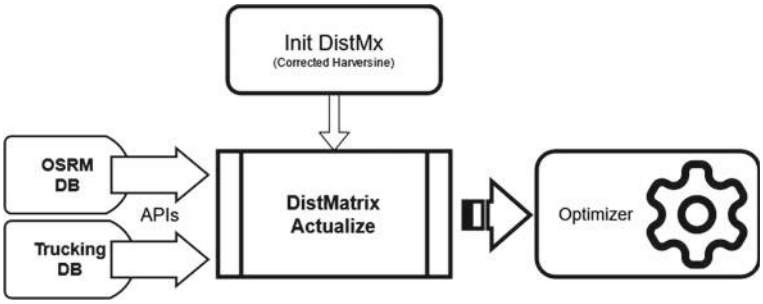


Fig. 4 Distance matrix actualization idea for dynamic VRP problem

- datetimeStart—Date (Year-Month-Day hh:mm:ss format in the UTC time zone) that marks the start of the period from which to retrieve historical trip data.
- datetimeStop—Date (Year-Month-Day hh:mm:ss format in the UTC time zone) that marks the finish of the period from which to retrieve historical trip data.

Single inquiries about the history of journeys of individual vehicles, whose identifiers were collected from the getVehiclesExtended service, are sent to the website, for three months (typical retention policy of database). The reply was a list of objects containing datetime, longitude, latitude, mileageKm (vehicle mileage in kilometers), as well as information such as heading (value read from the compass, which informs about the direction in which the vehicle is moving) and ignitionStatus (engine ignition status: 1—ignition on, 0—ignition off).

According to the assumption, the highest priority is assigned to routes and ABC-Track databases, then from the OSRM database, and the lowest for data from the updated CorrDMx. The ABC-Track distance has the most significant credibility because it considers actual driving conditions, restrictions imposed on trucks, and the resulting vehicle behavior on the route and allows for labeling drivers in the long run. The least reliable is the theoretical temporary distances stored in the CorrDMx. At the same time, the mean error is negligible, making it possible to accept these values as acceptable from the point of view of minimizing the global cost in optimizing the WFM problem.

3.4 Reordered Updating and Their Influence on Optimization Results

In the previous subsection, a representative flow chart was defined for the DMx value update process. In point 5, it was emphasized that updating data from OSRM or similar services might require proper query aggregation. By default, the queries are aggregated in the default order resulting from the structure of the destination list. Multiple updates can be obtained in a single aggregated query, especially if they cover one common point. In other words, the default update is to upgrade all data from one row or column in DMx. This approach is beneficial, e.g., when a new delivery point appears in the destination list and the appropriate distances should be completed. On the other hand, not all values in DMx are necessary. Unfortunately, at this stage it is not possible to determine which values the optimization algorithm is referring to, therefore it is advisable to update all values.

It was found that the type of DMx may significantly distort the reception of the effectiveness of optimization algorithms. Figure 5 shows a comparison of total routing costs for the considered problem of delivering 1103 parcels between 202 destinations. The values obtained from the manual solution, the greedy algorithm (fast but less effective) and the 2-opt algorithm (slow but more effective) are given. The total cost was close to ten thousand km. The total costs calculated using RealDMx (from OSRM service) were approx. 2% higher than the costs estimated on the basis

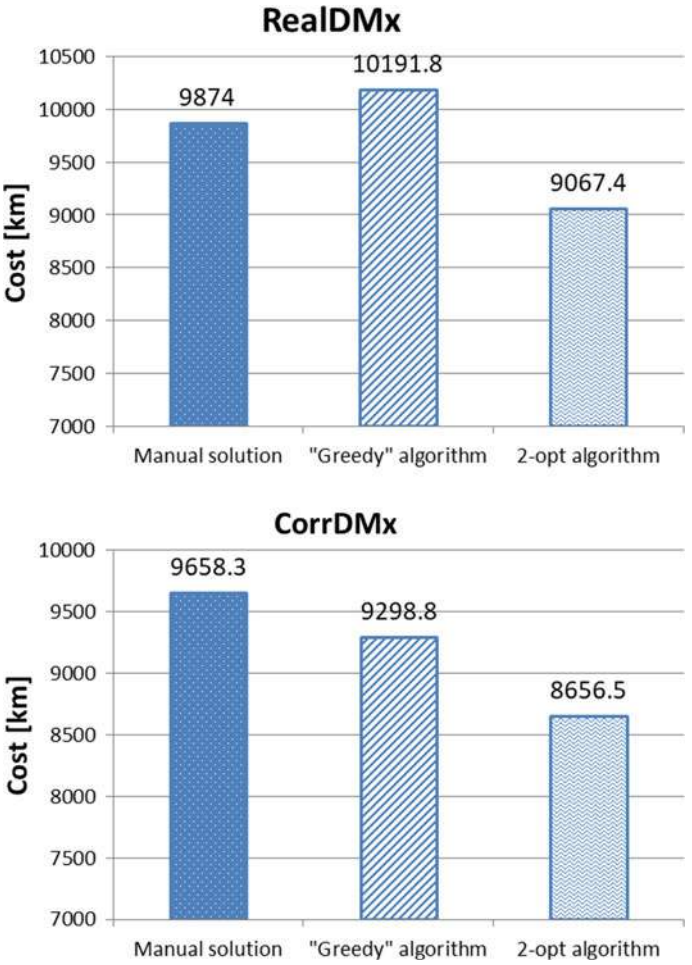


Fig. 5 Optimizer cost values versus type of DMx

of CorrDMx (without OSRM service). It is worth noting that the average value of the RealDMx matrix elements was only ~0.5% higher than the corresponding value for the CorrDMx matrix. Higher differences appear when comparing the costs of routes generated by the optimization algorithms—the difference is up to approx. 10% for "greedy" and more than 4% for 2-opt variants. The different assessment of the effectiveness of the algorithms is also observed. This mainly concerns the "greedy" algorithm, where for RealDMx it returns a solution by ~3.2% worse than the manual solution, while for CorrDMx the solution is ~3.7% better. Therefore, the lack of data updates may contribute to an overly optimistic assessment of the optimization effectiveness. Moreover, the above observations indicate the need for at least a partial update of the data.

Due to the potential time-consuming nature of a comprehensive DMx update, the possibility of partial data update (up to 50%) was considered in the next stage, and the impact of such an approach on the results returned by the optimization algorithms was analyzed. The proposed variants of reordering were based on two alternative hypotheses. In one of them, it is assumed that the largest values in the DMx table have the greatest impact on the total cost of the routing, therefore these values should be updated first. In an alternative version, the update should start with the smallest values because, as shown in Sect. 3.2, they are characterized by a significantly higher relative error. Additionally, it is assumed that routing algorithms will refer to small values a bit more often. For this reason, a total of ten partially updated matrices were calculated, divided into two datasets. In the first set, CorrDMx elements were partially actualized, beginning from the lowest values. They have been replaced with the corresponding values from the RealDMx matrix. In the second set, updating was realized starting from the highest values. Upload level was 10, 20, 30, 40 and 50% numbers of matrix's elements. Total cost reduction (in %) was calculated for any case. Results are presented in Fig. 6. In both cases, the evolution of the cost reduction value starts from the 10.37% value previously returned for the CorrDMx matrix. With the "from highest" approach, the trend seems to be quasi constant in the update range of the first 50% of the values and is not tending to the final value of ~8.17%. In the "from lowest" approach, the trend seems to be trending toward the final value. It allows us to defend the thesis about the validity of updating distances starting from smaller values.

The analyzes indicate that the hypothesis about the more significant influence of small distance values is more appropriate. This may be due to more frequent calls made by the optimization algorithm. This applies in particular to the greedy algorithm, the specificity of which assumes searching for the best local solution. Furthermore, assuming that the DMx matrix can only be updated with whole rows (columns), it is recommended to start the update process from those vortices (columns) with the lowest average value.

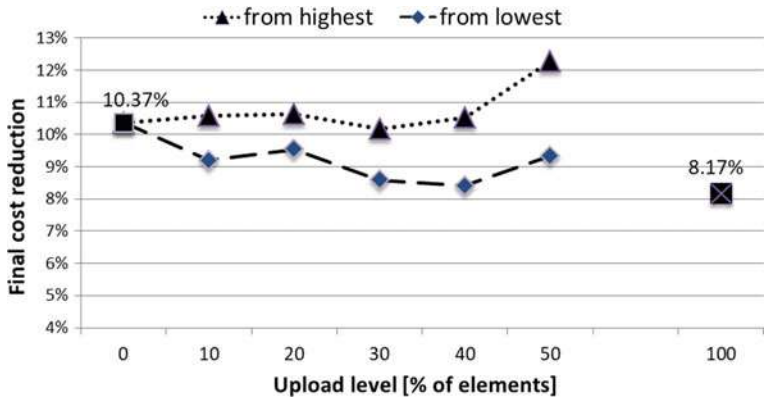


Fig. 6 Influence of partially DMx updating on cost reduction in 2-opt optimization algorithm

4 Conclusion

Optimization of the routing problem requires the actual values of elementary travel costs between two selected points. This cost can be obtained from free OSRM services, but the update time may be too long. In such a case, knowledge of provisional, approximate values would be advisable. This article shows that it can only be obtained by knowing the delivery point locations and applying a correction function. It has also been shown that the temporary values may, to some extent, affect the correctness of the generated routes. The differences are quite small, but they can introduce discrepancies in the total cost of a few percent.

The proposed solution assumes the possibility of only a partial update of distance values so that the obtained results are much closer to the real ones. As a result, it is not always necessary to submit a complete data set. This is especially important when the number and location of the destination are so frequently modified that a full update is difficult or impossible.

Acknowledgements This research was founded by the European Union's Smart Growth Operational Program 2014–2020, under grant agreement no POIR.04.01.04-00-0091/19-00.

References

1. Zhang H, Ge H, Yang J, Tong Y (2021) Review of vehicle routing problems: models, classification and solving algorithms. Arch Comput Methods Eng 1–27 (2021)
2. Konstantakopoulos GD, Gayialis SP, Kechagias EP (2020) Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. Oper Res Int J
3. Hota J, Debjani Ghosh D (2013) Workforce analytics approach: an emerging trend of workforce management. AIMS Int J Manage 7(3):167–179
4. Workforce management: how to optimize team productivity. https://asana.com/pl/resources/workforce-management. Last accessed 2021/09/05
5. Subramanian A, Uchoa E, Ochi LS (2013) A hybrid algorithm for a class of vehicle routing problem. Comput Oper Res 40(10):2519–2531
6. Braekers K, Ramaekers K, Van Nieuwenhuyse I (2016) The vehicle routing problem: state of the art classification and review. Comput Ind Eng 99:300–313
7. Zhang Q, Wei LR, Hu R, Yan R, Li LH, Zhu XN (2013) A review on the bin packing capacitated vehicle routing problem. Adv Mater Res 853:668–673
8. Archetti C, Speranza MG (2008) The split delivery vehicle routing problem: a survey. In: Golden B, Raghavan S, Wasil E (eds) The vehicle routing problem: latest advances and new challenges. Operations research/computer science interfaces 43. Springer, Boston, MA
9. Jambu M (1991) Classification and analysis of proximities data sets. In: Exploratory and multivariate data analysis, pp 407–418
10. Karner T, Weninger B, Schuster S, Fleck S, Kaminger I (2017) Improving road freight transport statistics by using a distance matrix. Austrian J Stat 46(2):65–80
11. Zgonc B, Metka Tekavčič M, Jakšič M (2019) The impact of distance on mode choice in freight transport. Eur Transp Res Rev 11 (Article number: 10)
12. Kuehnel N, Ziemke D, Moeckel R, Nagel K (2020) The end of travel time matrices: individual travel times in integrated land use/transport models. J Transp Geogr 88(102862):1–12

13. The Distance Matrix API for transport logistics and freight. https://distancematrix.ai/logistics-solutions. Last accessed 2021/09/05
14. TravelTime Distance Matrix API. https://traveltime.com/features/distance-matrix. Last accessed 2021/09/05
15. Time-distance matrix and three ways to use it. https://www.geoapify.com/time-distance-matrix-and-three-ways-to-use-it. Last accessed 2021/09/05
16. Google maps Platform. https://developers.google.com/maps/documentation/distance-matrix/overview. Last accessed 2021/09/05
17. Open Routing Source Machine—Modern C++ routing engine for shortest paths in road networks. http://project-osrm.org/. Last accessed 2021/09/05
18. Karagul K, Aydemir E, Tokat S (2016) Using 2-opt based evolution strategy for travelling salesman problem. Int J Optim Control Theor Appl 6(2):103–113
19. Kim, N. S., Van Wee, B. The relative importance of factors that influence the break-even distance of intermodal freight transport systems. J Transp Geogr, 19, 859–875. (2011).
20. Cooper JC (1983) The use of straight line distances in solutions to the vehicle scheduling problem. J Oper Res Soc 34:419
21. Barthélemy M (2011) Spatial networks. Phys Rep 499(1–3):1–101
22. Domínguez-Caamaño P, Benavides JAC, Prado JCP (2016) An improved methodology to determine the wiggle factor: an application for Spanish road transport. Brazilian J Oper Prod Manag 1:52
23. Facó JLD (1989) A generalized reduced gradient algorithm for solving large-scale discrete-time nonlinear optimal control problems. IFAC Proc 22:45–50

Chapter 8
Optimized Hysteresis Region Authenticated Handover for 5G HetNets

Vincent Omollo Nyangaresi, Zaid Ameen Abduljabbar, Mustafa A. Al Sibahee, Ayad Ibrahim, Ali Noah Yahya, Iman Qays Abduljaleel, and Enas Wahab Abood

1 Introduction

Cellular networks have continued to evolve, with the fifth generation (5G) promising massive connectivity, high bandwidths, extremely low latencies and high reliability [1, 2]. This has seen these networks being deployed in numerous Internet of things (IoT) scenarios such as remote surgery, smart homes and cities, intelligent transportation among others [3]. The 5G networks support multiple mobile heterogeneous networks (HetNets) that facilitate seamless connectivity for offering access to numerous data services. Due to vast number of devices supported and the need for the maintenance of high quality of service (QoS), mobility management is a challenging

V. O. Nyangaresi (✉)
Faculty of Biological and Physical Sciences, Tom Mboya University College, Homabay, Kenya
e-mail: vnyangaresi@tmuc.ac.ke

Z. A. Abduljabbar · A. Ibrahim
Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
e-mail: zaid.ameen@uobasrah.edu.iq

A. Ibrahim
e-mail: ayad.abdulsada@uobasrah.edu.iq

Z. A. Abduljabbar
Huazhong University of Science and Technology, Shenzhen Institute, Shenzhen, China

M. A. Al Sibahee
College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
e-mail: mustafa@sztu.edu.cn

Computer Technology Engineering Department, Iraq University College, Basrah, Iraq

A. N. Yahya
Science and Research Branch, Islamic Azad University, 1477893855 Tehran, Iran
e-mail: ali.noah@iuc.edu.iq

Department of Computer Technology Engineering, Iraq University College, Basra 61004, Iraq

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_9

task [4]. As pointed out in [5], the 5G ultra-dense networks bring forth challenges in radio resource allocations, handover cell selection, power management and mitigation of interference [5, 6]. In cell selection, a decision is made regarding the cell to which the user equipment (UE) should be handed-over to [7]. Due to the many QoS that need to be fulfilled and the many factors that need to be considered during the handover process, cell selection degenerates into a non-deterministic hard (NP-hard) optimization problem [2]. Here, the computational complexity exponentially surges as the network size increases [8].

The increasing subscriber demands in accessing a myriad of services renders handover decisions critical. These handovers should take into consideration network conditions and user preferences [4]. In HetNets, the UE has increased flexibility in the selection of radio technologies during handovers. This decision can be influenced by location and availability. As such, the UE needs to possess some intelligence so as to automatically choose the most optimal radio access technology. In this scenario, machine learning algorithms (MLs) such as neural networks come handy [9]. This is because each of the available radio access technology may have diverse specifications that offer different levels of QoS based on channel status and subscriber density. According to [10], the ability of artificial neural networks (ANNs) to produce precise results for some unseen inputs during the training process renders it applicable in cell selection.

However, as explained in [11], the design of vertical handovers in HetNets presents some challenges with regard to the enhancement of QoS which requires non-interruption of ongoing communications. Although numerous handover schemes have been presented in literature, seamless handovers among the HetNets cells remain a mirage [12]. As such, there are still heavy packet losses and high latencies during the handover process [13]. The main cause of this is the handover decision phase, and hence, there is need to address inefficient communication and poor QoS during handovers [14]. As explained in [15], the conventional handovers prioritize the received signal strength indicator (RSSI) as the main criteria in the selection of the target cell. However, reliance on RSSI is detrimental in 5G ultra-dense networks as it often leads to ping-pong handovers [13]. This requires the incorporation of machine learning algorithms for intelligent cells selection, reduction of processing time and computational complexity.

Apart from efficiency of the handover process, security and privacy are other challenges that require attention. According to [16], security and privacy issues in 5G networks center around UEs, access network and core network. The support of many use cases, services and devices in 5G networks introduce numerous attack

I. Q. Abduljaleel
Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basra, Iraq
e-mail: iman.abduljaleel@uobasrah.edu.iq

E. W. Abood
Department of Mathematics, College of Science, University of Basrah, Basra, Iraq
e-mail: enas.abood@uobasrah.edu.iq

vectors that may be employed to compromise other devices [17]. Authors in [18] identify transparency, privacy, decentralization, interoperability and security as key issues in 5G networks. Privacy is particularly crucial due to massive exchange of personal information among 5G-enabled IoT devices. This paper makes the following contributions:

- An optimized hysteresis region authenticated handover is developed for improved efficiency in 5G HetNets.
- A robust handover protocol is developed based on dynamic sequence numbers and timestamps to protect against replay attacks.
- BAN logic security evaluation shows that the proposed protocol establishes a session key between the UE and source gNB.
- It is shown through various lemmas and their proofs that this protocol offers mutual authentication, anonymity, untraceability, backward and forward key secrecy. In addition, it thwarts MitM, replay, privileged insider, spoofing and impersonation attacks.

The rest of this paper is structured as follows: Sect. 2 presents related work, while Sect. 3 elaborates the system model. On the other hand, Sect. 4 presents and discusses the results, while Sect. 5 concludes the paper and gives future work.

2 Related Work

Many schemes have been presented in literature to curb the numerous efficiency, security and privacy issues in 5G networks. For instance, authors in [11] have introduced an ANN-based handover decision protocol in HetNets while recurrent neural network (RNN) has been deployed in [19]. Here, RSSI is used to train the model, and the results show that this scheme has a 98% accuracy in target cell prediction. Similarly, authors in [20] have deployed ANN for handover decision within the hysteresis region. The main criteria used here is traffic intensity, and the scheme reduced number of executed handovers. On the other hand, a signal-to-interference noise ratio (SINR)-based machine learning handover protocol is introduced in [21] for target cell selection, yielding a 90% accuracy.

Using RSSI and ANN, a handover decision scheme is presented in [22], while a Q-learning algorithm has been deployed in [23] for handover decisions. The feed forward ANN algorithm has been introduced in [24] using UE locations as an input. A machine learning scheme based on hidden Markov model is developed in [25] for target cell selection. Similarly, an intelligent ML scheme has been presented in [2] for best cell selection. The scheme in [2] resulted in improved handover execution time and reduced complexity. On the other hand, a fuzzy logic (FL)-based protocol is introduced in [26] for seamless handovers. However, this scheme failed to incorporate critical network parameters such as SINR and transmission rate.

Similarly, FL-based scheme is developed in [27] while authors in [28] have deployed ANN for handover decisions. Although the scheme in [28] enhanced efficiency, this protocol has high complexity. On the other hand, a blockchain (BC)-based handover protocol is developed in [29] for software defined networking (SDN) environment. However, the utilization of BC leads to high storage and computation complexities [30].

All the above schemes address efficiency and cell selection issues during the handover process but ignore security and privacy issues. During 5G handovers, the third generation partnership group (3GPP) has specified authentication and key agreement (5G-AKA) and extensible authentication protocol–improved AKA (EAP-AKA') in its Release 16(3GPP R16). However, these AKA protocols are still vulnerable to attacks such as denial of service (DoS), impersonation and man-in-the-middle (MitM) [31]. To address some of these issues, group-based schemes have been presented in [32–35]. However, existence of malicious group members that may compromise the communications, high communication overheads and the group leader presenting a single point of failure are some of the issues in these protocols [31]. The bilinear pairing (BP)-based handover authentication technique introduced in [36] has increased computation and communication costs due to extensive BP operations [37]. Similarly, the handover authentication scheme in [38] has relatively high computation and communication overheads.

In summary, efficiency, security and privacy are very elusive issues in 5G networks as none of the schemes above effectively addresses this trio. Efficient handovers assures higher data rates and effective utilization of the network resources [39]. In addition, there is need for an authentication protocol that has very little communication and computation costs so as to be energy efficient in terms of power consumptions [40]. This is particularly important for the resource-constrained IoT devices that are extensively supported by 5G networks.

3 System Model

A review of the current ML-based target cell selection algorithms has shown that they fail to incorporate sufficient parameters as inputs to the prediction models. The focus is normally paid to network level parameters such as RSSI, ignoring user level, device features, service requirements and application level parameters. As such, the selected target cells quite often fail to offer the required QoS levels and results in ping-pong handovers. In addition, the conventional ML-based schemes fail to incorporate authentication phases in their architectures. As such, there is need for an intelligent handover protocol that not only boosts efficiency but also authenticates the communicating entities during the handover process. This section presents the mathematical preliminaries, handover optimization and the authentication process as discussed below.

3.1 Mathematical Preliminaries

This sub-section provides some mathematical basis for the deployed artificial neural network. This is elaborated using mathematical relations (1)–(7) as derived below.

Taking A, B and C as the neurons in the input, hidden and output layers, respectively, the ANN model is built using the log-sigmoid transfer function depicted in (1):

$$f(x) = \left(1 + \mathrm{e}^{-x}\right)^{-1} \tag{1}$$

To ensure constant regulation of the ANN weight values, the error function (EF) and error back propagation (BP) are deployed. In essence, the regulation of the ANN weights via the error feedback ensures that the offset value of EF is closer to the anticipated value. Mathematically, taking e_i as the anticipated values of the FOMs and $\mathbb{Q}_i$ as the corresponding output values computed by the ANN, EF is denoted as in (2):

$$\mathrm{EF} = \frac{\sum_i (\mathrm{e}_i + \mathbb{Q}_i)^2}{2} \tag{2}$$

In the proposed ANN model, the neurons as the input vector $\breve{I} = (\breve{I}_1, \breve{I}_2, \breve{I}_3, \ldots \breve{I}_n)$, and the corresponding weight values for $\breve{I}$ in the input neuron as $\underline{z} = (\underline{z}_1, \underline{z}_2, \underline{z}_3, \ldots \underline{z}_n)$. On the other hand, the network weights are set as $(\underline{z}_{ij}, \underline{h}_{ij})$, while the neuron threshold is taken as $\ddot{t}$. The activation function F of this model is given in (3):

$$f(x) = \begin{cases} 1, & \breve{I} \geq 0 \\ -1, & \breve{I} < 0 \end{cases} \tag{3}$$

Taking $\breve{I}_j$ as the jth input layer node, $\underline{K}_j$ as the jth hidden layer node, and $\underline{L}_j$ as the jth output layer node, the neural network output is expressed as in (4):

$$y = f\left(\sum_{i=1}^{n} \underline{Z}_i \breve{I}_i - \ddot{t}\right) \tag{4}$$

On the other hand, the hidden layer and output layer node outputs are given in (5) and (6), respectively:

$$\underline{K}_i = f\left(\sum\nolimits_j \underline{Z}_j \breve{I}_j - \ddot{t}_i\right) \tag{5}$$

In essence, (5) gives the activation function of the ith network, $f(i\text{th network})$.

$$\underline{L}_k = f\left(\sum_j \underline{h}_{ij} \breve{I}_j - \ddot{i}_k\right) \tag{6}$$

Similarly, (6) gives the activation function of the kth network, $f(k\text{th network})$. Using the values computed in (5) and (6), the output layer node error is represented as in (7):

$$EF = \frac{f\left(\sum_k (e_k - \underline{L}_k\right)}{2} \tag{7}$$

In essence, the objective of the back propagation neural network is to reduce *EF* during training and learning.

3.2 Handover Optimization

The execution of the proposed protocol is triggered whenever the UE is detected at the hysteresis region in which it can handover to any of the possible neighboring target gNBs (TgNBs). Here, each of these TgNBs constructs back propagation ANN in which the neuron weight for each layer is influenced by the theoretical values of the deployed figures of merit (FOMs). These FOMs included blocking probability, traffic intensity, power density, received carrier power and path loss. The rationale for the selection of these particular FOMs is explained in [13]. Whenever the UE enters the hysteresis region where the coverage areas of N TgNBs overlap, the actual values of these FOMs are collected and coupled into the trained ANN models. In these trained ANN models, the predicted value of the cell candidacy value (CCV) is computed, and the TgNB with the highest value of CCV is selected as the ideal target cell for the UE.

The tracking area was partitioned into three regions corresponding to logic low, medium and high as explained in [41]. Afterward, based on both random waypoint mobility and random direction mobility models [42], the UE moved through the tracking area as the required FOMs is measured and buffered [43]. Whenever the UE is within the hysteresis region, the ANN is deployed to optimize the hysteresis margin, after which the fuzzy logic (FL) helped identify the most ideal TgNB [44]. Detailed description of the operation of ANN and FL during the handover process can be found in [44]. Figure 1 gives the data flow in the proposed protocol.

As shown in Fig. 1, the AKA process begins by having the UE measure and buffer FOMs, after which the trained ANN model is loaded to offer FOMs predictions for the current as well as all probable TgNBs. Next, using 5G's maximum radio frequency coverage distance D of 248 m in accordance with the modified SUI model, the protocol determines whether the UE is within SgNB or not. If it is within SgNB, it continues to measure and buffer FOMs, otherwise if buffers the current FOMs in its handover decision table (HDT). Afterward, the trained ANN model evaluates the FOMs from SgNB and all possible TgNBs and their candidacy values (CVs) which

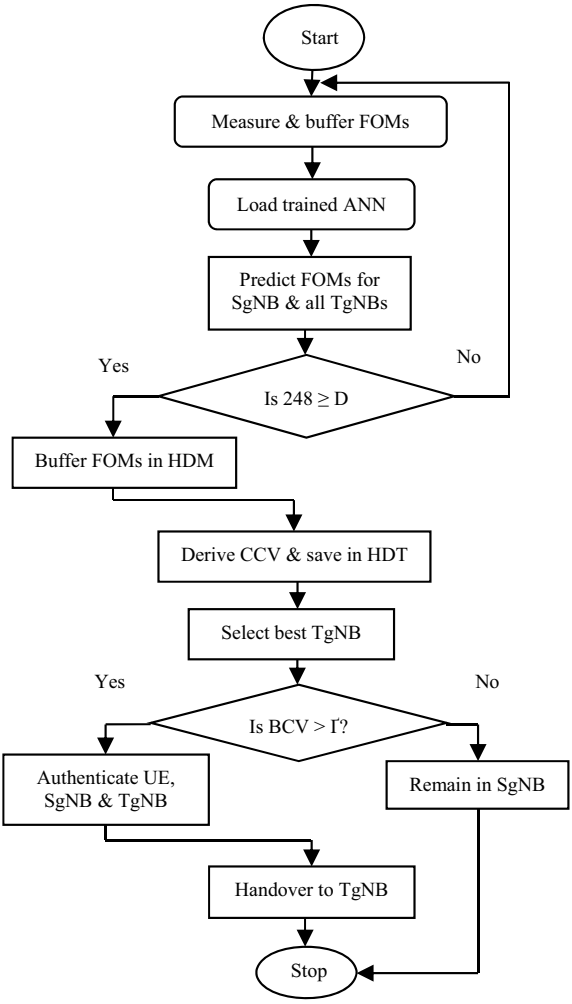


Fig. 1 Data flows in the proposed protocol

are then saved in HDT. Matching is then executed in HDT to select the cell with the best CV that is then checked against the handover factor Γ′. Here, if the best CV is greater than Γ′, the handover entities are authenticated and handover executed; otherwise, the UE remains in SgNB.

3.2.1 UE-TgNB Initialization Phase

This phase involves the initialization of some cryptographic primitives that are deployed during the UE and TgNB authentication and key agreement phase. It is executed through steps 1–4 described below.

Step 1: TgNB generates secret key $\mathfrak{B}$ and selects one-way hashing functions $\mathfrak{H} = \{h_0(.), h_1(.), h_2(.)$ and $h_3(.)\}$. The TgNB buffers $\mathfrak{B}$ before broadcasting $\mathfrak{H}$.

Step 2: The UE selects secret key $\mho$, random number R_1, its pseudo-identity PID_{UE} and secret token $\mathfrak{P}_{\text{UE}}$.

This is followed by the derivation of $\text{A}=\text{h}_0(\text{PID}_{\text{UE}}||\mathfrak{P}_{\text{UE}}||\text{R}_1)$. It then composes $M_1 = E_{\mho}(\text{PID}_{\text{UE}}, A)$ before sending $\text{M}_1$to the TgNB.

Step 3: Upon receiving M_1, the TgNB decrypts it and verifies whether PID_{UE} is in its identity database and if it is not, it chooses random numbers R_2, R_3 and R_4. Then, it sets $\bar{Y}_1 = R_3$, $\tilde{U}_1 = \tilde{U}_2 = R_4$ before computing long term secret key $\mathbb{Z}_{\text{UT}} = \text{h}_1(\text{PID}_{\text{UE}}||\mathfrak{B}||\text{R}_2)$, $\mathfrak{H}_1 = \left(\mathbb{Z}_{UT}||\bar{Y}_1\right) \oplus A$ and $\mathfrak{H}_2 = h_3(h_2(\mathbb{Z}_{UT}||A))$. The TgNB appends $\{\tilde{U}_1, \tilde{U}_2, \text{PID}_{\text{UE}}, \bar{Y}_1, R_2\}$ into its identity database. Afterward, it composes $M_2 = E_{\mathbb{Z}_{UT}}(\Re, \mathfrak{H}_1, \mathfrak{H}_2)$ before sending it to the UE.

Step 4: On receiving M_2, the UE chooses random Boolean number R_5, instantiates it to zero and buffers this value together with the contents of M_2 in its memory.

3.2.2 SgNB-TgNB Initialization Phase

This phase is similar to the one in Sect. 3.2.1 above and is executed through steps 1 to 3 described below.

Step 1: The SgNB selects pseudo-identity PID_{SgNB} and computes $M_3 = E_{\mathbb{Z}_{ST}}(R_6, \text{PID}_{\text{SgNB}})$ before sending M_3 to the TgNB.

Step 2: Upon receiving M_3, the TgNB decrypts it and checks whether PID_{SgNB} is in its identity database, and if it is not, it chooses random number R_7 before setting $\bar{Y}_2 = R_7$. It then initializes sequence number generators $\Im_S = \Im_T = 0$ before appending $\{\text{PID}_{\text{SgNB}}, \Im_T, \bar{Y}_2\}$ to its identity database. Afterward, it composes $M_4 = E_{\mathbb{Z}_{ST}}(\Im_S, \bar{Y}_2)$ before sending it to the SgNB.

Step 3: After receipt of M_4, the SgNB decrypts it and buffers its contents in its memory.

3.3 Authentication and Key Agreement

This phase is triggered whenever the any of the 5G supported devices requests any services from the core network. For this paper, the requested service is a handover from the current base station SgNB toward the target base station TgNB. This handover is described in steps 1–8 explained below. Table 1 presents the deployed symbols and their brief description.

Step 1: The user inputs PID_{UE} and $\mathfrak{P}_{\text{UE}}$ after which the UE derives $\text{A}=\text{h}_1(\text{PID}_{\text{UE}}||\mathfrak{P}_{\text{UE}}||\text{R}_1)$, $\mathbb{Z}_{UT}||\bar{Y}_1 = \mathfrak{H}_1 \oplus A$, $\mathfrak{H}_2^* = h_3(h_2(\mathbb{Z}_{UT}||A)$. Afterward, it checks whether $\mathfrak{H}_2^* = \mathfrak{H}_2$, and if this is not the case, user login is rejected. However, if this check is successful, it further checks whether $R_5 = 0$, and if it is, the UE executes the following updates: $\overline{Y}_1^* = \text{h}_1(\bar{Y}_1)$, $\mathfrak{H}_1^* = (\mathbb{Z}_{\text{UT}}||\bar{Y}_1{}^*) \oplus A$, $R_5 = 1$.

Table 1 Symbols

Symbol	Description
SgNB, TgNB	Source gNB and target gNB respectively
Ⱥ	TgNB system secret key
PID_{UE}	UE pseudo-identity
Ᵽ_{UE}	UE one-time secret token
$\mho$	UE secret key
R_i	Random numbers
$h(.)$	One-way hashing operation
$E_{\mho}$, $E_{\mathbb{Z}_{\text{UT}}}$	Encryption using key $\mho$ and $\mathbb{Z}_{\text{UT}}$ respectively
$\bar{Y}_1$	Dynamic hash chain value shared between UE and TgNB
$\bar{Y}_2$	Dynamic hash chain value shared between SgNB and TgNB
$\tilde{\mathrm{U}}_1$, $\tilde{\mathrm{U}}_2$	Two one-time identities assigned to UE at the TgNB
$\mathbb{Z}_{\text{UT}}$	Long-term shared secret key between UE and TgNB
$\Re$	TgNB assigned UE pseudonym
PID_{SgNB}	SgNB pseudo-identity
$\Im_S$, $\Im_T$	SgNB and TgNB sequence number generators respectively
T	ith timestamp
$\|\|$	Concatenation operation
$\oplus$	XOR operation
$\wp$	Session key between UE and SgNB
Γ	Threshold sequence number

Step 2: The UE chooses random number R_8 that it deploys to compute $\mathrm{N}_1 = (R_8 \| \text{PID}_{\text{SgNB}}) \oplus h_0(\Re \| \mathbb{Z}_{\text{UT}} \| \bar{Y}_1)$ and $\tilde{n}_1 = h_3(\text{PID}_{\text{UE}} \| \text{PID}_{\text{SgNB}} \| \Re \| R_8 \| \mathbb{Z}_{\text{UT}} \| \bar{Y}_1 \| T)$. It then composes $M_5 = \{T, \Re, \mathrm{N}_1, \tilde{n}_1\}$ before transmitting it to the TgNB.

Step 3: On receiving M_5, the TgNB executes freshness checks against the received T such that if M_5 fails the freshness check, then the authentication session is aborted. However, if this check is successful, the TgNB looks up its identity database to establish the $\{\tilde{\mathrm{U}}_1, \tilde{\mathrm{U}}_2\}$ that is associated with this $\Re$. This process begins by having the TgNB checking whether the received $\Re$ matches with either $\tilde{\mathrm{U}}_1$ or $\tilde{\mathrm{U}}_2$. Here, if $\Re = \tilde{\mathrm{U}}_1$ the implication is that the UE identity and $\bar{Y}_1$ were updated in the previous authentication session. As such, the TgNB is required to update it too by executing $\overline{Y}_1^* = h_1(\overline{Y}_1)$, followed by the computation of $\mathbb{Z}_{\text{UT}} = \mathrm{h}_1(\text{PID}_{\text{UE}} \| \text{Ⱥ} \| \mathrm{R}_2)$, $(\mathrm{R}_8 \| \text{PID}_{\text{SgNB}}) = \mathrm{N}_1 \oplus \mathrm{h}_0(\tilde{\mathrm{U}}_1 \| \mathbb{Z}_{\text{UT}} \| \bar{\mathrm{Y}}_1{}^*)$ $\tilde{n}_1^* = h_3(\text{PID}_{\text{UE}} \| \text{PID}_{\text{SgNB}} \| \tilde{\mathrm{U}}_1 \| R_8 \| \mathbb{Z}_{\text{UT}} \| \bar{Y}_1{}^* \| T)$. It then checks whether $\tilde{n}_1^* = \tilde{n}_1$. If this check is false, the session is aborted; otherwise, a new pseudonym $\tilde{\mathrm{U}}_1{}^*$is chosen followed by the setting of $\tilde{\mathrm{U}}_2 = \tilde{\mathrm{U}}_1$, $\tilde{\mathrm{U}}_1 = \tilde{\mathrm{U}}_1{}^*$ and $\overline{Y}_1 = \overline{Y}_1^*$.

Step 4: On condition that $\Re = \tilde{U}_2$, the implication is that $\Re$ and $\bar{Y}_1$ on the user side and $\bar{Y}_1$ in the TgNB were not refreshed in the preceding authentication session, but $\tilde{U}_1$ in the TgNB is refreshed. As such, the TgNB derives $\mathbb{Z}_{UT} = h_1(PID_{UE}||B||R_2)$, $(R_8||PID_{SgNB}) = N_1 \oplus h_0(\tilde{U}_2||\mathbb{Z}_{UT}||\bar{Y}_1)$, $\tilde{n}_1^* = h_3(PID_{UE}||PID_{SgNB}||\tilde{U}_2||R_8||\mathbb{Z}_{UT}||\bar{Y}_1||T)$. It then checks whether $\tilde{n}_1^* = \tilde{n}_1$, and if this is false, the session is aborted; otherwise, the TgNB chooses a new pseudonym $\tilde{U}_1^*$ before setting $\tilde{U}_1 = \tilde{U}_1^*$. On the other hand, on condition that $\Re \neq \tilde{U}_2$ and $\Re \neq \tilde{U}_1$, the TgNB aborts the authentication session.

Step 5: The TgNB stochastically chooses session key $\wp$ and derives $N_2 = (\wp||PID_{UE}) \oplus h_0(\bar{Y}_2||PID_{SgNB}||\Im_T)$, $\tilde{n}_2 = h_3(PID_{UE}||PID_{SgNB}||\wp||\bar{Y}_2||\Im_T)$. Thereafter, TgNB refreshes as $\overline{Y}_2^* = h_1(\bar{Y}_2||PID_{SgNB})$ and $\Im_T^* = \Im_T + 1$. Finally, TgNB constructs $M_6 = \{N_1, \tilde{n}_2, \Im_T^*\}$ and transmits it to the SgNB.

Step 6: Upon receipt of M_6, the SgNB confirms whether $1 \leq \Im_T^* - \Im_S \leq \Gamma$ and if this condition is false, the SgNB aborts the session. However, if this condition is true, the SgNB sets $\overline{Y}_2^* = \overline{Y}_2$ and derives $(\Im_T^* - \Im_S - 1)$ times $(\overline{Y}_2^* = h_1(\overline{Y}_2^*||PID_{SgNB})$. On condition that $\Im_T^* - \Im_S - 1 = 0$, then no hashing operations are executed, and as such, the SgNB derives $(\wp||PID_{UE}) = (N_2 \oplus h_0(\overline{Y}_2^*||PID_{SgNB}||(\Im_T - 1))$, $\tilde{n}_2^* = h_3(PID_{UE}||PID_{SgNB}||\wp||\overline{Y} \lim_{x \to \infty}_2^* ||(\Im_T - 1))$. This is followed by the confirmation of whether $\tilde{n}_2^* = \tilde{n}_2$, and if this condition is false, the session is aborted; otherwise, the SgNB computes $\tilde{n}_3 = h_3(PID_{SgNB}||PID_{UE}||\wp||\overline{Y}_2^*)$. It then executes the following updates: $\overline{Y}_2 = h_1(\overline{Y}_2^*||PID_{SgNB})$, $\Im_S = \Im_T$. Next, the SgNB constructs $M_7 = \{PID_{SgNB}, \tilde{n}_3\}$ and transmits it to the TgNB.

Step 7: Upon receiving M_7, the TgNB computes $\tilde{n}_3^* = h_3(PID_{SgNB}||PID_{UE}||\wp||\bar{Y}_2)$ and confirms whether the calculated $\tilde{n}_3^*$ matches the received $\tilde{n}_3$ in M_7. If this condition is false, the session is terminated; otherwise, the TgNB derives $N_3 = (\wp||\tilde{U}_1) \oplus h_0(R_8||\tilde{U}_2||\mathbb{Z}_{UT}||\bar{Y}_1)$, $\tilde{n}_4 = h_3(PID_{SgNB}||PID_{UE}||\wp||R_8||\tilde{U}_1)$. The TgNB composes $M_8 = \{N_3, \tilde{n}_4\}$ and transmits it to the UE.

Step 8: After receiving M_8, the UE computes $(\wp||\tilde{U}_1) = N_3 \oplus h_0(R_8||\Re||\mathbb{Z}_{UT}||\bar{Y}_1)$, $\tilde{n}_4^* = h_3(PID_{SgNB}||PID_{UE}||\wp||R_8||\tilde{U}_1)$. It then confirms whether the derived $\tilde{n}_4^*$ matches $\tilde{n}_4$ in the received M_8, and if this is false, the UE cannot authenticate the TgNB and the authentication session is aborted. However, if there is a match, the UE executes the following updates: $\Re = \tilde{U}_1$ and $R_5 = 0$.

4 Results and Discussion

This part presents the security evaluation as well as the performance evaluation of the proposed protocol. The simulation parameters and environment are similar to those in [13].

4.1 Security Evaluation

The Burrows–Abadi–Needham (BAN) logic is deployed to formally analyze the security features of the proposed algorithm. In addition, informal security analysis is executed to show that this protocol thwarts most of the 5G handover attacks.

4.1.1 Formal Security Analysis

To show the security and privacy features of the proposed protocol during the mutual authentication and key agreement phase, Burrows–Abadi–Needham (BAN) logic is deployed. In addition, informal security analysis is executed to show that the proposed protocol is resilient against some of the predominant attack models in 5G HetNets. In essence, BAN logic proofs the establishment of session key between the UE and SgNB upon successful execution of the proposed protocol. Table 2 presents the BAN logic notations in which S and T are the principles in the AKA process while F and G are the statements.

The BAN logic rules in Table 3 are also utilized during the formal analysis of the proposed protocol.

During the BAN logic-based proofs, the security goals in Table 4 are formulated.

The messages exchanged M_5, M_6, M_7 and M_8 among the UE, SgNB and TgNB during the authentication process are then idealized as shown in Table 5.

Afterward, the initial state assumptions (IAs) in Table 6 are made during the mutual authentication and authentication procedures.

Afterward, the following BAN logic steps (BLSs) are deployed to proof the attainment of the goals formulated in Table 4.

Based on M_5, it is straightforward to have BLS_1:

Table 2 BAN logic notations

Symbol	Description
H	Secret key known only to S and T
$S\mid\equiv F$	S believes statement F
$S\mid\sim F$	S once said F
$s \triangleleft F$	S sees F
$\#(F)$	Statement F is fresh
$<F>G$	F is combined with G
$(F)_H$	F is hashed using secret key H
$S \overset{H}{\leftrightarrow} T$	S and T deploy share secret key H for their communication
$S \overset{H}{\rightleftharpoons} T$	Secret key H is only known to S and T
(F, G)	Either F or G is part of statement (F,G)
$S\mid\Rightarrow F$	S has jurisdiction over F

Table 3 BAN logic rules

Rule	Description
$\frac{S\mid\equiv\#(F)}{S\mid\equiv\#(F,G)}$	Freshness rule (FR)
$\frac{S\mid\equiv S\overset{H}{\leftrightarrow}T, S\lhd\{F\}_H}{S\mid\equiv T\mid\sim F}$	Message-meaning rule (MMR)
$\frac{S\mid\equiv\#(F), S\mid\equiv T\mid\sim F}{S\mid\equiv T\mid\equiv F}$	Nonce verification rule (NVR)
$\frac{S\mid\equiv T\Rightarrow F, S\mid\equiv T\mid\equiv F}{S\mid\equiv F}$	Jurisdiction rule (JR)
$S\mid\equiv F,\ \frac{S\mid\equiv G}{S\mid\equiv(F,G)},\ \frac{S\mid\equiv(T\mid\equiv(F,G))}{S\mid\equiv(T\mid\equiv(F))},$ $\frac{S\mid\equiv(T\mid\sim(\mathrm{F,G}))}{S\mid\equiv(T\mid\sim(F))}$	Believe rule (BR)
$\frac{S\lhd(F,G)}{S\lhd F},\ \frac{S\lhd(F)_H)}{S\lhd F},\ \frac{S\lhd(F)_H), S\mid\equiv S\overset{H}{\leftrightarrow}T}{S\lhd F}$	Seeing rule (SR)

Table 4 Security goals

S. No.	Goal
SG-1	$\mathrm{UE}\mid\equiv(\mathrm{UE}\overset{\wp}{\leftrightarrow}\mathrm{SgNB})$
SG-2	$\mathrm{UE}\mid\equiv\mathrm{SgNB}\mid\equiv\left(\mathrm{UE}\overset{\wp}{\leftrightarrow}\mathrm{SgNB}\right)$
SG-3	$\mathrm{SgNB}\mid\equiv\left(\mathrm{UE}\overset{\wp}{\leftrightarrow}\mathrm{SgNB}\right)$
SG-4	$\mathrm{SgNB}\mid\equiv\mathrm{UE}\mid\equiv\left(\mathrm{UE}\overset{\wp}{\leftrightarrow}\mathrm{SgNB}\right)$

Table 5 Idealized messages

M_5	**UE → TgNB**: $\{\mathrm{T}, \Re, \mathrm{N}_1, \tilde{n}_1\}$ $\left(\mathrm{UE}\overset{R_8}{\leftrightarrow}\mathrm{TgNB},\ \mathrm{PID}_{\mathrm{SgNB}}\right)_{\mathrm{UE}\overset{\mathbb{Z}_{\mathrm{UT}}\|\overline{\Upsilon}_1}{\leftrightarrow}\mathrm{TgNB}}$ $<\mathrm{PID}_{\mathrm{UE}},\mathrm{PID}_{\mathrm{SgNB}},\Re,\ \mathrm{R}_8\ ,\mathrm{T}>_{\mathrm{UE}\overset{\mathbb{Z}_{\mathrm{UT}}\|\hat{\Upsilon}_1}{\leftarrow}\mathrm{TgNB}}$
M_6	**TgNB → SgNB**: $\{\mathrm{N}_1, \tilde{n}_2, \Im_T^*\}$ $\left(\mathrm{TgNB}\overset{\wp}{\leftrightarrow}\mathrm{SgNB},\ \mathrm{PID}_{\mathrm{UE}}\right)_{\mathrm{TgNB}\overset{\overline{\Upsilon}_2}{\leftrightarrow}\mathrm{SgNB}}$ $<\mathrm{PID}_{\mathrm{UE}},\ \mathrm{PID}_{\mathrm{SgNB}},\ \mathrm{TgNB}\overset{\wp}{\leftrightarrow}\mathrm{SgNB},\ \Im_{\mathrm{T}}>_{\mathrm{TgNB}\overset{\overline{\Upsilon}_2}{\leftrightarrow}\mathrm{SgNB}}$
M_7	**SgNB → TgNB:** $\{\mathrm{PID}_{\mathrm{SgNB}}, \tilde{n}_3\}$ $<\mathrm{PID}_{\mathrm{SgNB}},\ \mathrm{PID}_{\mathrm{UE}},\ \mathrm{SgNB}\overset{\wp}{\leftrightarrow}\mathrm{TgNB}>_{\mathrm{SgNB}\overset{\overline{\Upsilon}_2}{\leftrightarrow}\mathrm{TgNB}}$
M_8	**TgNB → UE:**$\{\mathrm{N}_3, \tilde{n}_4\}$ $(\mathrm{TgNB}\overset{\wp}{\longleftrightarrow}\mathrm{UE}, \tilde{\mathrm{U}}'_1)_{\mathrm{UE}\overset{\mathbb{Z}_{\mathrm{UT}}\|\hat{\Upsilon}_1}{\leftarrow}\mathrm{TgNB}}$ $<\mathrm{PID}_{\mathrm{SgNB}},\mathrm{PID}_{\mathrm{UE}},\mathrm{TgNB}\overset{\wp}{\longleftrightarrow}\mathrm{UE},\tilde{\mathrm{U}}'_1>_{\mathrm{UE}\overset{H}{\leftrightharpoons}\mathrm{TgNB}}$

Table 6 Initial state assumptions

IA_{1s}	$TgNB\vert\equiv \#(\mathfrak{T})$
IA_2	$TgNB\vert\equiv \#(R_8)$
IA_3	$SgNB\vert\equiv \#(\wp)$
IA_4	$UE\vert\equiv \#(\wp)$
IA_5	$UE\vert\equiv UE \overset{\mathbb{Z}_{UT}\vert\vert\overline{\Upsilon}_1}{\leftrightarrow} TgNB$
IA_6	$TgNB\vert\equiv UE \overset{\mathbb{Z}_{UT}\vert\vert\overline{\Upsilon}_1}{\leftrightarrow} TgNB$
IA_7	$SgNB\vert\equiv SgNB \overset{\overline{\Upsilon}_2}{\leftrightarrow} TgNB$
IA_8	$TgNB\vert\equiv SgNB \overset{\overline{\Upsilon}_2}{\leftrightarrow} TgNB$
IA_9	$UE\vert\equiv TgNB\vert\Rightarrow UE \overset{\wp}{\leftrightarrow} SgNB$
IA_{10}	$SgNB\vert\equiv TgNB\vert\Rightarrow UE \overset{\wp}{\leftrightarrow} SgNB$

$\mathbf{BLS_1}$: $TgNB \lhd \left(UE \overset{R_8}{\leftrightarrow} TgNB,\ PID_{SgNB}\right)_{UE \overset{\mathbb{Z}_{UT}\vert\vert\overline{\Upsilon}_1}{\leftrightarrow} TgNB}$.

According to IA_6 MMR is applied on BLS_1 to yield BLS_2:

$\mathbf{BLS_2}$: $TgNB\vert\equiv UE\vert \sim (UE \overset{R_8}{\leftrightarrow} TgNB, PID_{SgNB})$.

Based on IA_6 and FR, BLS_3:

$\mathbf{BLS_3}$: $TgNB\vert\equiv \#(\ PID_{UE}, PID_{SgNB}, \Re, UE \overset{R_8}{\longleftrightarrow} TgNB, \mathfrak{T})\cdot$

Applying the NVR on both BLS_2 and BLS_3 yields BLS_4:

$\mathbf{BLS_4}$: $TgNB\vert\equiv (\ PID_{UE}, PID_{SgNB}, \Re, UE \overset{R_8}{\longleftrightarrow} TgNB, \mathfrak{T})\cdot$

Based on M_6, it is straight forward to obtain BLS_5:

$\mathbf{BLS_5}$: $SgNB \lhd \left(TgNB \overset{\wp}{\leftrightarrow} SgNB,\ PID_{UE}\right)_{TgNB \overset{\overline{\Upsilon}_2}{\leftrightarrow} SgNB}$.

Using IA_7, MMR is applied on BLS_5 to get BLS_6:

$\mathbf{BLS_6}$: $SgNB\vert\equiv TgNB\vert \sim (TgNB \overset{\wp}{\leftrightarrow} SgNB, PID_{UE})$.

Based on IA_3 and FR, BLS_7 is obtained:

$\mathbf{BLS_7}$: $SgNB\vert\equiv \#(\ PID_{UE},\ PID_{SgNB},\ TgNB \overset{\wp}{\leftrightarrow} SgNB,\ \Im_T)$.

On the other hand, the application of NVR on both BLS_6 and BLS_7 yields BLS_8:

$\mathbf{BLS_8}$: $SgNB\vert\equiv TgNB\vert \equiv (\ PID_{UE},\ PID_{SgNB},\ TgNB \overset{\wp}{\leftrightarrow} SgNB, \Im_T)$.

Based on M_7, BLS_9 is obtained:

$\mathbf{BLS_9}$: $TgNB \lhd < PID_{SgNB}, PID_{UE}, \wp >_{SgNB \overset{\overline{\Upsilon}_2}{\leftrightarrow} TgNB}$.

According to IA_3, MMR is applied in BLS_9 to yield BLS_{10}:

$\mathbf{BLS_{10}}$: $TgNB\vert\equiv SgNB\vert \sim (\ PID_{SgNB},\ PID_{UE},\ SgNB \overset{\wp}{\leftrightarrow} TgNB)$.

The application of NVR on BLS_{10} results in BLS_{11}:

$\mathbf{BLS_{11}}$: $TgNB\vert\equiv (SgNB\vert \equiv (\ PID_{SgNB},\ PID_{UE},\ SgNB \overset{\wp}{\leftrightarrow} TgNB)$.

According to M_8, BLS_{12} can be inferred:

$\mathbf{BLS_{12}}$: $UE \lhd UE \lhd (TgNB \overset{\wp}{\leftrightarrow} UE, \tilde{U}'_1)_{UE \overset{\mathbb{Z}_{UT}\vert\vert\bar{Y}_1}{\longleftrightarrow} TgNB}$.

Using IA_5, MMR is applied on BLS_{12} to obtain BLS_{13}:

BLS$_{13}$: UE|$\equiv$ UE|$\equiv$ TgNB|$\sim$(TgNB $\overset{\wp}{\leftrightarrow}$ UE, $\tilde{U}'_1$)·
Applying FR on IA$_4$ results in BLS$_{14}$:
BLS$_{14}$: UE|$\equiv$ #(PID$_{\text{SgNB}}$, PID$_{\text{UE}}$, TgNB $\overset{\wp}{\leftrightarrow}$ UE, $\tilde{U}'_1$)·
Based on BLS$_{13}$ and BLS$_{14}$, the NVR is applied to yield BLS$_{15}$:
BLS$_{15}$: UE|$\equiv$ (PID$_{\text{SgNB}}$, PID$_{\text{UE}}$, TgNB $\overset{\wp}{\leftrightarrow}$ UE, $\tilde{U}'_1$)·
On the other hand, using BR on BLS$_6$ and BLS$_7$ results in BLS$_{16}$:
BLS$_{16}$: SgNB| $\equiv$ (TgNB $\overset{\wp}{\leftrightarrow}$ SgNB).
The application of BR on BLS$_8$ yields BLS$_{17}$:
BLS$_{17}$: SgNB|$\equiv$ (TgNB| $\equiv$ (TgNB $\overset{\wp}{\leftrightarrow}$ SgNB)).
However, the usage of BR on BLS$_{11}$ results in BLS$_{18}$:
BLS$_{18}$: TgNB|$\equiv$ (SgNB| $\equiv$ (SgNB $\overset{\wp}{\leftrightarrow}$ TgNB).
On the other hand, applying BR on both BLS$_{13}$ and BLS$_{14}$ yields BLS$_{19}$:
BLS$_{19}$: UE | $\equiv$ (TgNB $\overset{\wp}{\leftrightarrow}$ UE).
Similarly, BR is applied on BLS$_{15}$ to yield BLS$_{20}$:
BLS$_{20}$: UE|$\equiv$ (TgNB| $\equiv$ (TgNB $\overset{\wp}{\leftrightarrow}$ UE).
Based on IA$_{10}$ and BLS$_{16}$, BLS$_{21}$ is obtained:
BLS$_{21}$: SgNB| $\equiv$ (UE $\overset{\wp}{\leftrightarrow}$ SgNB), achieving SG-3.
However, based on IA$_{10}$ and BLS$_{17}$, BLS$_{22}$ is obtained:
BLS$_{22}$: SgNB|$\equiv$ (UE|$\equiv$ $\left(\text{UE} \overset{\wp}{\leftrightarrow} \text{SgNB}\right)$), attaining SG-4.
Similarly, from IA$_9$, BLS$_{18}$ and BLS$_{19}$, BLS$_{23}$ is attained:
BLS$_{23}$: UE|$\equiv$ (SgNB $\overset{\wp}{\leftrightarrow}$ UE)), hence SG-1 is realized.
Based on IA$_9$, BLS$_{18}$ and BLS$_{20}$, BLS$_{24}$ is obtained:
BLS$_{24}$: UE|$\equiv$ (SgNB|$\equiv$ $\left(\text{SgNB} \overset{\wp}{\leftrightarrow} \text{UE}\right)$, attaining SG-2.

The realization of the four security goals proofs that both UE and SgNB share a session key $\wp$.

4.1.2 Informal Security Analysis

The lemmas below and their proofs are deployed to demonstrate the robustness of the proposed protocol.

Lemma 1 *The proposed protocol is robust against MitM attacks.*

Proof To prevent an adversary ¥ from intercepting the exchanged messages during the mutual authentication and key agreement, the proposed protocol deploys $\mathbb{Z}_{\text{UT}}$, $\bar{Y}_1$ and $\bar{Y}_2$. As such, it is difficult for ¥ to forge messages M_5, M_6, M_7 and M_8 exchanged during the AKA phase, devoid of these secret parameters.

Lemma 2 *The proposed protocol offers mutual authentication.*

Proof During the UE and TgNB communication, the UE is authenticated by the TgNB through the computation of $\tilde{n}_1^* = h_3(\text{PID}_{\text{UE}} \| \text{PID}_{\text{SgNB}} \| \Re \| \text{R}_8 \| \mathbb{Z}_{\text{UT}} \| \bar{Y}_1 \| T)$ which

is then checked against the received $\tilde{n}_1$ in M_5. On the other hand, the UE authenticates TgNB by computing $\tilde{n}_4^* = h_3(\mathrm{PID}_{\mathrm{SgNB}} \| \mathrm{PID}_{\mathrm{UE}} \| \wp \| \mathrm{R}_8 \| \tilde{\mho}_1)$ that is then checked against the received $\tilde{n}_4$ in the received M_8. Since ¥ requires secrets $\mathbb{Z}_{\mathrm{UT}}$ and $\bar{Y}_1$ to forge any of the exchanged messages either for the UE or TgNB. On the other hand, during message exchanges between TgNB and SgNB, the TgNB is authenticated by SgNB by computing $\tilde{n}_2^* = h_3(\mathrm{PID}_{\mathrm{UE}} \| \mathrm{PID}_{\mathrm{SgNB}} \| \wp \| \overline{Y}_2^* \| (\Im_T - 1))$ and confirming whether it matches $\tilde{n}_2$ received in message M_6. Similarly, TgNB authenticates SgNB through $\tilde{n}_3^* = h_3(\mathrm{PID}_{\mathrm{SgNB}} \| \mathrm{PID}_{\mathrm{UE}} \| \wp \| \bar{Y}_2)$ which is checked against $\tilde{n}_3$ received in M_7. As such, it is difficult for ¥ to forge messages exchanged between SgNB and TgNB without a valid $\bar{Y}_2$.

Lemma 3 *Replay attacks are effectively thwarted in the proposed protocol.*

Proof To curb this attack, the initial communication between the UE and TgNB involves timestamp Ŧ for message freshness checks. However, sequence numbers are deployed for SgNB and TgNB communication to prevent packet replay attacks. As such, upon the execution of the proposed AKA protocol, all the three entities are assured that this session is current.

Lemma 4 *The proposed protocol is resilient against privileged insider attacks.*

Proof During the initialization phase, the UE transmits $\mathrm{PID}_{\mathrm{UE}}$ and $A = h_0(\mathrm{PID}_{\mathrm{UE}} \| \text{Ꝑ}_{\mathrm{UE}} \| \mathrm{R}_1)$ to the TgNB instead of its one-time secret token Ꝑ_{UE} that will otherwise help ¥ to identify this particular UE. Since A deploys a one-way hash function and random number R_1 that is unknown to ¥, the privileged insider ¥ cannot derive it and hence this attack fails.

Lemma 5 *Anonymity and untraceability are assured in the proposed protocol.*

Proof The proposed protocol deploys stochastic pseudonym $\Re$ for the UE instead of its real identity. This parameter is randomly chosen and refreshed upon successful authentication as in Step 8. As such, it is not possible for ¥ to decipher the real identity of the users. Similarly, it is cumbersome for the attacker ¥ to trace users using $\Re$ due to its dynamic nature.

Lemma 6 *The proposed protocol is resilient against spoofing attacks.*

Proof Suppose that attacker ¥ attempts to masquerade as legitimate UE or SgNB. To accomplish this, ¥ must derive $\tilde{n}_1 = h_3(\mathrm{PID}_{\mathrm{UE}} \| \mathrm{PID}_{\mathrm{SgNB}} \| \Re \| R_8 \| \mathbb{Z}_{\mathrm{UT}} \| \bar{Y}_1 \| T)$ and $\tilde{n}_3 = h_3(\mathrm{PID}_{\mathrm{SgNB}} \| \mathrm{PID}_{\mathrm{UE}} \| \wp \| \overline{Y}_2^*)$. Parameter $\tilde{n}_1$ incorporates dynamic security parameter $\Re$, random number R_8 and long-term shared secret key between UE and TgNB, $\mathbb{Z}_{\mathrm{UT}}$. In addition, timestamp T and dynamic hash chain value shared between UE and TgNB, $\bar{Y}_1$ is involved. Similarly, $\tilde{n}_3$ involves dynamic hash chain value shared between SgNB and TgNB, $\overline{Y}_2^*$ and session key $\wp$. Consequently, the correct computation of both $\tilde{n}_1$ and $\tilde{n}_3$ by adversary ¥ is infeasible and hence cannot spoof the UE or SgNB.

Lemma 7 *Backward and forward key secrecy is assured in the proposed protocol.*

Table 7 Security features comparisons

Attack model	[38]	[36]	[3GPP R16]	[34]	Proposed
Eavesdropping	√	√	x	√	√
Mutual authentication	√	√	√	√	√
Forward key secrecy	–	–	x	–	√
Key agreement	√	√	√	√	√
MitM	√	√	x	√	√
Spoofing	–	–	x	–	√

Legend – Not considered, √ effective, x ineffective

Proof Suppose that adversary ¥ has captured $\mathbb{Z}_{UT}$, $\bar{Y}_1$ and $\bar{Y}_2$ belonging to the UE, SgNB and TgNB. The objective is then to derive session key $\wp$ deployed between UE and SgNB. However, this computation will fail since $\bar{Y}_1$ and $\bar{Y}_2$ are refreshed after every successful AKA procedures as in step 5 to step 8. In addition, previous values for $\bar{Y}_1$ and $\bar{Y}_2$ cannot be derived from $\overline{Y}_1^*$ and $\overline{Y}_2^*$ owing to the deployed one-way hashing function.

Lemma 8 *The proposed protocol is robust against impersonation attacks.*

Proof Suppose that ¥ wants to masquerade as the UE by attempting to forge a legitimate authentication message $M_5 = \{T, \Re, N_1, \tilde{n}_1\}$ sent from the UE toward the TgNB. However, since $N_1 = (R_8 || PID_{SgNB}) \oplus h_0(\Re || \mathbb{Z}_{UT} || \bar{Y}_1)$ and $\tilde{n}_1 = h_3(PID_{UE} || PID_{SgNB} || \Im || R_8 || \mathbb{Z}_{UT} || \bar{Y}_1 || T)$ this forgery requires knowledge of $\bar{Y}_1$. Since this dynamic hash chain value shared between UE and TgNB, $\bar{Y}_1$ is unavailable to ¥, this attack flops. Table 7 presents the security comparison of the proposed protocol against other related schemes.

As shown in Table 7, the proposed protocol has more security features among all its peers.

4.2 Performance Analysis

In this section, the handover success rate, execution time and bandwidth requirements are presented.

Handover success rate: To evaluate the target cell performance of the proposed protocol, the number of successful handovers was investigated against the total number of executed handovers as shown in Fig. 2. This number of successful handovers was then compared with that of the conventional 3GPP R16. It is clear from Fig. 2 that the proposed protocol has higher handover success rate that 3GPP R16. This is attributed to the deployed ANN-FL model that facilitated faster and optimum selection of the target cell during the handover process.

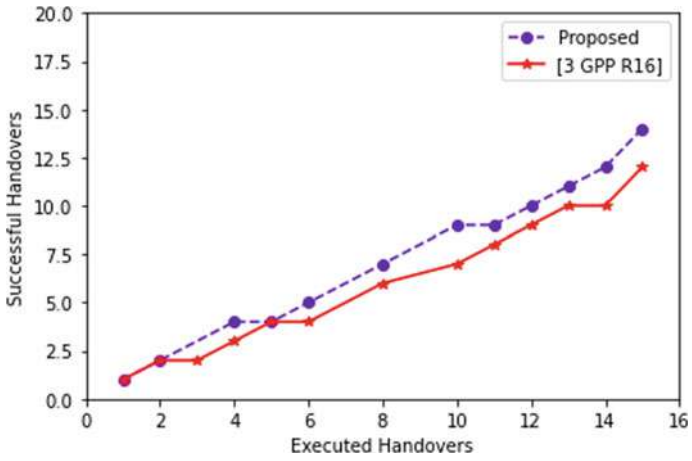


Fig. 2 Handover success rate

Execution time: During the AKA phase, the UE executes 8 one-way hashing operations while the TgNB executes 10 hashing operations. On the other hand, the SgNB carries out 4 hashing operations. Consequently, a total of 22 hashing operations are executed in the proposed protocol.

On the other hand, the protocols in [34, 36, 38] and 3GPP R16 have execution durations of 73.6652 ms, 67.1758 ms, 0.0134 ms and 0.0078 ms, respectively, as shown in Table 8.

Bandwidth requirements: In the proposed protocol, messages $M_5 = \{T, \Re, \mathrm{N}_1, \tilde{n}_1\}$, $M_6 = \{\mathrm{N}_1, \tilde{n}_2, \Im_T^*\}$, $M_7 = \{\mathrm{PID}_{\mathrm{SgNB}}, \tilde{n}_3\}$ and $M_8 = \{\mathrm{N}_3, \tilde{n}_4\}$ are exchanged during AKA procedures. Using the values in [34], identity, pseudo-identity, advanced encryption standard (AES) key, hash, random number and timestamps are 128 bits, 256 bits, 128 bits, 64 bits, 128 bits and 17 bits, respectively. As such, the total bandwidth requirement is 1041 bits as derived below:

$M_5 = \{T, \Re, \mathrm{N}_1, \tilde{n}_1\}$: $(T = 17, \Re = 256, \mathrm{N}_1 = \tilde{n}_1 = 64) = 401$ bits.

$\mathrm{M}_6 = \{\mathrm{N}_1, \tilde{n}_2, \Im_T^*\}$: $(\mathrm{N}_1 = \tilde{n}_2 = \Im_T^* = 64) = 192$ bits.

$\mathrm{M}_7 = \{\mathrm{PID}_{\mathrm{SgNB}}, \tilde{n}_3\}$: $(\mathrm{PID}_{\mathrm{SgNB}} = 256, \tilde{n}_3 = 64) = 320$ bits.

$\mathrm{M}_8 = \{\mathrm{N}_3, \tilde{n}_4\}$: $(\mathrm{N}_3 = \tilde{n}_4 = 64) = 128$.

Table 8 Computation costs comparisons

Scheme	Execution time (ms)
[36]	73.6652
[38]	67.1758
[34]	0.0134
[3GPP R16]	0.0078
Proposed	0.0286

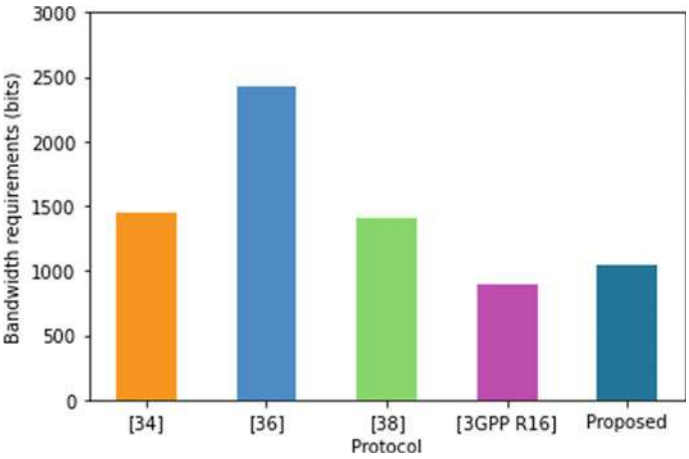


Fig. 3 Bandwidth comparisons

On the other hand, the schemes in [34, 36, 38] and 3GPP R16 have bandwidth requirements of 2432 bits, 1408 bits, 1442 bits and 896 bits, respectively, as shown in Fig. 3.

As shown in Fig. 3, the protocol in [36] had the highest bandwidth requirements while the scheme 3GPP R16 has the lowest bandwidth requirements. However, this AKA protocol has several security issues such as susceptibility to impersonation and DoS attacks.

5 Conclusion and Future Work

The current intelligent cell selection protocols have been observed to incorporate insufficient parameters as inputs to the trained model. This has been noted to result in ping-pong handovers as well as diminished quality of service in the target cells. Worse still, these intelligent cell selection protocols rarely take into consideration the security and privacy of the handover process. Consequently, other schemes presented in literature have attempted to address these issues, exampled by 3GPP's AKA protocols. However, these protocols face many security and privacy shortfalls such as susceptibility to DoS, impersonation and MitM attacks. A number of protocols have therefore been presented to address 5G AKA protocol challenges. Unfortunately, these schemes fail to comprehensively address these issues, and in some cases, they result in extensive computation and communication costs. The proposed protocol has been shown to have reduced handover latencies, average execution time and communication overheads. Moreover, it provides increased security and privacy compared with other related protocols. Future work lies in the assessment of the proposed protocol using figures of merit that were not covered in this work.

References

1. Ahmad WS, Radzi NAM, Samidi FS, Ismail A, Abdullah F, Jamaludin MZ, Zakaria MN (2020) 5G technology: towards dynamic spectrum sharing using cognitive radio networks. IEEE Access 8:14460–14488
2. Zhang Y, Xiong L, Yu J (2020) Deep learning based user association in heterogeneous wireless networks. IEEE Access 8:197439–197447
3. Aljohani SL, Alenazi MJ (2021) MPResiSDN: multipath resilient routing scheme for SDN-enabled smart cities networks. Appl Sci 11:1900
4. Mahira AG, Subhedar MS (2017) Handover decision in wireless heterogeneous networks based on feed forward artificial neural network. In: Computational intelligence in data mining, Springer, Singapore, pp 663–669
5. Kamel M, Hamouda W, Youssef A (2017) Performance analysis of multiple association in ultra-dense networks. IEEE Trans Commun 65:3818–3831
6. Alablani IA, Arafah MA (2021) An adaptive cell selection scheme for 5G heterogeneous ultra-dense networks. IEEE Access 9:64224–64240
7. Zakeri A, Khalili A, Javan MR, Mokari N, Jorswieck E (2021) Robust energy-efficient resource management, SIC ordering, and beamforming design for MC MISO-NOMA enabled 6G. IEEE Trans Signal Process 69:2481–2498
8. Alablani IA, Arafah MA (2021) Enhancing 5G small cell selection: a neural network and IoV-based approach. Sensors 21(19):6361
9. Mroue M, Prevotct JC, Nouvel F, Mohanna Y (2018) A neural network based handover for multi-RAT heterogeneous networks with learning agent. In: 2018 13th international symposium on reconfigurable communication-centric systems-on-chip (ReCoSoC), IEEE, pp 1–6
10. Bielza C, Larranaga P (2014) Discrete bayesian network classifiers: a survey. ACM Comput Surveys (CSUR) 47(1):1–43
11. Alotaibi NM, Alwakeel SS (2015) A neural network based handover management strategy for heterogeneous networks. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA), IEEE, PP 1210–1214
12. Aibinu AM, Onumanyi AJ, Adedigba AP, Ipinyomi M, Folorunso TA, Salami MJ (2017) Development of hybrid artificial intelligent based handover decision algorithm. Eng Sci Technol Int J 20(2):381–390
13. Nyangaresi VO, Rodrigues AJ, Abeka SO (2020) Secure handover protocol for high speed 5G networks. Int J Adv Netw Appl 11(6):4429–4442
14. Parambanchary D, Rao VM (2020) WOA-NN: a decision algorithm for vertical handover in heterogeneous networks. Wireless Netw 26(1):165–180
15. Waheidi YM, Jubran M, Hussein M (2019) User driven multiclass cell association in 5G HetNets for mobile IoT devices. IEEE Access 7:82991–83000
16. Bagheri H, Noor-A-Rahim M, Liu Z, Lee H, Pesch D, Moessner K, Xiao P (2021) 5G NR-V2X: toward connected and cooperative autonomous driving. IEEE Commun Stand Magaz 5(1):48–54
17. Fang D, Qian Y (2020) 5G wireless security and privacy: Architecture and flexible mechanisms. IEEE Veh Technol Mag 15(2):58–64
18. Hojjati M, Shafieinejad A, Yanikomeroglu H (2020) A blockchain-based authentication and key agreement (AKA) protocol for 5G networks. IEEE Access 8:216461–216476
19. Wickramasuriya DS, Perumalla CA, Davaslioglu K, Gitlin RD (2017) Base station prediction and proactive mobility management in virtual cells using recurrent neural networks. In: Proceedings of the 2017 IEEE 18th wireless and microwave technology conference (WAMICON), IEEE, FL, USA, pp 1–6
20. Adewale AA, Ekong EE, Ibikunle FA, Orimogunje A, Abolade J (2019) Ping-pong reduction for handover process using adaptive hysteresis margin: a methodological approach. In: IOP conference series: materials science and engineering, IOP Publishing, vol 640 no (1), pp 012118

21. Qian Zhang S, Xue F, Ageen Himayat N, Talwar S, Kung H (2018) A machine learning assisted cell selection method for drones in cellular networks. In: Proceedings of the 2018 IEEE 19th international workshop on signal processing advances in wireless communications (SPAWC), IEEE, Kalamata, Greece, pp 1–5
22. Kunarak S, Sulessathira R, Dutkiewicz E (2013) Vertical handoff with predictive RSS and dwell time. In: 2013 IEEE region 10 conference (31194), IEEE, pp 1–5
23. Perez JS, Jayaweera SK, Lane S (2017) Machine learning aided cognitive RAT selection for 5G heterogeneous networks. In: Proceedings of the 2017 IEEE international black sea conference on communications and networking (BlackSeaCom), IEEE, pp 1–5
24. Zappone A, Sanguinetti L, Debbah M (2018) User association and load balancing for massive MIMO through deep learning. In: Proceedings of the 2018 52nd Asilomar conference on signals, systems, and computers, Pacific Grove, CA, USA, pp 1262–1266
25. Balapuwaduge IAM, Li FY (2019) Hidden markov model based machine learning for mMTC device cell association in 5G networks. In: Proceedings of the ICC 2019—2019 IEEE international conference on communications (ICC), Shanghai, China, pp 1–6
26. Pandey D, Kim BH, Gang HS, Kwon GR, Pyun JY (2018) Maximizing network utilization in IEEE 802.21 assisted vertical handover over wireless heterogeneous networks. J Inf Proc Syst 14(3):771–789
27. Marwan A, Mourad O, Mousa H (2020) A survey of fuzzy logic in wireless localization. EURASIP J Wirel Commun Netw 2020:89
28. Shinkuma R, Nishio T, Inagaki Y, Oki E (2020) Data assessment and prioritization in mobile networks for real-time prediction of spatial information using machine learning. EURASIP J Wirel Commun Netw 2020:1–19
29. Yazdinejad A, Parizi RM, Dehghantanha A, Choo KK (2020) P4-to-blockchain: a secure blockchain-enabled packet parser for software defined networking. Comput Secur 88:101629
30. Nyangaresi VO, Petrovic N (2021) Efficient PUF based authentication protocol for internet of drones. In: 2021 international telecommunications conference (ITCEgypt), IEEE, pp 1–4
31. Nyangaresi VO, Rodrigues AJ, Abeka SO (2020) Neuro-fuzzy based handover authentication protocol for ultra dense 5G networks. In: 2020 2nd global power, energy and communication conference (GPECOM), IEEE, pp 339–344
32. Alawe I, Hadjadj-Aoul Y, Ksentini A, Bertin P, Darche D (2018) On the scalability of 5G core network: the AMF case. In: CCNC 2018–2018 15th IEEE annual consumer communications and networking conference, pp 1–6
33. Ferrag MA, Maglaras L, Argyriou A, Kosmanos D, Janicke H (2018) Security for 4G and 5G cellular networks: a survey of existing authentication and privacy-preserving schemes. J Netw Comput Appl 101:55–82
34. Xue K, Meng W, Zhou H, Wei DS, Guizani M (2020) A lightweight and secure group key based handover authentication protocol for the software-defined space information network. IEEE Trans Wireless Commun 19(6):3673–3684
35. Fortino G, Messina F, Rosaci D, Sarne GM (2019) Using blockchain in a reputation-based model for grouping agents in the Internet of Things. IEEE Trans Eng Manage 67(4):1231–1243
36. Lai C, Li H, Lu R, Jiang R, Shen X (2014) SEGR: a secure and efficient group roaming scheme for machine to machine communications between 3GPP and WiMAX networks. In: Proceedings of 2014 IEEE international conference on communications (ICC), IEEE, pp 1011–1016
37. Nyangaresi VO, Rodrigues AJ, Abeka SO (2020) Efficient group authentication protocol for secure 5G enabled vehicular communications. In: 2020 16th international computer engineering conference (ICENCO), IEEE, pp 25–30
38. He D, Chan S, Guizani M (2015) Handover authentication for mobile networks: security and efficiency aspects. IEEE Network 29(3):96–103
39. Javaid N, Sher A, Nasir H, Guizani N (2018) Intelligence in IoT-based 5G networks: opportunities and challenges. IEEE Commun Mag 56:94–100
40. Liu X, Zhang X (2019) Rate and energy efficiency improvements for 5G-Based IoT with simultaneous transfer. IEEE Internet Things J 6:5971–5980

41. Nyangaresi VO, Abeka SO, Rodrigues AJ (2020) Delay sensitive protocol for high availability LTE handovers. Am J Netw Commun 9(1):1–10
42. Nyangaresi VO, Abeka SO, Rodrigues AJ (2020) Tracking area boundary-aware protocol for pseudo stochastic mobility prediction in LTE Networks. I.J. Inf Techand Comput Sci 5:52–62
43. Nyangaresi VO, Abeka SO, Rodgrigues AJ (2018) Secure timing advance based context-aware handover protocol for vehicular ad-hoc heterogeneous networks. Int J Cyber-Secur Dig Forens 7(3):256–275
44. Nyangaresi VO, Rodrigues AJ, Abeka SO (2020) ANN-FL secure handover protocol for 5G and beyond networks. In: International conference on e-infrastructure and e-services for developing countries, Springer, Mauritius, pp 99–118

Chapter 9
Performance Improvement of CTNR Protocol in Wireless Sensor Network Using Machine Learning

Shalini Sharma, Amandeep Kaur Sohal, and Mandeep Kaur Walia

1 Introduction

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are now used in a variety of disciplines, including naval surveillance, medical sciences, home security, smart cities, catastrophe avoidance (such as floods and earthquakes), marketing firms, wildlife monitoring, weather monitoring, pollution monitoring, and so on. SNs in WSNs typically act as cameras that transmit images and video content to viewers. These cameras collect images and video data and transmit it back to the viewers. WSNs are also used by the army, when SNs are stationed in enemy territory to collect data and send it to the headquarters. There are a variety of smaller sensor networks that are meant to sense data from the field where they are deployed, process it, and send it to BS. These SNs were launched from the air to regions where humans could not go, such as volcanoes and oceans. The nodes in a WSN typically contain a radio transceiver, a microcontroller, and a power source or battery [1]. SNs use a variety of sensors designed for a precise goal, as an example temperature, pressure, humidity, lighting conditions, vehicle movement, speed detection, object size measurements, and stress measurements, along with appearance/non-appearance detections. Nano-sensing and unwired communication are used in many applications, including maritime applications that rely on SNs to detect attacks, communicate, and control operations. Environment applications include flood detection, fire detection, and field monitoring. All SNs have a short battery life. Battery replacements for SNs are not possible. It is crucial to keep

S. Sharma (✉) · A. Kaur Sohal · M. Kaur Walia
Guru Nanak Dev Engineering College, Ludhiana, India
e-mail: sharmavatish98@gmail.com

A. Kaur Sohal
e-mail: amandeepkaursohal@gndec.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_10

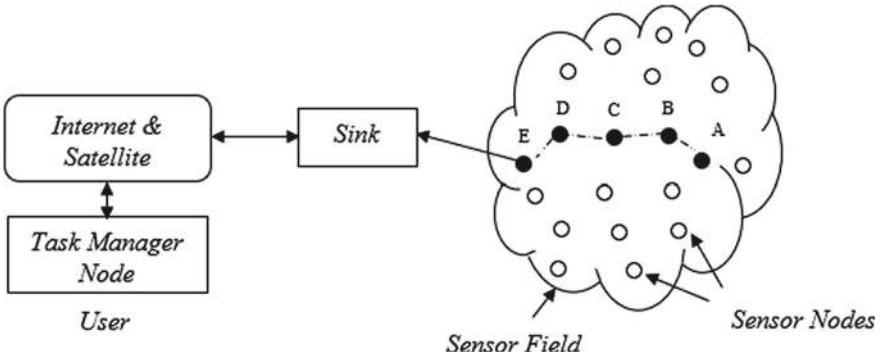


Fig. 1 Wireless sensor network

in mind that a major shortcoming about sensor networks is energy usage, which is the result of communication among nodes [2]. A number of protocols are available in WSNs that minimize energy usage as well as lengthen the network's existence. WSNs define two points as follows: **Sensor Node**: SNs, also known as mote nodes, are sensor network nodes that can conduct some processing, collect sensory data, and interact with other network nodes. **Base Station (BS)**: Data is sensed by SNs and passed over to BSs for further processing and storage. As part of WSN, BS deployment is very important for energy conservation, SNs coverage, and reliability issues. Battery replacements for SNs are not possible [3]. A number of protocols are available in WSNs that minimize energy usage as well as lengthen the network's existence. GAF is a routing protocol that uses less energy [4]. Figure 1 shows a typical WSN.

2 Literature Review

Kaur (2020) proposed a novel clustering protocol named DEEC as well as a priority queue for balancing the energy within the WSN and prolonging the network duration [5]. Using the distributed energy-efficient clustering technique and the priority queue, the packet drop rate was reduced. The dropped packet was initially stirred in the priority queue, from which it was obtained and transmitted to the next CH with sufficient energy to complete the transmission. Furthermore, the energy consumption was alleviated while transmitting the packets considering the distance amid the nodes and CH. The suggested simulation included the nodes that would die at different intervals or rounds. Compared with other traditional protocols DEEC, the proposed approach performed significantly better in simulation. By applying the suggested approach, the result was enhanced by about 20%.

Yadav et al. (2020) presented the correlations amid hub segments and energy [6]. The EESW method has been proposed in WSN to address energy conservation and

life expectancy concerns. A modified network energy model, a location-aware variable, and a novel trigger technique were used to develop this algorithm. The energy consumption function and threshold function were put forward through the introduced algorithm for WSN. The NS2 simulator was employed to conduct simulation for the efficient energy awareness scheme, and it was compared with the conventional model concerning various metrics including duration of network, E2E delay, and PDR. Compared to the conventional AODV protocol, the presented algorithm showed superior performance.

El Khediri et al. (2020) recommended a new protocol named MW-LEACH for WSN [7]. CHs were selected according to residual energy, distance among CHs, and the ideal number of nodes within each CH. Creating an initial set CH candidates focused on selecting nodes with high remaining energy nearer to the density center. Afterward, the candidates were sent to various locations to gather the information from their members, then this information was transmitted to a sink. This protocol was less complex concerning time and message. Additionally, it provided a longer life expectancy for the network and a proper fault tolerance level. An experimental investigation revealed that the recommended protocol outperformed existing protocols on PDR, duration, and latency measures.

Shah et al. (2020) intended a strategy known as the DBDDCA algorithm. The long-distance nodes were transmitted from CH in relatively less time for saving the energy in this algorithm [8]. On the other hand, these nodes were broadcasted for a higher time in case of closer distance to the CH. The comparison of intended DBDDCA was done with the other strategies named LEACH and two MAC protocols. Certain network metrics were utilized for quantifying the performance of the intended and state-of-the-art strategies. These performance metrics were computed in diverse network settings such as the increased number of nodes, rounds. Outcomes of the simulation represented that the intended strategy outperformed the state-of-the-art strategies in terms of various network components.

Hassan et al. (2020) presented an extension of energy-efficient clustering protocol called IEECP to lengthen the serving period of the IoT depend on WSNs [9]. The protocol presented in this work was composed of three consecutive segments. For overlaying even clusters, the algorithm originally found an optimum number of clusters. Later, using a redesigned fuzzy C-means algorithmic technique, this algorithm developed balanced-static clusters. For this purpose, this algorithm was further combined with a mechanism to cut down and balance the energy consumed by the sensing devices. Finally, the selection of CHs was made in the best positions. Specifically, the new protocol improved the formation of clusters by reducing and balancing the energy consumed by the SNs. This protocol was best suited for the networks requiring a prolonged period of service. The outcomes-based on assessment confirmed that the new protocol outperformed the state-of-the-art protocols based on network service time.

Yeh et al. (2020) put forward a fresh meta-heuristic algorithm called bi-objective simplified swarm optimization (BSSO) method which was inspired by the ideas of routing [10]. This work performed a test based on ten WSN standards extending from 100 to 1000 sensors to depict the presentation and suitability of the presented BSSO.

In order to do tests, this work considered eight dissimilar parameter scenarios. This work further compared the BSSO algorithm against NSGA-II which was most of the accepted algorithms for solving multi-objective issues. The outcomes demonstrated that the presented BSSO could fruitfully meet the objective of this research.

Tong et al. (2020) presented an energy efficient, QoS-assured roadmap managing scheme. This approach, first of all, initiated a novel QoS model to enhance the QoS estimation [11]. After this, an effectual service selecting scheme was presented depending on this QoS model that considered both energies as well as the QoS factor. In addition, this work also presented an optimization schema for achieving balance in the consumed energy. Eventually, this work obtained an efficient local service selection. The results of tested assessments depicted the potential of the introduced program in order to improve energy efficiency and extended access service time.

Abderrahim et al. (2019) presented an energy-efficient multi-hop transmission method for WSNs on the basis of the Dijkstra algorithm. A WSN that was set up with N SNs had been taken into account [12]. At first, the SNs were assembled into clusters in accordance with their position in the monitoring region. Afterward, a suitable node was selected as CH and other nodes were classified as working nodes and resting nodes for organizing the nodes in each cluster. The set of reliable relays that was cooperated for forwarding the data with the least transmission power had been chosen. The presented transmission method was capable of enhancing the alleviation of power consumption in comparison with the earlier transmission method in an efficient way. The outcomes of the simulation demonstrated that the presented novel algorithm was adaptable to enlarge the energy savings for extending the duration of the network.

3 Methodology

This section represents the network model and energy model used in proposed algorithm. The network operation of proposed algorithm is also described in this section.

3.1 Network Model

There are few assumptions taken into consideration to implement the proposed algorithm. These are as follows:

1. All the sensor nodes are randomly distributed.
2. Base Station is deployed outside the network.
3. The location of the base station is constant.
4. All the sensor nodes are homogenous in nature.

5. Received signal strength indicator (RSSI) is used to determine the distance between two nodes.
6. The identification of sensors is unique.
7. The initial energy level of sensors is fixed.
8. The initial energy of SNs and gateway nodes is same.

3.2 Energy Model

The transmitter, power amplifier, and receiver are all part of the radio model. The transmitter relies on energy to run its circuitry. To execute transmitter circuitry actions, the transmitter consumes energy. At the time of data transmission to the receiver, the power amplifier consumes energy. During receiving information, the receiver uses energy to execute receiver circuitry activities. Free-space and multi-route propagation models are the two types of propagation models. Based on the distance between transmitter and receiver, the both free-space propagation model (d^2 power loss) and the multi-path propagation model (d^4 power loss) have been used. The energy used by sending a message over a wireless link between v_i and v_j is equal assuming that the channel is perfectly symmetrical. v_i and v_j are the SNs. If the communication distance $d(v_i, v_j)$ from vi to vj is longer than threshold distance d_0, i.e., $d(v_i, v_j) \geq d_0$, the multi-path fading ($d^4(v_i, v_j)$ loss of consumption) channel model is employed otherwise, a free-space (d^2 (v_i, v_j) consumption loss) channel model has been used. Figure 2 shows the energy model for the proposed work. As a result, for sending an l-bits message over $d(v_i, v_j)$, the energy consumption is as follows $\left(E_T l, d\left(v_i, v_j\right)\right)$:

When $d\left(v_i, v_j\right) < d_0$

$$E_T = l \cdot E_{\text{elec}} + l \cdot \epsilon_{fs} \cdot d^2\left(v_i, v_j\right) \tag{1}$$

When $d\left(v_i, v_j\right) \geq d_0$

$$E_T = l \cdot E_{\text{elec}} + l \cdot \epsilon_{\text{amp}} \cdot d^4\left(v_i, v_j\right) \tag{2}$$

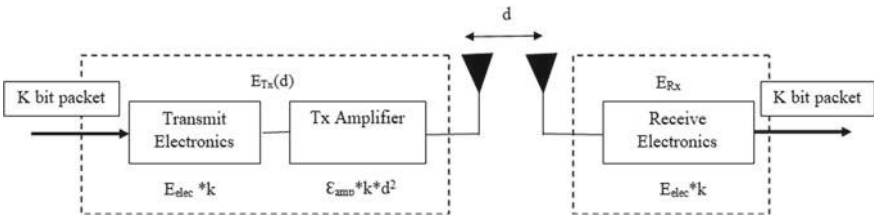


Fig. 2 Energy model

The following formula can be used to compute the energy used to receive the *l*-bits message across *d*(*vi*, *vj*):

$$E_R(l) = l.\ E_{\text{elec}}, \tag{3}$$

where,

l message of $l_{\text{bits}}$to be broadcasted over $d(v_i, v_j)$,
E_{elec} The signal's digital coding, modulation, filtering, and spreading all cause circuit loss.
E_{amp} Amplification energy of the transmitter.

The energy coefficients of power amplification for two distinct channel types are

$$\epsilon_{fs}.d^2(v_i, v_j) \quad \text{and} \quad \epsilon_{\text{amp}}.d^4(v_i, v_j)$$

3.2.1 Network Operation of Proposed Algorithm

In this section, the proposed algorithm for reduction in energy consumption due to energy holes near to sink region has been described. The proposed method consists four phases, namely initialization, cluster formation, cluster head selection, and data transmission. Initial set up for the SNs has been done in the initialization phase. Cluster formation is performed in cluster formation phase. The process for selection of CH is discussed in cluster head selection phase. Data is transmitted to sink and is performed in this phase. The flowchart of these phases are diagrammatically illustrated in Fig. 3.

A. ***Initialization***

In this phase, SNs are randomly deployed in the (200 × 250 m^2) monitoring area. The sink is located in the (150 × 250) position. The homogeneous type SNs are used in the proposed method. This phase also includes the deployment of the four gateway nodes. The gateway nodes are deployed at one hop near to the sink. The gateway nodes are the nodes that do not sense any information, it will just forward information which is already sensed like a hub. The gateway nodes are used in the data transmission phase. After deployment, all the information is updated. The location of the SNs is static. The initial energy is getting updated after every round. The updated information will be sent to all the neighbors. Then, the neighbors will update the information.

B. ***Cluster Formation***

Initially, the CHs are chosen at random. For the first round, they share their information with neighboring nodes. After then, cluster members are created. The data is subsequently transmitted by the CMs that calculate the distance.

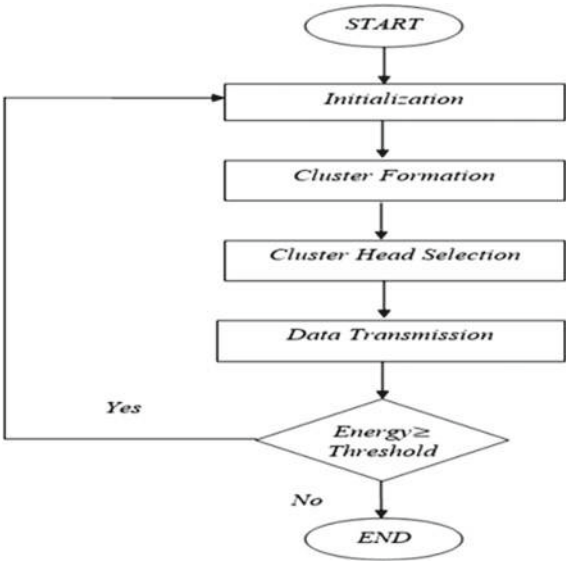


Fig. 3 Flowchart of methodology of proposed algorithm

After updating of information with neighboring nodes, the next step is formation of clusters. In the next round, the particular SN is chosen as cluster head node based on its cluster head probability.

In the proposed protocol, the selection probability of clusters is 0.1% of the SNs. The chosen cluster head node sends a broadcast message to all other SNs, including their IDs and remaining energy. The SNs used radio signal strength indicator (RSSI) for sharing the information among the neighboring SNs for formation of clusters. The CMs receive signals at high intensity. The SNs which receives with high strength signal they becomes the CM of that particular CH. Signals with great strength come from close range, whereas signals with low strength arrive from a long distance.

The distance of the CMs has been calculated from their respective CHs. For example, the distance among SNs are calculated from Euclidean distance. The CM location is x_i and y_i. The location of the sink is x_j and y_j. To calculate the distance between CM and CH the equation of this as follows:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{4}$$

C. ***Cluster Head Selection Process***

In proposed algorithm CH selection process, selects two kinds of nodes, namely CH and leader node. CHs are used to collect the sensed information from CMs and forwards the information. Secondly, leader node is used to collect the information from CHs and forwards this information to gateway nodes.

In the first round, CHs are selected randomly. The CH is chosen in subsequent rounds based on two parameters: distance and lingering energy. The SNs with

maximum energy and least distance to the sink are eligible for selection of CH. The proposed protocol selects CH based on the distance and lingering energy of a SN. The lingering energy is defined as to remain or stay on in a place longer than is usual. The lingering energy parameter helps select the CH dynamically in the network after each round. Figure 4 shows the flowchart of cluster head selection process.

The first step of the CH selection process, CH transmits a Hello message with its ID. By measuring the signal strength, the sink is able to determine its distance from a SN. All SNs transmit the INITIAL-MSG, which includes the presence of their IDs and the distance to the sink for each SN. For the calculation of CH per round, radius of the CH, node distance from the sink, distance of the nearest nodes and sink,

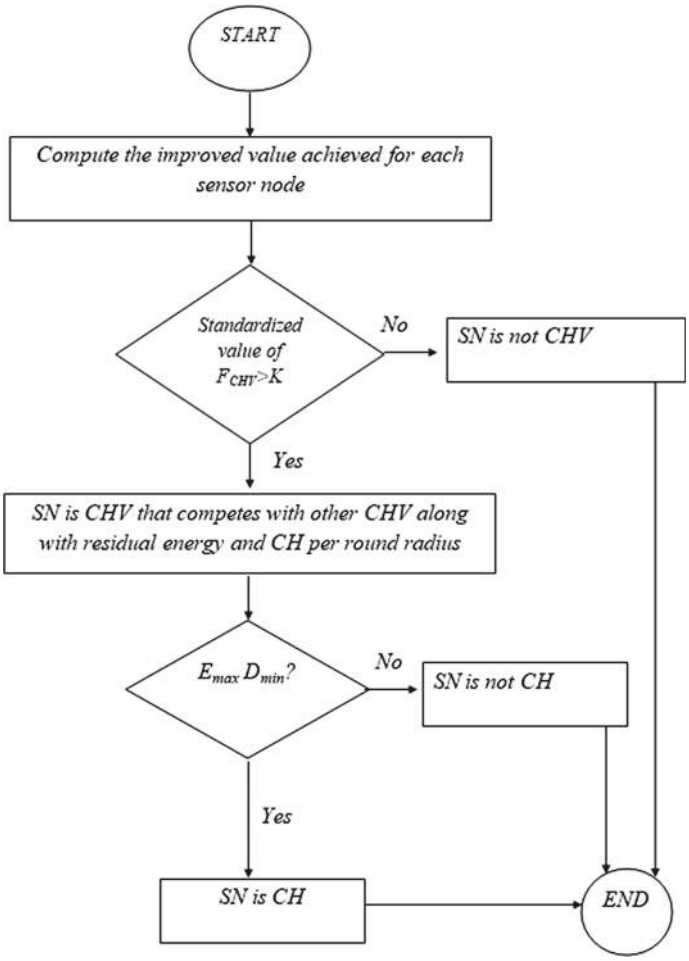


Fig. 4 Flowchart of cluster head selection process

distance present inside the farthest node, and sink are used [13].

$$R_{\mathrm{CH}} = R_{\min} * \left[1 + \left(\frac{d_{\mathrm{BS}} - d_{\mathrm{BS\,min}}}{d_{\mathrm{BS\,max}} - d_{\mathrm{BS\,min}}}\right)\right] \tag{5}$$

R_{CH}	Cluster head per round.
$R_{\min}$	Radius of cluster.
d_{BS}	Nodes distance from sink.
$d_{\mathrm{BS\,min}}$	Distance of the nearest node and sink.
$d_{\mathrm{BS\,max}}$	Distance present inside the farthest node and sink.

For the calculation of final CH, the number of neighbor nodes of certain node, mean square distance among the neighboring nodes, distance present between the sink and every node, and improved value achieved for each node are used. Each SN notices other SNs by varying the improvement value from zero to one. A random value is created for every SN that ranges from zero to one. The SN will be satisfied as a cluster if the random value generated by SN is greater, then $F_{\text{CH-value}}$. SNs are compared based on their radius and residual energy after being chosen as a CH. In Eq. (3.6), the following calculation method is used to calculate improved CH value [13].

$$F_{\text{CH-value}} = \alpha * N_{\mathrm{deg}} + \frac{\beta}{\mathrm{MSD}_{\mathrm{deg}}} + \frac{\gamma}{d_{\mathrm{BS}}} \tag{6}$$

α, β, and γ	Constant weights which have values varying from zero to one.
N_{deg}	Radius R_{CH} of the number of neighboring nodes.
$\mathrm{MSD}_{\mathrm{deg}}$	Mean square distance among the neighboring nodes.
d_{BS}	Distance present between the sink and every node.
$F_{\text{CH-value}}$	Improved value achieved for each node.

The second level of SNs are the leader nodes. The selection of leader nodes within the network is the second phase of the proposed technique. The SNs that are not selected as CHs could also be selected as leader nodes. The data is collected from SNs and passed to the CH with the help of these leader nodes. The leader nodes are responsible to gather the data from the SNs and pass the sensed data to the CH. The nodes which have the maximum energy but do not have the least distance to sink is considered as a leader node. Figure 5 shows the flowchart of leader node selection.

D. *Data Transmission*

There are five phases in the data transmission phase as shown in Fig. 6. These five phases play a role in data transmission phase. The SNs forwards information to respective CHs, CHs send information to the leader node. The leader node pass the collected information to the gateway nodes which further pass the information to the sink. The gateway nodes do not sense any information, it will only forward information to the sink.

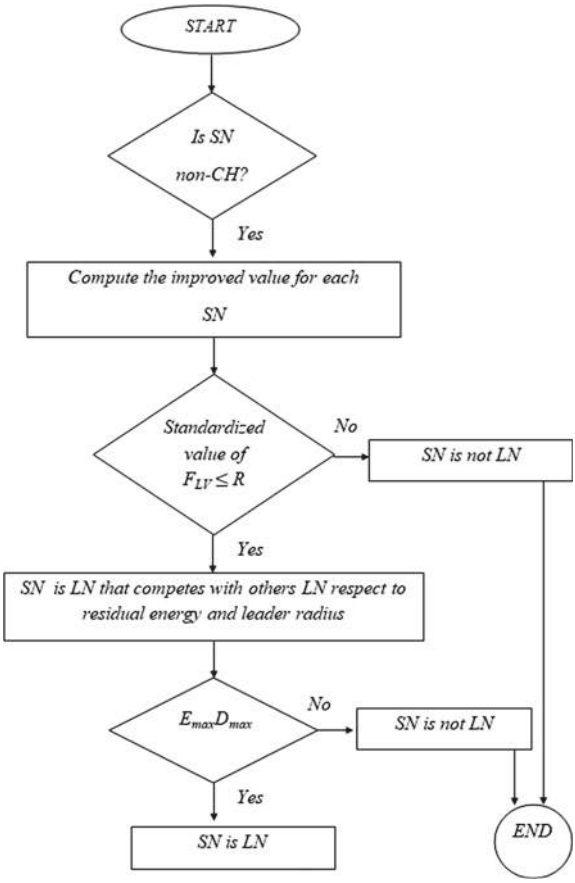


Fig. 5 Flowchart of leader node selection

There are four gateway nodes used in the proposed algorithm. In order to determine path from CH to gateway, the distance and lingering energy are used. For calculating the distance from CH to gateway nodes, first of all distance of CH to the leader node is calculated, then leader node to gateway nodes distance is calculated.

The proposed artificial bee colony (ABC) technique is applied for path establishment. The MATLAB has inbuilt functions which are used for the path establishment using the ABC technique. In the implementation, the cacheable function for the path establishment has been described. For path establishment, this function has taken the *x*- and *y*-coordinates of the sensor node, which is the source node as input, as well as the *x*- and *y*-coordinates of the destination.

The colony of artificial bees in ABC is made up of three types of bees: employed bees who are associated with certain food sources, onlooker bees who are observing the employed bees dance within the hive to select a food source, and scout bees that are looking for food sources at random. Onlookers and scouts are both referred to as unemployable. Group of employee bees will search for food sources in their

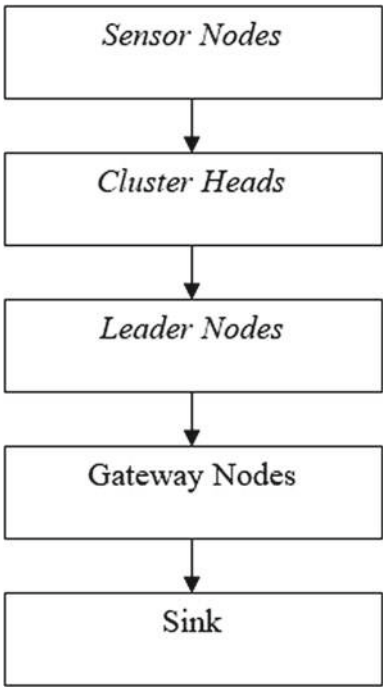


Fig. 6 Flowchart of data transmission phase

neighborhood and store the locations in the memory. Employee bees inform their searching to onlooker bees. Onlooker bees will look for the food resource in the neighborhood of the previous search informed by employee bees, and gets best possible solution from it. The onlooker selects a food source based on the likelihood value attached to that food source.

Onlooker bees eagerly await the opening of the dance floor so they can observe more nectar-rich food sources. After the CHs have been chosen, the employed bees are in charge of collecting nectar (information) from the CHs. The data is subsequently forwarded to the leader nodes. The parameters are examined on a regular basis, and if one node has less nectar than another, the role is assigned to other nodes depending on the target function. A higher nectar value indicates that the node meets more criteria.

In the utilized bee phase of the ABC method, the source and destination are selected first. The source of the data is where it is stored, and the destination is where it will be transferred. The several paths are checked in the onlooker phase from source to destination. After checking, the paths minimum distance is computed. The path is chosen depending on distance and energy. After checking all the combinations, the best value has been calculated. Scout node represented as a leader node.

The proposed technique ensures that data transfer is uninterrupted and free of congestion. Because each node's energy levels are constantly monitored, data packets

are directed to the correct path and data is transmitted without error. The best path will also take consume least energy for the data transmission from CH to sink (Fig. 7).

Algorithm of the Proposed Scheme

1. Initialize
2. B=Base station
3. S_{alive}=Set of alive nodes in the network
4. K: Number of cluster heads
5. N_{alive}=The number of alive nodes in the network
6. S_{CH} =The set of cluster heads
7. S_{NCH}=The set of non-cluster head nodes
8. S_{NCH2}=The set of non-cluster nodes assigned to cluster
9. S_{NC}=The set of non-clustered nodes
10. S_{LN}= The set of leader nodes
11. S_{GN}=The set of gateway nodes
12. Start
13. Process:
14. Input
15. For every node in S_{alive}do
16. Send energy level to Base station
17. For every node in S_{NCH}do
18. For every node in S_{CH}do
19. If Distance (Node1,Node2)< Minimum-distance
20. Minimum distance=Distance (Node1,Node2)
21. Cluster-head(Node1)=Node2
22. End if
23. End For
24. End For
25. For every node in S_{NC}do
26. If Distance(Node1,BS) <Minimum-distance
27. Minimum distance=Distance (Node1,BS)
28. Leader-Node(BS)=Node1
29. End if
30. End For
31. S_{NCH} send data to S_{CH}
32. S_{CH} send data to S_{LN}
33. S_{LH} send data to S_{GN}

Fig. 7 Algorithm of proposed method

4 Simulation Results

The thesis work is simulated using following simulation parameters, and results shows that proposed approach effectively solves the issues of energy hole problem. Several network parameters are examined, including the number of dead nodes, the number of live nodes, data packets transmitted to the sink, and energy consumption.

In simulation, 200 m * 250 m dimension is assumed. 100 SNs are randomly deployed. Initial energy of SNs are set to 0.1 J. Total 200 simulation rounds are covered to attain results. Location BS is set to 150 m * 250 m. The results of the simulations are shown in Table 1.

Modified CTNR

Figure 8 reveals the partition of the entire network into clusters. The green color represents the CHs and transmission of the data to the cache node. Moreover, the cache nodes are utilized to send the data to the sink. This Fig. 8 shows the information that is transmitted amid CH and cache nodes with the red color.

4.1 Total Number of Dead SNs

Figure 9 illustrated the total number of dead SNs with 200 rounds in network of proposed approach conducted from simulation. With an increase in the number of rounds, the number of CHs and leader nodes decreases in each cycle. In this regard, 200 rounds are chosen to illustrate the actual distribution of WSN system.

Table 1 Simulation parameters

Description	Value
Area of network (A)	(200, 250)
BS location (BS)	(150, 250)
Number of nodes (N)	100
Initial energy (E_{initial})	0.1 J
Free-space channel model	50 nJ/bit
Multi-path fading channel model	0.0013 pJ/bit/m^4
Distance threshold (d_0)	87 m
Data aggregation energy (E_{DA})	5 nJ/bit/signal
Data packet size	4000 bits
Control packet size	200 bits

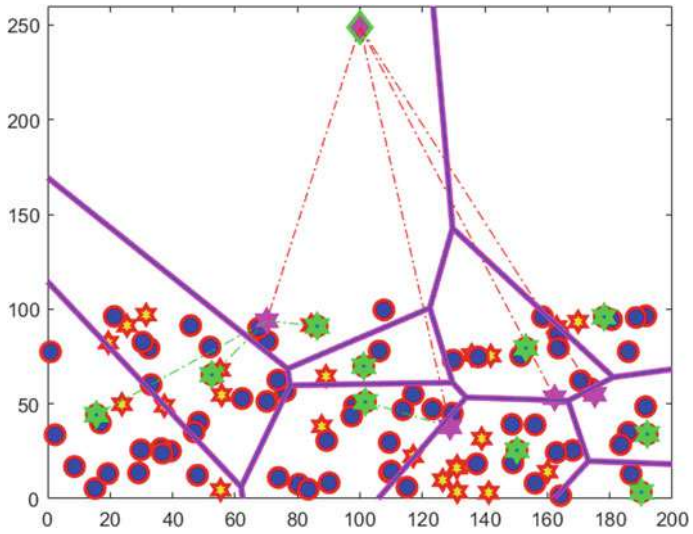


Fig. 8 M-CTNR

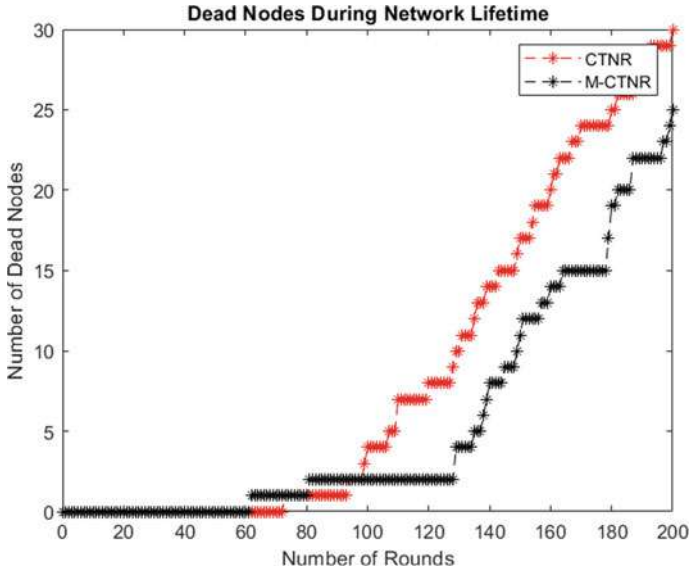


Fig. 9 Number of dead nodes

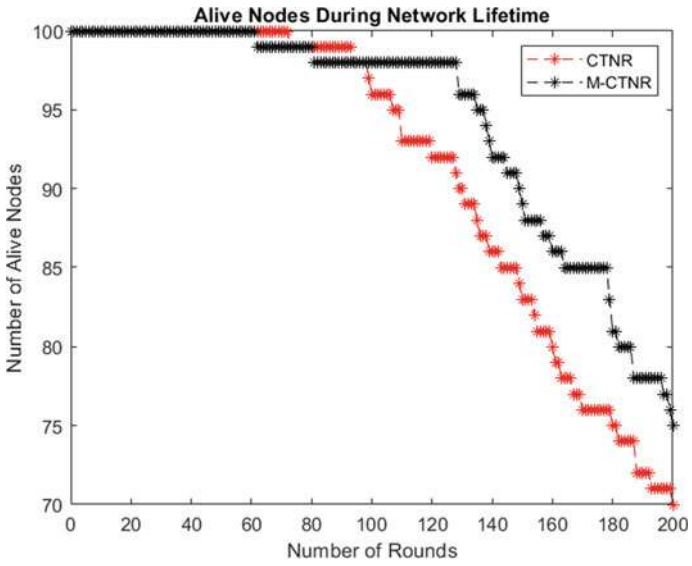


Fig. 10 Number of alive node

4.2 Number of Alive Nodes

Figure 10 the number of alive nodes is plotted against the number of rounds. From our simulation, 200 rounds are chosen to show the real distribution of the WSN system behavior in terms of the alive node as the number of rounds increase. Based on the results, traditional CTNR as the protocol has fewer alive nodes compared to modified CTNR.

4.3 Energy Consumption

Figure 11 illustrates that the base paper is compared with the suggested method. It is observed that the suggested method lessens the amount of energy consumption as compared to the other schemes. The suggested method utilizes the cache node with the cache information for some time that provides lower energy consumption using the suggested method than the existing method.

4.4 Packets Transmitted to BS

Figure 12 represented that the number of the packet transmitted to the sink using the suggested method, are compared with the base paper, LEACH and cache method. In

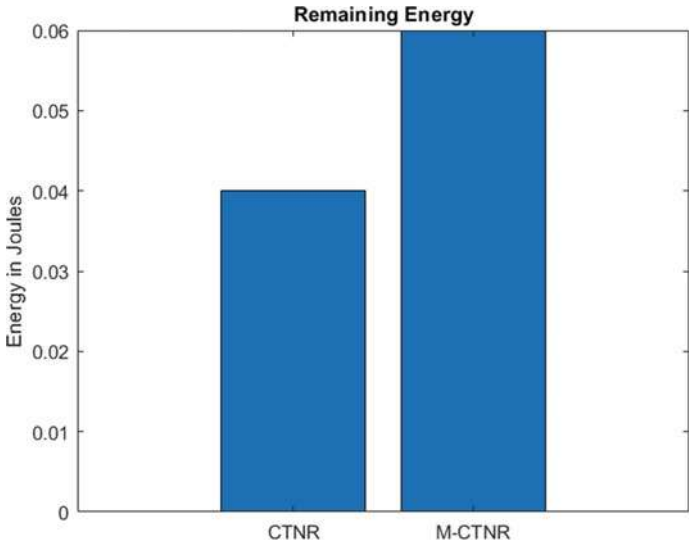


Fig. 11 Energy consumption

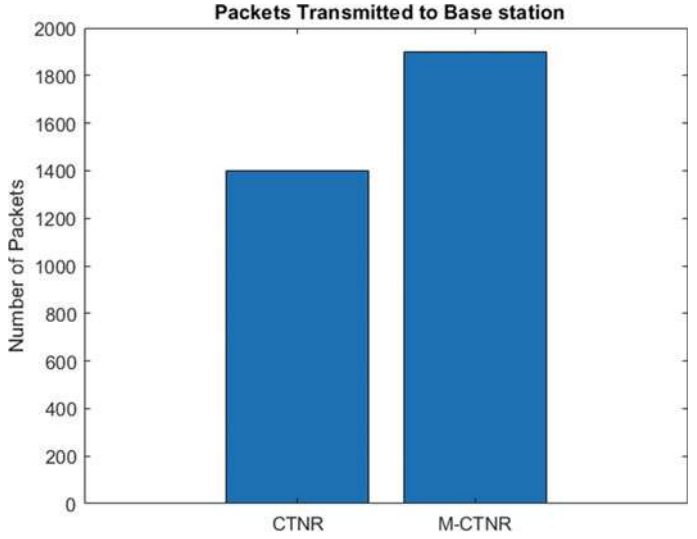


Fig. 12 No of packets transmitted

comparison with the other ways, the suggested method can send a large number of packets. As the number of dead nodes in the network decreases, more packets are sent to the sink.

5 Conclusion

The proposed technique divides the entire network into clusters through location-based clustering. In every cluster, the have been chosen depending on distance and energy. SN which has the least distance to sink and the largest amount of energy selected as the CH. The nodes which are unable to be selected as CH have been elected as leader nodes based on the energy. Finally, the network's gateway nodes have been implemented. The SNs transmit information to CH which later be transmitted to leader nodes and leader nodes have been transmit data to gateway node. The gateway node send data to sink. The proposed method is designed in MATLAB, and the outcomes are measured in relation to the number of alive nodes, dead nodes, and packets sent across the network. According to the results, the suggested technique reduces the number of dead nodes, increases the number of alive nodes, and enhances the number of packets transmitted when compared to an existing technique.

WSNs are commonly utilized in the military to detect enemy activity and communicate data to a control room. Insecure routing attacks can cause confidential information to leak, be harmed, or be altered. Data privacy issues arise when essential data is lost. The low energy efficiency of SNs, network bandwidth, and computer power make security a difficult challenge in WSN. When physical security is not an option, WSNs are used. In WSNs, data security is a major concern. The proposed approach for safe data routing will be refined in the future.

References

1. Alhogbi BG (2017) Data-centric routing in WSN for energy conservation using directed diffusion. J Chem Inf Model 53(9):21–25
2. Praghash K, Ravi R (2018) Energy consumption architecture for wireless sensor networks with different clusters. In: ICONSTEM 2017—Proceedings of 2017 Third IEEE international conference on science technology engineering & management, vol 2018 Jan, pp 414–418. https://doi.org/10.1109/ICONSTEM.2017.8261356
3. ElhabyanRS, Yagoub MCE (2014)Energy efficient clustering protocol for WSN using PSO. In: 2014 global information infrastructure and networking symposium (GIIS), GIIS 2014, pp 2–4. https://doi.org/10.1109/GIIS.2014.6934271
4. TaneeruS, Jain PC (2016) Energy efficient multilayer protocol for wireless sensor networks. In: Proceedings of 201539th national systems conference (NSC), NSC 2015. https://doi.org/10.1109/NATSYS.2015.7489118
5. Kaur K (2020) Protocol, pp 20–24
6. Yadav RS, Mishra A, Srivastava AP, Singh MK (2020) A novel energy efficient scheme for wireless sensor network, 2020 International conference on computation, automation and knowledge management (ICCAKM), pp 466–470. https://doi.org/10.1109/ICCAKM46823.2020.9051515
7. El Khediri S, Khan RU, Nasri N, Kachouri A (2020) MW-LEACH : low energy adaptive clustering hierarchy approach for WSN. IET Wirel Sens Syst 10:126–129. https://doi.org/10.1049/iet-wss.2019.0195
8. Shah IK, Maity T, Dohare YS (2020) Algorithm for energy consumption minimisation in wireless sensor network. IET Commun 14(8):1199–1208. https://doi.org/10.1049/iet-com.2019.0465

9. Hassan AAH, Shah WM, Habeb AHH, Othman MFI, Al-Mhiqani MN (2020) An improved energy-efficient clustering protocol to prolong the lifetime of the WSN-based IoT. IEEE Access 8:200500–200517. https://doi.org/10.1109/ACCESS.2020.3035624
10. Yeh WC, Jiang Y, Huang CL, Xiong NN, Hu CF, Yeh YH (2020) Improve energy consumption and signal transmission quality of routings in wireless sensor networks. IEEE Access 8:198254–198264. https://doi.org/10.1109/ACCESS.2020.3030629
11. Tong E, Chen L, Li H (2020) Energy-aware service selection and adaptation in wireless sensor networks with QoS guarantee. IEEE Trans Serv Comput 13(5):829–842. https://doi.org/10.1109/TSC.2017.2749227
12. Abderrahim M, Hakim H, Boujemaa H, Touati F (2019) Energy-efficient transmission technique based on Dijkstra algorithm for decreasing energy consumption in WSNs. In: 19th international conference on sciences and techniques of automatic control and computer engineering, STA 2019, pp 599–604, 2019. https://doi.org/10.1109/STA.2019.8717210
13. Padmanaban Y, Muthukumarasamy M (2018) Energy-efficient clustering algorithm for structured wireless sensor networks. IET Netw 7(4). https://doi.org/10.1049/iet-net.2017.0112

Chapter 10
Generating Attack–Defense Tree by Automatically Retrieving Domain-Specific Security Attack Patterns

Mohammad Aijaz, Mohammed Nazir, and Malik Nadeem Anwar

1 Introduction

Complex system organizations such as industrial control, smart transportation, and health care are gradually incorporating IT-enabled technologies to implement an information system and take advantage of operational and performance competence that outcomes from being networked. For instance, an electronic health record (EHR) system digitized the medical record system and provided longitudinal health records of a patient, an efficient computerized decision-support system, and networking. Thus, these technologies augment these complex systems and improve the communication between the stakeholders. However, the adoption of these technologies brings numerous security vulnerabilities and associated risks to the systems. These vulnerabilities lead to various security attacks compromising the security, privacy, and safety of the system. As these security vulnerabilities and attacks are often traced back to the requirement stage, and patching later in the development life cycle is more expensive, we need a more proactive mechanism to discover and analyze a relevant set of attacks as early as possible in development stage.

A number of threat modeling schemes have been proposed at the early stage of system development, such as STRIDE [1], attack tree [2], and misuse case [3]. However, the majority of them count on the knowledge and experience of security experts. The average time estimated to perform a complete threat analysis for a security analyst is found to be 160 h [4]. Most of the time in threat analysis is applied to identify relevant threat scenario, their likelihood and impact in a given context. Security analysts identify pertinent threats manually, which takes a lot of time and

M. Aijaz (✉) · M. Nazir · M. N. Anwar
Department of Computer Science, Jamia Millia Islamia (A Central University), Jamia Nagar, New Delhi, India
e-mail: aijaz15mca@gmail.com

M. Nazir
e-mail: mnazir@jmi.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_11

is also error-prone and tedious. As complex organizations gradually incorporate software and networking capabilities into their environment, there is urgent need to produce methods that help in expert decision-making process in threat modeling to their context.

In order to help security analysts in this task, MITRE and NIST contributed security repositories such as CVE [5], CWE [6], and CAPEC [7] that can be used for security and threat analysis. MITRE's Common Attack Pattern Enumeration and Classification (CAPEC) provides a catalog of known security attack patterns that an adversary may commodify or tailored to attack an IT-enabled system. These attack patterns provide various fields that can help to divide an attack into several steps that an adversary may take to exploit a system [8]. These attack patterns are useful to support analysts in investigating the software system in requirements stage without forcing them to remember numerous concepts and saving time they would devote to look for confirmation and recommendations. However, only a subset of the security attack patterns is relevant to a specific IT system. Therefore, this is critical for a security analyst to determine which attack patterns are relevant and need to be considered during threat modeling. However, manually analyzing this extensive database to find potential attacks against a specific system is cumbersome and error-prone. Therefore, there is a need to automatically retrieve pertinent security attack patterns while modeling the threats to the system.

CAPEC attack patterns, which describe an attack with natural language, are often associated with contextual information such as functional requirements. As attack patterns are the non-desirable function of the system that requires a loophole in the system to perform successfully, there is a textual relationship between attack patterns and functional requirements. However, the major challenge is that there is no direct labeling between security repositories (SRs) text data and the natural language text describing the functional model of the system. Word mover's distance (WMD), established over word2vec word embedding technique, has the power to retrieve similar documents to a query based on the individual similarity between query and documents words. This paper represents a text mining method leveraging domain-specific pre-trained word2vec and word mover's distance technique to map security repository, CAPEC, entries to a freely available natural artifact of HL7 EHR-S functional model [9] and generate ADTree using these attack patterns. The essence behind generating ADTree from CAPEC attack patterns is twofold: (1) attack patterns provide attack steps and can be seen as a branch of ADTree and (2) has sufficient and necessary knowledge contributed by many security experts. The contribution of the study can be summarized as:

- *This method automates the brainstorming process to identify the relevant security attack patterns at the requirement phase.*
- *This method mitigates the term mismatch problem in cybersecurity task by employing a pre-trained cybersecurity word2vec model.*
- *The methodology can be used to develop relevant and domain-specific attacks along with the countermeasures in the form of ADTree*

The remainder of this paper is structured as follows: Sect. 2 presents a review of related research. Section 3 describes the methodology used in this research. Section 4 demonstrates the application of the approach, and Sect. 5 presents the conclusion of the paper with a discussion of future research.

2 Related Work

In this paper, we utilize a text mining technique to retrieve attack patterns against functional requirements to generate domain-specific ADTree. Mining relevant CAPEC knowledge and attack–defense tree generation are two aspects of related work. Hence, the summary of reviewed articles is divided into two subsections.

2.1 Mining Relevant CAPEC Knowledge

It is well believed that security attack repositories can be used in early phases of secure software development as a knowledge resource. However, one of the main issues in retrieving security knowledge against text-based requirements artifacts is term mismatch problem as security databases contain technical terms that show no relationships with usual terms that specify requirements and design of the system. In this regard, Kaiya et al. [10] presented manually created term map that maps use case descriptions keywords to CAPEC jargon based on architecture assumptions. Wang et al. [11] used LSI to mine relevant CVE vulnerabilities against functional requirements to generate conformance criteria in security requirements. William [12] proposed a use case ontology-based collaborative recommender system to extract relevant CAPEC/CWE to elicit security requirements. Adam et al. [13] have investigated LDA to select relevant set of attack patterns against a text description of system model using distance metric KL-divergence. Vanamala et al. [14] also used LDA to retrieve attack patterns against text description of SRS document and used cosine similarity to calculate similarity between SRS topic distribution and CAPEC attack patterns topic distributions.

However, the above methods either suggested mostly manually tasks or consider text as a *bag of words* that shows no semantic relationships among words. In this paper, we utilized distributional vector space model (D-VSM), word2vec that trains on cybersecurity text and understand specific similarities and differences among cybersecurity text. We also used word mover's distance (WMD) as a distance metric that calculates the distance between two functional requirements text and CAPEC text as a collective distance between two nearest embedded words available in documents, therefore preserving the syntactic and semantic relationship among different words. Thus, our method provides an automatic process to retrieve relevant attack patterns against each functional requirement. Many of the above methods generate misuse

case model using attack patterns. Misuse cases, however, are unable to fully exploit attack pattern knowledge since attack patterns split an attack into many phases. Therefore, we used ADTree to fully utilize CAPEC knowledge.

2.2 Attack–Defense Tree

Attack tree-based threat modeling has received much attention by researchers and practitioner over last two decades due to its user-friendly and intuitive visualization as well as rigorous qualitative and quantitative security analysis[15–17]. Attack–defense tree (ADTree) proposed by Kordy at el. [18] is an extension of attack tree. ADTree has two different nodes: attack node and defense node. Attack node represents attack step or refinements and defense node represents mitigation or countermeasure representing corresponding action using label within nodes. It provides graphical representation as well as formalization using mathematical semantics. There are also various algorithms available for quantitative analysis over ADTree. However, a little attention is given to relevant security knowledge extraction to represent particular actions. In this work, we automatically retrieve pertinent CAPEC knowledge against functional requirements of the system and generate ADTree for each functional requirements and identified attacker goals.

3 Methodology

This section describes a summary of our approach. The overview of the proposed method is depicted in Fig. 1. The proposed method includes the five steps:

1. The preprocessing of text data.
2. Mining of attack patterns.
3. Filtering and classification of attack patterns.
4. Attack–defense tree generation.

3.1 The Preprocessing of Text Data

The dataset is cleaned and prepared for further processing in this stage. CAPEC dataset offers an enumeration of attack patterns. For each attack pattern entry, CAPEC provides ‘ID,’ ‘Name,’ ‘Description,’ ‘Prerequisites,’ ‘Typical severity,’ ‘likelihood of attack,’ and other important fields to assess the attack patterns. In this research, we use the ‘Description’ field for similarity matching against FRs. A description of attack patterns explains the sequence of events that occurs during a successful attack providing sufficient information to perform text mining.

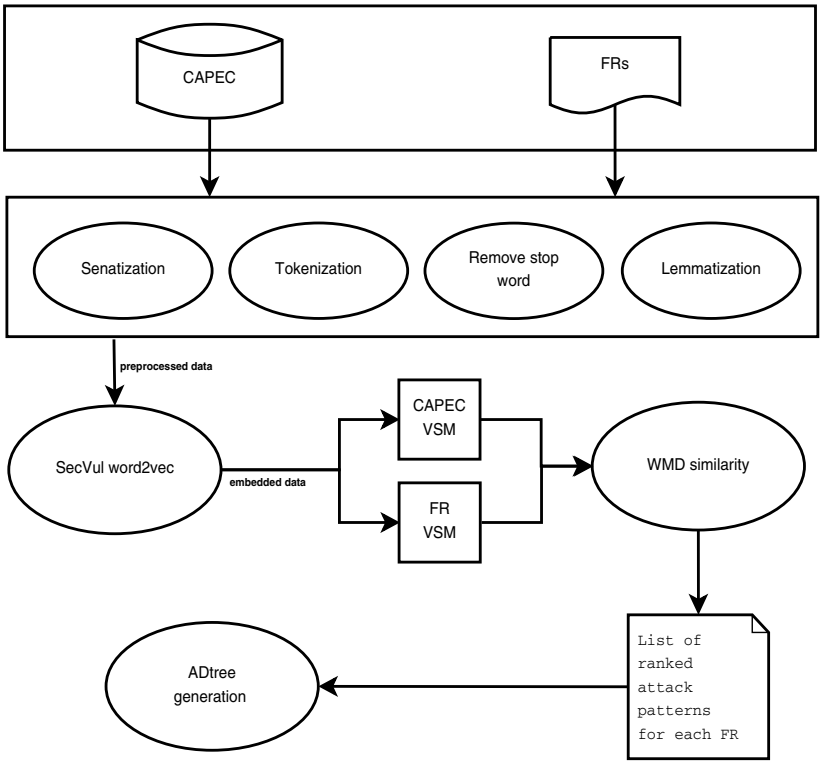


Fig. 1 Proposed methology. VSM: vector space model, WMD: word's mover distance. FR: functional requirement

The length of the 'Description' field in the CAPEC database varies depending on the attack pattern as shown in Fig. 2. As a consequence, retrieval of the attack pattern is biased. To mitigate this, we normalized the length of documents so that retrieval of attack patterns unbiased to the length of the document. There are total of 541 attack patterns in CAPEC. First, we removed the missing value while preprocessing attack patterns. Only two attack patterns have no 'Description' field, thus these two attack patterns are removed, and the remaining 539 attack patterns are used for further preprocessing.

Special letters and numerals are also removed since they offer no value to text data similarity. Then, English stop words like 'a,' 'the,' 'for,' and 'then' are eliminated because they are not useful in the text-similarity process. Aside from common stop words, the CAPEC database has a list of frequently recurring terms that appear in virtually every text. Figure 3 depicts the top ten most common such phrases; these words are useless in mining attack patterns. As a result, such words are removed from the dataset. We additionally lemmatized the remaining words in order to transform each word to its root form, reducing noise, and speeding up the task.

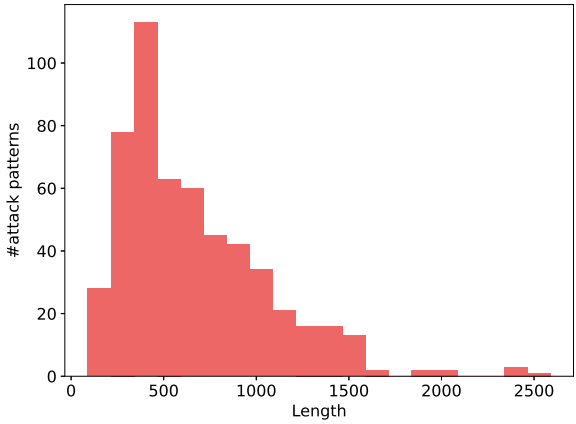


Fig. 2 Length of *Description* of attack patterns

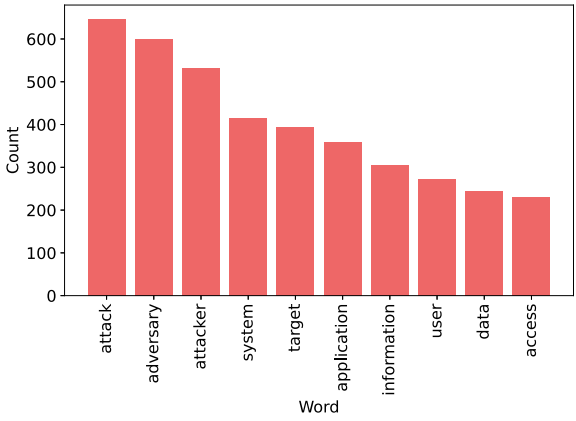


Fig. 3 Top ten most repeated words in CAPEC

3.2 Mining of Attack Patterns

We embed each word in the preprocessed data into a vector space model using word2vec [19] after preprocessing the CAPEC text data. CAPEC describes each attack pattern using technical terms that are usually not found in the text describing the requirement specification model. This problem, known as the term mismatch problem, causes two main challenges: synonymy and polysemy. Synonymy problem arises when users use different terms to describe the same object or concept. For instance, 'SQL' and 'database' are used interchangeably in software engineering. On the other hand, polysemy words are the exact words with different meanings. Therefore, to mitigate this problem, we needed a model that understands the context of the word and the relationship between technical terms and usual terms.

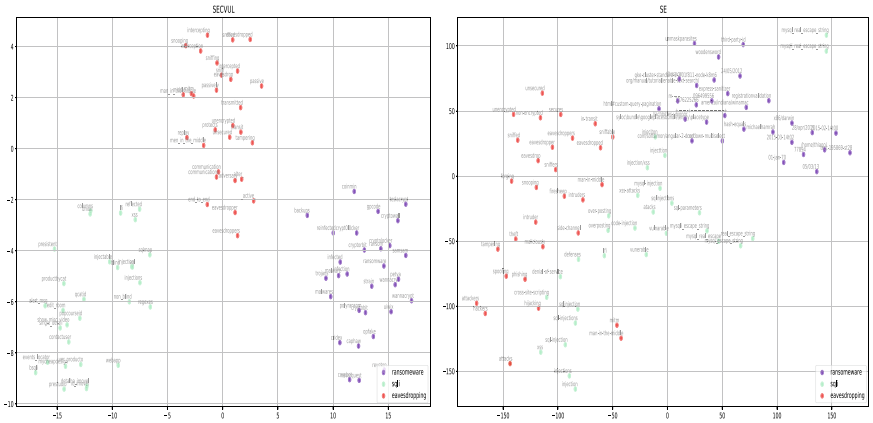
Word2vec is typically pre-trained on massive corpus of Google News data, which contains billions of words and millions of vocabulary. One disadvantage of employing a pre-trained model for domain-specific information retrieval is that the model may

Table 1 Most similar words returned by different word2vec model

Keyword	SecVul word2vec	SE word2vec	Google News word2vec
Record	Track, _lengths_,	Row, table, entry,	Equalling, mark,
	Entry, previous	employee, field	Eclipsing, equalled, recod
Display	Show, displaying, appear,	Show, displaying, diplay,	Displaying, exhibited, displays,
	Screen, render	Disply, shown	Diplay, display
click	Clicking, button, tab,	Clicking, clicks,press,	Clicking, newslinks, featuredlinks,
	Dialogue, menu	Tap, clic	Thislinks, forgot_password
Data	Information, sensitive, encrypted,	Values, database, records,	Dataset, database, statistics,
	Serialized, transfer_key	Information	Information, data_system_IPEDS

be trained on data that has no relevance to the domain, resulting in ambiguous findings. In order to overcome this problem, researchers trained the word2vec model on a specific domain, such as biomedical [20], software engineering [21], and cybersecurity [22] as well as for specific tasks in software engineering [23, 24]. We compared the varied meanings of terms in each model and carefully evaluate several pre-trained models for obtaining domain-specific security attack knowledge. Table 1 compares three word2vec models trained on (1) SecVul [22], (2) software engineering question answering website stack overflow dataset (SE) [21], and (3) general word embedding trained on Google News texts. We tested each word embedding using different keywords as our problem required a specific word embedding in the cybersecurity and software engineering domain. Because the Google News word2vec delivers more generic terms in response to provided input words, it is ineffective for our problem. As stated in Table 1, the stack overflow website is a Q&A website supplied by a variety of developers (from professionals to novice) and may contain several syntactical mistakes. For example, for the word 'display,' it gives 'diplay' and 'disply' as synonyms. SecVul word2vec is a cybersecurity-specific text vector space model (VSM) that has no syntactical mistakes as it is trained on cybersecurity text data taken from multiple security repositories. Furthermore, Fig. 4a depicts the word distribution in SecVul word embedding for three different types of attacks: ransomware, sqli, and eavesdropping. It clearly distinguishes between different types of attacks and embeds the appropriate words. Figure 4b depicts the same attack vocabulary distribution in software engineering VSM, which fails to distinguish cybersecurity content for distinct types of attacks. As a result, the SecVul word2vec model was chosen as our word embedding model in this technique.

Functional requirements are mainly represented in the form of use case description, and an attack pattern can be represented as negative use case (misuse case) to the system. Attack patterns are the system's undesirable functions that, in order to be effective, need a vulnerability in the system. In order to build an attack patterns retrieval system against FRs, we examined word mover's distance (WMD) similarity function that works on the word2vec model, as the literature [25, 26] finds it perform-



(a) SecVul word2vec (b) SE word2vec

Fig. 4 Word distribution in word2vec trained on security text (**a**) and software engineering text (**b**) of different types of attack (ransomeware, eavesdropping, sqli)

ing better than existing methods. WMD has a power to retrieve similar documents to a query based on the individual similarity between query and documents words. We used WMD (leveraging pre-trained word2vec) to retrieve similar documents (attack patterns) to a query (functional requirement).

3.3 Filtering and Classification of Attack Patterns

The aim of this step to filter out from further processing those attack patterns that are not relevant to the queried requirements. For attack patterns that are considered as relevant should fulfill the following criteria: Retrieved attack patterns threaten the process involved in FR. The model was run by each of the three authors separately, and the relevant and irrelevant patterns were highlighted. Select appropriate attack patterns for each functional requirement based on mutual agreement. Following that, we manually determine the attacker aims and motivations by studying attack patterns, and we classify each attack pattern against functional requirements and attacker goals.

3.4 Generating Attack–Defense Tree

After the classification of retrieved attack patterns, we develop an attack tree model for each functional requirement. We first extract ‘Execution of Flow’ and ‘Mitigation’ attributes of attack patterns and generate attack–defense tree for each identified attacker’s goal against each requirements.

4 Application of the Proposed Approach

The EHR system is used as a case study in this paper. An EHR system uses information technology in the healthcare system to create a longitudinal collection of health records that allow an authorized individual rapid access to health information. It also has a decision-support design that enhances the safety, quality, and efficiency of patients in healthcare environments. Because the EHR is a complex information system, adopting such a strategy necessitates domain-specific and thorough threat modeling in order to avoid or withstand undesirable and hostile occurrences.

The Health Level Seven (HL7) standard development organization encourages the establishment of an EHR system in the healthcare setting from the beginning. The HL7 healthcare standard is a widely acknowledged platform that outlines the functional requirements for electronic health records in detail. To obtain relevant attack patterns from the CAPEC database, we chose this natural language artifact as a piece of domain knowledge. The HL7 EHR version 2.0.1 functional model establishes a common model of EHR system functionalities. This model contains detailed information on the set of functions that may be present in an EHR system. This functional model is used to extract the functional requirements for our proposed method. We ran our model on each of the extracted functional requirements, treating each as a query to model. The extracted attack patterns are collected for further processing.

On the basis of our filtering criteria, we compute mean reciprocal rank (MRR) and mean average precision (MAP) for each extracted attack patterns. MRR is the simplest metric in recommender system that computes the first occurrence of the relevant document in the output of the retrieval system. Due to more than one attack pattern being possible for a functional requirement, we compute MAP for our model, which measures average precision for each relevant item across the retrieved list and averages to get the MAP for all inputs. Figure 5 shows the MAP calculation procedure for ‘Originate and retain record entry’ and ‘Manage clinical history’ FRs.

We investigated two pre-trained word2vec model (w2v SecVul and w2v SE) with the WMD similarity matrix and the cosine similarity.[1] Table 2 shows a comparison result among these techniques. Results show that word2vec trained on cybersecurity text perform significantly better than word2vec trained on software engineering text. We also noticed that WMD perform slightly better than cosine similarity. As a result, we consider the security attack patterns extracted from the SecVul word2vec model with WMD in greater detail. However, this is a subjective measurement evaluated by authors and is not to be taken as ‘best’ recommendation.

Moreover, we acknowledge that there is a possibility of false negative that means some relevant attack patterns may miss by our model as we are only concerned with the top retrieved attack documents. To that end, we expect more robust combination

[1] Note: Cosine similarity is measured using average vector of a document taking each word in document as a vector.

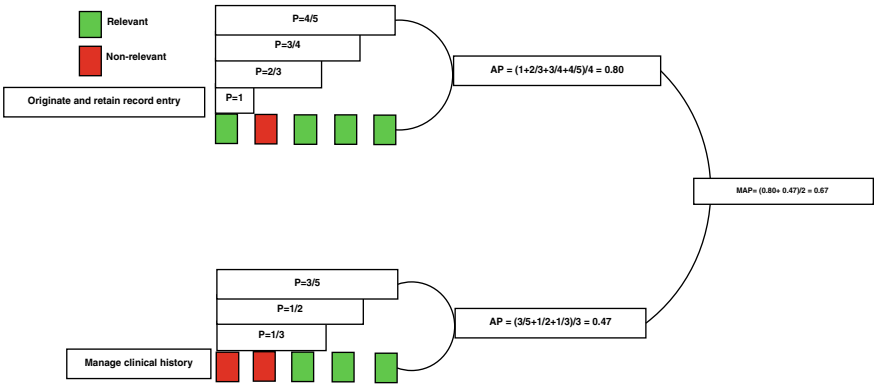


Fig. 5 MAP calculation for relevant attack patterns. P: Precision, AP = Average precision, MAP= Mean average precision

Table 2 Performance comparison of pre-trained word2vec models on cosine and WMD similarity metrics

	SecVul word2vec		SE word2vec	
	Cosine (%)	WMD (%)	Cosine (%)	WMD (%)
MRR	70	**72**	62	65
MAP	55	**60**	42	50

of retrieval techniques with appropriate labeling may improve the mapping of attacks in requirements engineering. In the future work, we intend to enhance the model by further experiments.

4.1 Mining Attack Patterns

For the purpose of assessing our model's ability to mine relevant attack patterns, we randomly selected two FRs from HL7 EHR functional model. For each requirement, we ran our algorithm that returns top 20 matches. As our aim to reduce the efforts in selecting the most relevant attack patterns, we follow a simple heuristic of including the top 5 retrieved attack documents. In what follows, we discuss the results and demonstrate generation of attack–defense tree using the retrieved documents for each requirement.

FR F1: Manage Clinical History In this FR, EHR system manages the patient's sensitive and private information including personal and health-related data. Healthcare provider, healthcare staff, and patients can access this information based on their privileges. An attacker can try to steal, modify, or disrupt the services that manage this information. The top 5 retrieved attack patterns using this method

are *Retrieve Embedded Sensitive Data (CAPEC-37)*, *Phishing (CAPEC-98)*, *Web application fingerprinting (CAPEC-170)*, *Accessing/intercepting/modifying HTTP Cookies(CAPEC-31)*, and *Voice phishing (CAPEC-656)*. Here, CAPEC-98 and CAPEC-656 are related to social engineering attack. CAPEC-656 is a type of CAPEC-98 and has similar mechanism. An attacker may utilize these patterns to lure or trick healthcare provider to access sensitive or private information. Although both attacks are relevant in our case, we select CAPEC-98 to generate ADTree as CAPEC-656 has same mechanism. CAPEC-170 and CAPEC-31 are web-based attacks and can be used to access or modify the sensitive information stored in EHR system. CAPEC-31 is a type of sniffing attack and likelihood of this attack is high. We select this attack pattern to generate ADTree. After filtering, we classify the attack patterns according to the attacker objective. In the above-retrieved attack patterns, generated ADTree for stealing data objective is shown in Fig. 6.

FR F2: Manage Record Lifecycle This FR manages every action performed and an audit log track each record entry. An attacker may misuse/manipulate audit log or spoofed an action performed in the system. The retrieved attack patterns for this FR are *Web Log Tempering (CAPEC-81)*, *Cross Site Request Forgery (CAPEC-62)*, *Clickjacking (CAPEC-103)*, *Log Injection Tampering Forging (CAPEC-93)*, and *Enumerate Mail Exchange Records (CAPEC-290)*. The CAPEC-81 and CAPEC-93 are similar in functionality, with the later being more abstract than former and both are related to log files. We selected CAPEC-93 for further analysis. Note that, in our model, both the attack patterns are considered relevant. CAPEC-290 is irrelevant in our case as we are concerned with attack patterns that threaten the given FR that is related to user action tracking. CAPEC-103 and CAPEC-62 both are related to action spoofing Both are relevant in our case. CAPEC-103 and CAPEC-62 both are related to action spoofing and both are relevant in our case. Modify data is the most common attacker objective in this type of attacks. Finally, the generated ADTree with *Modify data* as an attacker goal for FR F2 are depicted in Fig. 7.

The results show that the most retrieved security attack patterns are related to either stealing and modifying private and sensitive information or disruption or denial of services. These results also confirm the well-known reported security breaches in health records [27]. Also, the outcome shows that maximum attack patterns fall into the category of confidentiality, which refers again to a well-known challenge in the healthcare system. Furthermore, the results should not be treated as an exhaustive set of security attacks. However, may be taken as they provide a clue to a system designer and analyst about the possible attack surfaces in the system.

5 Conclusion and Future Work

In this paper, we have proposed an approach for modeling the security attacks, which automatically retrieves attack patterns from the CAPEC database in order to generate attack–defense tree (ADTree). CAPEC attack patterns are well-documented and thoroughly analyzed in real-life exploits, as a result of which the quality of the

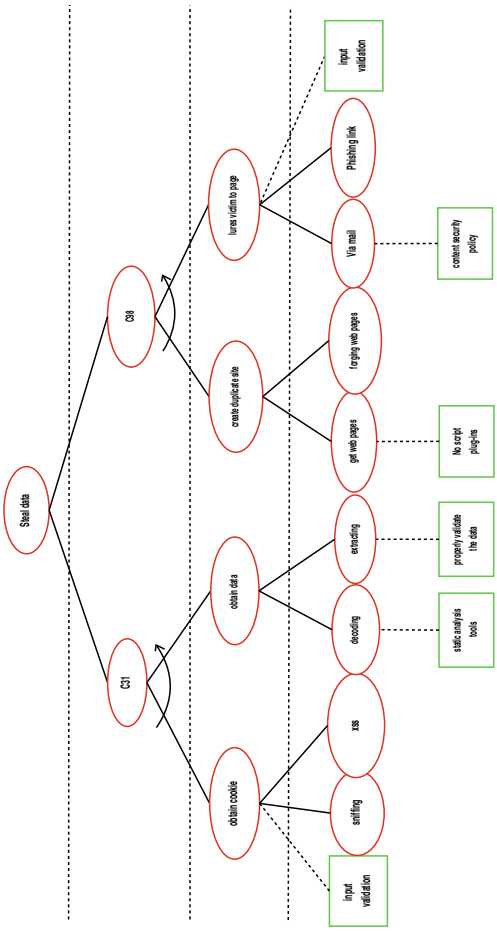


Fig. 6 Generated ADTree for manage clinical history

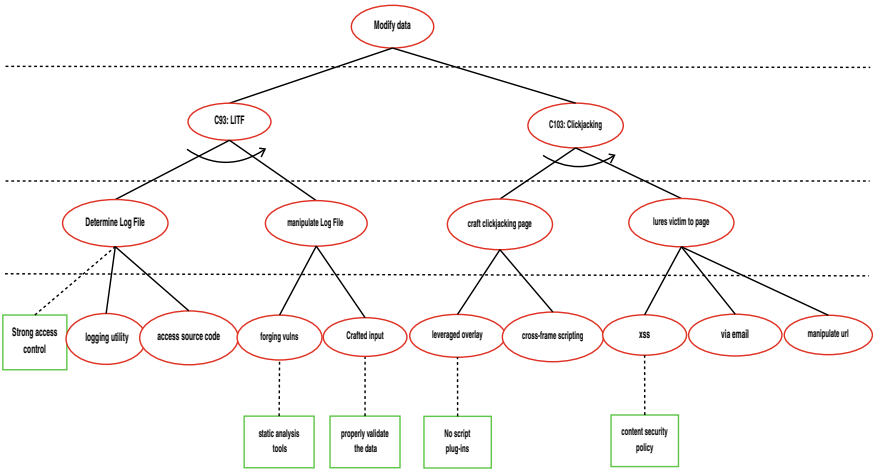


Fig. 7 Generated ADTree for manage record lifecycle

generated ADTree seems to be promising. Thus, the proposed approach provides relevant security knowledge and basic understanding for the system designer in the security domain that enables them to efficiently and effectively identify pertinent attacks at the initial phase of SDLC. Furthermore, the approach may also be helpful to visualize the security threats and associated risks to the system; this is projected in the form of an attack–defense tree that may guide us to visualize and uncover subtle attacks efficiently.

The results in the paper may not be generalized because we have applied the approach only to a subset of functional requirements considering an electronic health record system as a case example. Nevertheless, the work remains valid within the specified goals and objectives. However, the approach may be validated on a larger set in the future for better realization, and then it may be generalized. There are certain limitations, and the retrieved attack patterns need to be validated against expert knowledge for qualitative assessment, which is a subjective task leading to various interpretations. Moreover, some of the CAPEC database values are found missing, which is also a limitation in itself. In the future work, we plan to develop a comprehensive risk assessment framework based on the CAPEC database and ADTree that may overcome existing shortcomings.

References

1. Shostack A (2008) Experiences threat modeling at microsoft. MODSEC@ MoDELS 2008
2. Bruce S (1999) Attack trees. Dr. Dobb's J 24(12):21–29
3. Sindre G, Opdahl AL (2005) Eliciting security requirements with misuse cases. Requirements Eng 10(1):34–44

4. Whitmore J, Turpe S, Triller S, Poller A, Carlson C (2014) Threat analysis in the software development lifecycle. IBM J Res Dev 58(1):6:1–6:13
5. Mitre (2021) CVE–CVE
6. CWE (2021) CWE—Common Weakness Enumeration
7. MITRE Corporation (2021) CAPEC—Common attack pattern enumeration and classification (CAPEC)
8. Barnum S, Sethi A (2007) Attack patterns as a knowledge resource for building secure software. In: OMG Software Assurance Workshop: Cigital, pp 1–31
9. Dickinson G, Ritter J HL7—Electronic health record system functional model, Release 2.0.1 Anneke Goossen-Baremans MScN
10. Kaiya H, Kono S, Ogata S, Okubo T, Yoshioka N, Washizaki H, Kaijiri K (2014) Security Requirements analysis using knowledge in CAPEC. In: Lecture notes in business information processing, pp 343–348
11. Wang W, Gupta A, Niu N (2018) Mining security requirements from common vulnerabilities and exposures for agile projects. In: Proceedings—2018 1st international workshop on quality requirements in agile projects, quaRAP 2018, pp 6–9
12. Williams I (2018) An ontology based collaborative recommender system for security requirements elicitation. In: 2018 IEEE 26th international requirements engineering conference (RE). IEEE, pp 448–453
13. Adams S, Carter B, Fleming C, Beling PA (2018) Selecting system specific cybersecurity attack patterns using topic modeling. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). IEEE, pp 490–497
14. Vanamala M, Gilmore J, Yuan X, Roy K (2020) Recommending attack patterns for software requirements document. In: 2020 international conference on computational science and computational intelligence (CSCI). IEEE, pp 1813–1818
15. Lallie HS, Debattista K, Bal J (2020) A review of attack graph and attack tree visual syntax in cyber security. Comput Sci Rev 35:100219
16. Wojciech W, Maxime A, Barbara F, Sophie P (2019) Beyond 2014. ACM Comput Surveys 52(4):1–36
17. Hong JB, Kim DS, Chung CJ, Huang D (2017) A survey on the usability and practical applications of graphical security models. Comput Sci Rev 26:1–16
18. Kordy B, Mauw S, Radomirovic S, Schweitzer P (2014) Attack-defense trees. J Log Comput 24(1):55–87
19. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013—Workshop Track Proceedings, pp 1–12
20. Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S (2013) Distributional semantics resources for biomedical text processing. In: Proceedings of the 5th international symposium on languages in biology and medicine, Tokyo, Japan, vol 5, pp 39–44
21. Efstathiou V, Chatzilenas C, Spinellis D (2018) Word embeddings for the software engineering domain. In: Proceedings of the 15th international conference on mining software repositories, New York, NY, USA. ACM pp 38–41
22. Mumtaz S, Rodriguez C, Benatallah B, Al-Banna M, Zamanirad S (2020) Learning word representation for the cyber security vulnerability domain. In: 2020 International joint conference on neural networks (IJCNN), number July. IEEE, pp 1–8
23. Mantyla MV, Novielli N, Lanubile F, Claes M, Kuutila M (2017) Bootstrapping a lexicon for emotional arousal in software engineering. In 2017 IEEE/ACM 14th international conference on mining software repositories (MSR), number Msr. IEEE, pp 198–202
24. Ortu M, Murgia A, Destefanis G, Tourani P, Tonelli R, Marchesi M, Adams B (2016) The emotional side of software developers in JIRA. In: Proceedings of the 13th international conference on mining software repositories, New York, NY, USA. ACM pp 480–483
25. Kusner M, Sun Y, Kolkin N, Weinberger K (2015) From word embeddings to document distances. In: International conference on machine learning. PMLR, pp 957–966

26. Huang G, Guo C, Kusner MJ, Sun Y, Sha F, Weinberger KQ (2016) Supervised Word Mover's Distance. Adv Neural Inf Proc Syst 29
27. Newaz AKM, Sikder AK, Rahman MA, Uluagac AS (2020) A survey on security and privacy issues in modern healthcare systems: attacks and defenses. arXiv preprint arXiv:2005.07359

Chapter 11
Content Based Recommender System Using Machine Learning

Swati Suman, Riya, and Chandrani Chakravorty

1 Introduction

In recent times, lots of studies have been conducted for the prediction and the recommendation of movies. Many applications are into existence which recommends movies to the user such as Netflix, Hotstar, YouTube and Amazon Prime. In order to recommend the movie to the user collaborative filtering either the item based filtering or the user based filtering are certainly used. But the recommendation system is incomplete without the user ratings. So this system or the website provides user's ratings for each movie as well as it has been developed in such a way that it recommends movies according to the user's interests and also the movies which users are searching for. Users have problems in selecting the alternative movie. So they might get movies recommended by friends or by movie reviews that consume lots of time.

Currently there are a lot of recommender systems based on the user preferences, so what should be done if the website does not contains enough movies for recommendation. In order to watch a good online movies, we need to scroll through certain numbers of movies. This system will help users to search movies based on their interests and preferences.

After that, the representation of a movie has been solved, which shows how a system can understand a movie name. That is the precondition for comparing similarity between two movies. Movie features such as genre, actor and director is a way that can categorize movies. So these are the questions which can be raised:

S. Suman · Riya · C. Chakravorty (✉)
Department of MCA, RV College of Engineering, Bengaluru, India
e-mail: c.chandrani@gmail.com

S. Suman
e-mail: swatisinha5937@gmail.com

Riya
e-mail: nayan.riya9@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_12

- How to calculate the similarity between two movies
- Is it possible to set weight for each feature

Now after getting the questions it is found how exactly it is possible to map the two movies such that they are of the same genre and also match the similarity, As it is very complex and difficult for a machine to understand human's sentiments & emotions with different expressions by giving the same words but of different meanings.

Through sentiment analysis using machine learning, it will be easy for the machine to understand and tabulate the human's sentiment and emotion in three different types such as positive, negative and neutral.

2 Literature Survey

The recommendation system uses a variety of domains such as content based approach, collaborative based approach, knowledge based approach, utility based approach, hybrid based approach etc. Most of the online recommendation system for a variety of items uses ratings from previous users to make recommendations to current users with similar interests. Some of the surveys are listed below:

The comparative result states that there is an improvement in finding out the accuracy, quality and the scalability of the system. Further the author applied a hybrid approach for recommending the movies which is a web based recommendation system in which GUI can be shown by collecting user's ratings on a scale of 1–5 [1].

Based on item recommendation problems and proposed the movies which are popular in today's world and also according to the users' interests. A method called "content-based filtering" for the movies used for searching the movie. The feature such is genres had taken for the filtering purpose [2].

An alternative k-nearest-neighbour method has been applied instead of computing the cosine similarity between the two movies. This helps to find the relationships in such a way that the user who likes a particular movie will also like the same type of movie. Item to item similarity matrix has been computed by this approach [3].

Lund et. al. [4] proposed solution to improve scalability and quality of system. Further the author applied a hybrid approach by combining two methods, one is content based filtering and the other one is collaborative filtering for getting maximum accuracy. Author had also computed cosine similarity between two different movies to find out the similarity of two movies in the several given datasets which works efficiently in less amount of time and also it helped in reducing time computation of the system.

Apache Mahout Framework has been used to find out the user's rating and to suggest the rating list of users and item-based recommendation for the comparison on the basis of performance and efficiency of the system [5].

The percentage view approach for recommending movies to the users, it finds relevant movies for the customer and then compares the performance with a random

movie recommendation system for showing the accuracy of the project. The use of limitations on common items used in DSP, CFA results in better predictions of CFA [6].

A collaborative filtering approach as well as item-based method which results in more accuracy and efficiency for using the system. An item-based approach is non-dynamic in nature, so it can be applied in both online and offline mode. Further the author applied the KNN approach to find out the movies in the datasets which computes the nearest similar movie to the searched movie. The author also used cosine similarity [7].

Based on pattern recognition network and k-means clustering which is used to find out the movies with similar ratings. Further, the author had applied Neural networks to analyze the behaviour of users. He further proposed a method which can provide more accurate results and would be much more efficient to use [3].

Recommendation system with collaborative filtering and association rule mining using big data, an e-learning resource for learners using hybrid knowledge-based RS using ontology and pattern mining. Ontology is generally applied for knowledge representation. The experimental result shows that the proposed work overcomes the cold-start and data sparsity problems and also achieved better recommendation accuracy [8].

A route recommendation based on location, item and time (LIT). LIT with Prefix Span mining approach is used to generate frequent LIT sequential patterns. The experimental result ensures that the proposed approach can produce relevant visiting experiences for visitors and help managers understand visitors' behaviour [9].

3 Research Method

The proposed system adopted Client–Server based Architecture which provides interaction between the users and the system. Here, Python's Flask Framework is generating the similar movies from the Movie Datasets that has been used from Kaggle, and using Random Forest classifier for the classification of the reviews, it sends back the request generated from the server as a response (in Fig. 1).

The proposed system designed based on client server architecture (Fig. 1). The server contains HTML/CSS/JS and AJAX and it is sending request to the backend database of the TMDb API and in return it is sending the response to the server and on the other hand user can access it by sending the request to view.

As there are number of methods and algorithms for recommendation, the proposed system is using following methods/techniques so that it should meet all the requirements.

- Datasets

  Several movies' Datasets have been taken which have been collected from Kaggle websites with anonymous ratings from various genres of approx 6000 movies.

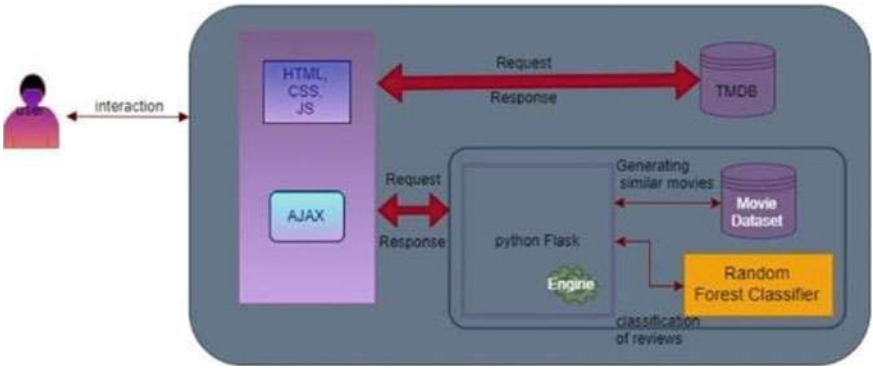


Fig. 1 Architecture between client and server [10]

Later API of TMDb has also been taken, so that, the system should cover almost all movies.

- Pre-processing

  Pre-processing has been done on the several datasets based on several features such as movie name, actor name, director name, genres and so on in which the data gets transformed in such a way that machine can easily parse it. After the pre-processing of data, features of the data can easily be interpreted by the algorithm that is used in the proposed system. The features that have been applied in this proposed system includes movie title, actor's name, director's name, genres, based on the specified features users can search for a movie and also based on these specified features recommendation will be done.

- Filtering and cosine similarity have been applied to find out the sentiment analysis and the similarity between the two movies respectively

(i) Content based filtering

Content filtering is mostly used in recommender systems and it has been designed to recommend movies to users based on user's preferences [2]. Content based has also been applied to find out the sentiment analysis of the two similar movies.

To implement Content based filtering, a set matching comparator has been applied. The genres of the two common movies will be returned by this comparator. It will merge the genres into a single set for each and every movie. In the same set, the genres of each particular movie are getting appended.

Equation for to predict the rating [4]

$$H = M * \left(\mathrm{Hr}/M'\right) \tag{1}$$

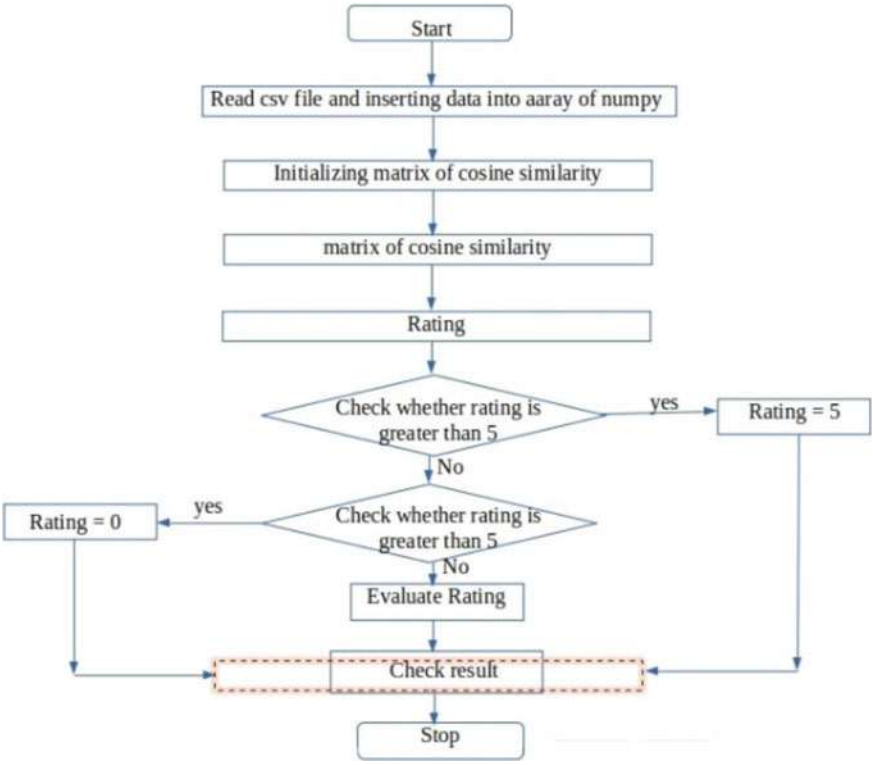


Fig. 2 Flow chart of the working of content filtering

where, Hr refers to the maximum rating that a user can assign to a movie. M refers to the number of common objects. M' refers to the maximum number of matching objects between two movies. R refers to the rating of an active movie.

Figure 2 depicts the working of content filtering where it is evaluating the ratings provided by the users. At first, it is reading the csv file/datasets and then getting data inserted into an array of numpy. Afterwards, initialize the cosine matrix. Then the system will check for rating where it first checks rating is greater than 5. If yes, then the rating will be stored as 5. If No, then the system will check for the rating is less than 5. If yes, the rating will be 0. If No, then it will evaluate the rating.

ii. Cosine similarity

Cosine similarity measures out the similarity between two vectors of a product. It is basically calculated by the cosine of the angle between two vectors and determines, if the two vectors are pointing in the same directions or not. It's in general used to measure the similarity in text analysis. It can be defined as the metric used to measure the exact similarity of the movies irrespective of their size. Mathematically, it can be concluded as the measure of the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is of great advantage because

even if the two similar movies are far apart by the Euclidean distance (cause of the size of the movie), still chances are they may be oriented closer together. Thus, the smaller the angle, higher is the cosine similarity [11].

Equation to find out the cosine similarity is as follows [12].

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} Ai \times Bi}{\sqrt{\sum_{i=1}^{n} (Ai)} \text{^}2 \times \sqrt{\sum_{i=1}^{n} (Bi)} \text{^}2} \quad (2)$$

where A and B are the vectors Ai and Bi are the components of the vectors A and B. And using the dot product of two vectors, cosine has been applied to find out the similarity. For this experiment, vector A is considered as movie1 and vector B for movie2. The cosine distance helped to analyse the similarity between two movies (in Fig. 7).

4 Result and Discussion

Data source: Data has been taken from the kaggle from which users can find or publish datasets in a web-based data science environment. Further, data has also been taken from TMDb using the API key. TMDb stands for "The Movie Database [12]. Basically, it is a movie and TV database which contains approx 679,867 movies. The size of the datasets which have been taken in the proposed system is approx to 230 Mb. After collecting the data, data has been pre-processed and also the extraction of the data has been done.

As there are several methods and techniques which can be applied to the proposed system. The two methods which we took for recommending the movies are Content based filtering and the Cosine similarity. So, the system uses two methods of machine learning. So, comparison (in Table 1) has been done on their own features to check

Table 1 Content based filtering and cosine similarity

Sl. no	Content based filtering	Cosine similarity
01	It recommends items to users based on previously rated highest items by the same user	It recommends movies to user by taking the cosine angle of the two movies
02	It works on the principle of user preferences for movies and recommends similar items-based movies	It works on the principle of similarity between the dot product of two non-zero vectors that measures the cosine of the angle between them

which one best suit the recommendation part. But after comparing this, we came to the results that it would be quite good, that both the methods together to be applied.

During evaluation of the proposed methodology, two recommended approaches have been applied i.e., Content Filtering and Cosine Similarity. So, Content based algorithm and Cosine Similarity method both are applied to find out the accurate recommendation of movies which the user is looking for based on the user's interests.

Comparison of two techniques based on their features

By concluding all the comparisons based on the features such as performance, scope, the results which came out of this is that it would be quite better to apply the hybrid approach of Content based filtering and Cosine Similarity. It would give an almost accurate result. So, the overall comparisons have been shown in Table 2 and in Fig. 3.

Comparison of Result of different Techniques

The graph (Fig. 3) is comprising the result of recommendation system built using Content Filtering (CF) and Cosine Similarity (CS) with the proposed techniques. On comparing among these two techniques, it is found that by combining these two

Table 2 Comparison between the techniques

Sl.no	Scope	Content	Technique used	Performance	Outcome
1	Identification of the features of different methods [1]	Movie dataset	Content based filtering, cosine similarity	Mean average	High performance
2	Review based on sentiment analysis [2]	TMDb Movie reviews and movie datasets	Content based filtering	Accuracy	Analysing accurate movies
3	Understanding the concept of content-based filtering [4]	Movie datasets	Content based filtering	Accuracy and user reviews	Working of content based filtering
4	Understanding the basic approach of the system [5]	Ratings from the several users	Cosine similarity and content filtering	User rating	Efficiency
5	Analysis of sentiment analysis [3]	Reviews from users	Content based filtering	Algorithm for sentiment analysis	Accurate sentiments
6	Recommendation of movie a/c to users searching [6]	Movie datasets	Cosine similarity and content based filtering	Accuracy	Recommended movies

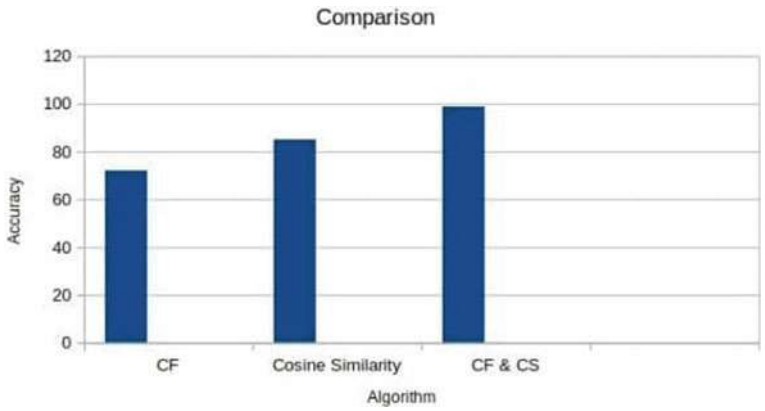


Fig. 3 Graph showing the comparison

techniques i.e., Content based filtering and Cosine Similarity, the best and accurate result can be found.

The graph in Fig. 4 depicts the movies that have been taken by using several datasets. Every dataset consists of approx 6000 movies till 2016. The graph shows the no of movies for every year. This graph is plotted using matplotlib which is a graphical plotting library used in python. It shows the number of movies from 1916 to 2016 [13].

Figure 5 is depicting the accuracy level which has been taken in percentage of the proposed system for no of movies taken together in a group. The user will search movies may be more than one time. This number of searched movies is being considered in a group of 5 or 10 movies for finding the similarity, The proposed system will compare the searched movies based on the criteria and also, its recommendation, it found the proposed methodology provides nearly 98% accurate result (in Fig. 5).

The proposed system experimented with various no of groups movies and compared with result. When a group made with 25 movies, it took 0.025 s for searching the desired movie from 6,000 movies (in Fig. 6).

Similarity Score

As there are several movies, how can it be decided which is the most similar movie in comparison to the movies that the user liked? So, the similarity score has been taken to find out the similar movie related to the movie which the user liked which helps in determining the two items that are similar to each other. It ranges from zero to one and is calculated by taking the cosine angle to find out the cosine similarity between two movie names which are in text format. For finding the optimised result, a similarity score has been calculated from cosine similarity.

Equation to find the accuracy:

$$\text{accuracy\_score}(\text{y\_test},\ \text{clf.predict}(\text{X\_test})) * 100 \tag{3}$$

```
In [6]: # we have movies only upto 2016
        import matplotlib.pyplot as plt
        data.title_year.value_counts(dropna=False).sort_index().plot(kind='barh',figsize=(15,16))
        plt.show()
```

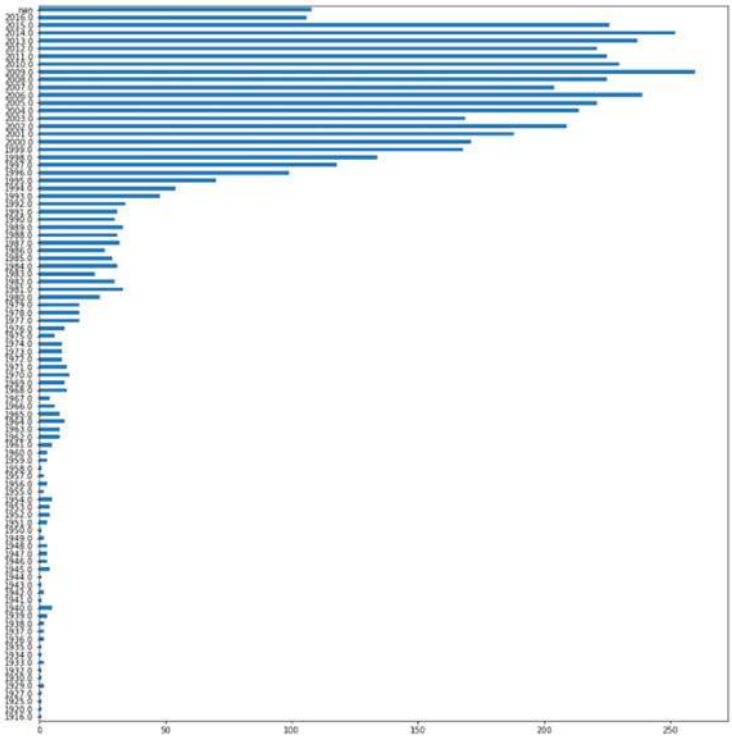


Fig. 4 Graph showing the movie

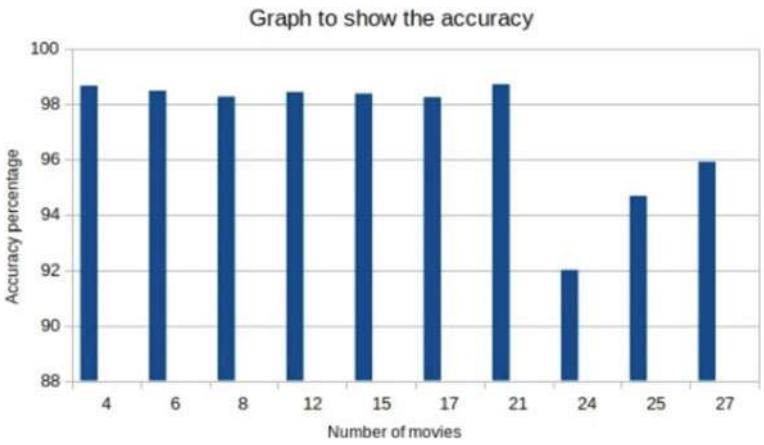


Fig. 5 Accuracy graph

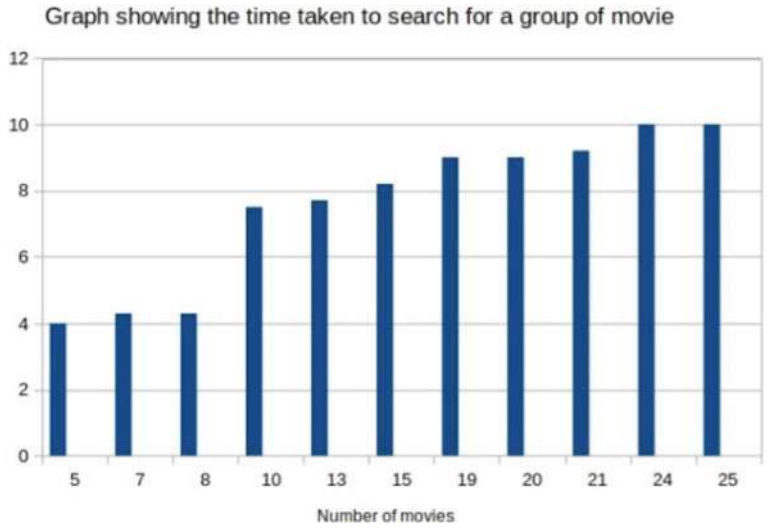
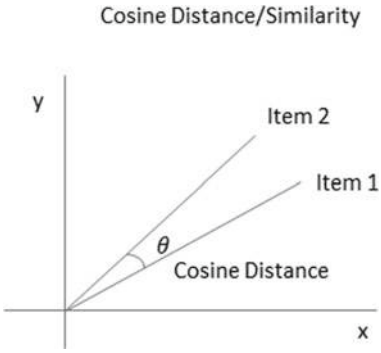


Fig. 6 Graph showing the time taken to Search for a group of movies

Fig. 7 Cosine similarity [14]



From the above equation the accuracy found is 98.77%.

The angle measures of two movies. It is ranging from Zero to One. As the user searches for a movie which is in text format. After searching, the text format changes into a vector format on the basis of the frequency of each word that occurs in the entire text. The cosine similarity now comes to the picture which checks the angle between the two movies. So, the concept of Cosine Similarity on the movie datasets achieves the similarity score required for obtaining the optimised search of the user [14] as shown in Fig. 7.

5 Conclusion

In this paper, various machine learning algorithms have been used to recommend movies based on the user's preferences. Also, it shows the details of the actor /actress who performed in those respective movies. Further we have done web scraping for the extraction of data which creates easy access to the scraped data. This system showed how to build a scalable based content filtering recommender system on jupyter notebook using HTML, CSS, JavaScript, Ajax for frontend. This system provides the model with an understanding of content-based filtering methods and the challenges for implementing them at scale. It also provides the implementations of both content-based filtering and cosine similarity algorithms on it. When the user's rating would not be enough or the movie's rating might be high or low, so in that case Cosine similarity would not work well. Hence for the improvement of this project, some other methods can be applied such as Adjusted Cosine Similarity which will help to overcome the drawbacks of cosine similarity.

References

1. Agarwal S. Goldberg K, Roeder T, Gupta D, Perkins C (2001) Eigenstate: a constant time collaborative filtering algorithm. Inform Retrieval 4(2):133–151. (Pub itemid 33636930)
2. Halder S, Samiullah M, Jehad Sarkar AM, Lee Y-K (2012) Movieswarm: information mining technique for movie recommendation system. In: Submission
3. Linden G, Smith B, York J (2003) Amazon. com recommendations: item-to-item collaborative filtering. Internet Computing IEEE 7(1):76–80
4. Lund J, Miller BN, Albert I, Lam SK, Konstan JA, Riedl J (2003) Movielens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th international conference on intelligent user interfaces. ACM, pp 263–266
5. Gupta M. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the fourteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., pp 43–52
6. Purnomo JE, Lekakos G, Caravelas P (2008) A hybrid approach for movie recommendation. Multimedia Tools Appl 36(1):55–70. (Pub itemid 350274737)
7. Singla R, Sarwar BM, Karypis G, Konstan J, Reidl J (2002) Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In: Proceedings of the 5th international conference on computer and information technology (ICCIT)
8. Gandhi SR, Pazzani M, Billsus D (1997) Learning and revising user profiles: the identification of interesting websites. Machine Learn 27(3):313–331. (Pubitemid 127510033)
9. Peng X, Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on computer supported cooperative work. ACM, pp 175–186
10. Movie-Recommendation-System-with-Sentiment-Analysis. https://github.com/kishan0725/AJAX-Movie-Recommendation-System-with-Sentiment-Analysis
11. AJAX-Movie-Recommendation-System-with-Sentiment-Analysis. https://github.com/kishan0725/AJAX-Movie-Recommendation-System-with-Sentiment-Analysis
12. Neo4jdocs. https://neo4j.com/docs/graph-data-science/current/alpha-algorithms/cosine/
13. TMDb(The Movie Database). https://www.themoviedb.org/

14. AJAX-Movie-Recommendation-System-with-Sentiment-Analysis. https://github.com/kishan0725/AJAX-Movie-Recommendation-System-with-Sentiment-Analysis/blob/master/.ipynb_checkpoints/preprocessing%201.ipynb
15. AJAX-Movie-Recommendation-System-with-Sentiment-Analysis. https://github.com/kishan0725/AJAX-Movie-Recommendation-System-with-Sntiment-Analysis

Chapter 12
Improved Bio-hashing Fingerprint Security Using Modified Arnold's Cat Map

Md Imteyaz Mohsin, Jyoti Bharti, and R. K. Pateriya

1 Introduction

Bio-hashing: a two-factor authentication template protection technique which uses invertible transformation function dependent on a user-specific key or secret to transform the original biometric features.

Figure 1 shows the Bio-hashing mechanism in general in which the acquired fingerprint undergoes transformation using a template transformation scheme to be represented in the form of a feature vector. A fixed-length bit-string is formed by taking a product between tokenized pseudo-random number generated using the secret or user-specific token as seed and the feature vector (fixed in length) obtained from acquired biometric data and then deciding upon each bit on the value of the result obtained by thresholding.

$f \in \mathbb{R}\ m$ having length m is a feature vector of fixed-length which is used to obtain a bit-string Bio-hash $\in \{0, 1\}n$, having size n such that bit-string ($n < m$), via a pseudo-random number matrix, $r \in \mathbb{R}\ n$. The elements of the matrix are between -1 and 1 which are distributed uniformally.

Most of the literature regard Bio-hashing as an invertible transform.

Bio-hashing is a popular technique because:

(i) there is a theoretical possibility of achieving EER ~0%; (ii) they are suitable for match-on-card application, (iii) authentication time is low due to linear operations and the fact that comparisons are performed in the binary domain, (iv) high recognition rates under ideal conditions.

However, the biggest drawback of Bio-hashing is that it is not robust to zero effort attacks, where an impostor gets the secret key and uses his/her own biometric data to secure an unauthorized access. The degraded performance in discriminating

M. Imteyaz Mohsin (✉) · J. Bharti · R. K. Pateriya
Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal, India
e-mail: imteyazmohsin@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_13

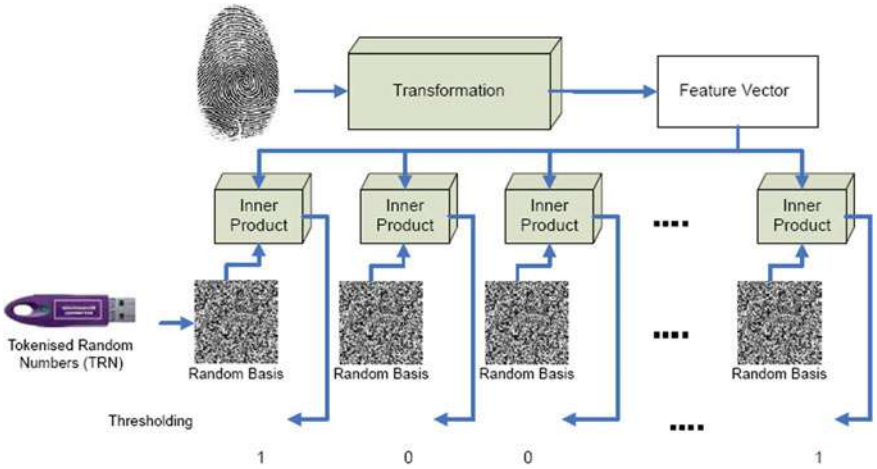


Fig. 1 Bio-hashing mechanism [1]

the actual population from impostor population can be improved by an increase in Bio-hash length; however, in real-life scenario the Bio-hash length could only be increased to an extent and is limited due to space requirement since projection matrix needs a lot of memory while computation as demonstrated by Marcel et al. [2]. (ii) Most literature term Bio-hashing as invertible transform because the scheme is invertible if both Bio-hash code and user-specific random matrix are known and if ratio of number of rows and number of columns of random matrix is close to 1. This is due to the fact that Bio-hashing is effectively a quantized under-determined linear equation system that can be partially solved using the pseudo-inverse method [3].

2 Literature Survey

It is possible for an attacker who has gained access to the token or secret key and the corresponding Bio-hash vector of a genuine user to obtain the original feature vector by inverting the Bio-hash vector. Attacker can thus perform direct attacks on the system using this feature vector.

Low Entropy of the Biometric system and the compromised key are the two main factors to be investigated while implementing Bio-hashing. A scheme that attempts to increase the Entropy of the fingerprint recognition system as well as secure the system against stolen token key motivated us to improve the Bio-hashing technique.

It is not much clear as to how to convert the minutiae to feature vector with ordered components and fixed-length to increase the Entropy of the Biometric system while maintaining the recognition performance. So, the literature survey also considers some work done in this area of representational schemes apart from studying some Bio-hashing template protection schemes already proposed in literature.

Teoh et al. (2004, 2006) proposed a Biometric salting approach named Bio-hashing which is based on user-specific random projection. However, performance of Bio-hashing scheme degrades significantly in case the impostor or attacker has the access to the secret token and utilizes it for the verification purpose using his own biometrics [4].

Hakan et al. used Bio-hashing combined with fixed-length template obtained using the complex spectral minutiae representation utilizing information from minutiae location and direction (x, y, and θ). The author claims to have obtained an EER of 0% on FVC2002 databases in an ideal scenario. However, in case of stolen token scenario the minimum EER achieved was 14.77% on FVC DB1 database [5].

Jin et al. proposed a variant of Bio-hashing dubbed "Index-of-Max" (IoM) hashing on the basis of localized random projection and rank correlation. IoM hashing transforms a real-valued biometric feature vector with externally generated random parameters into discrete index (max ranked) hashed code. It is said that the accuracy performance of the GRP-based IoM hashing is well-preserved even under stolen token scenario at EER 4.10% [6].

Olufade et al. used Arnold transformation to scramble the extracted biometric feature to address the issue of compromised secret key before applying Bio-hashing algorithm. However, the issue of compromise of both the keys during Arnold transformation and Bio-hashing stage has not been addressed. And the descrambling of biometric features due to compromised key during Arnold transformation could subsequently lead to partial recovery of original template. The author claims to have achieved a maximum FAR and FRR at 0.2% and 0.8%, respectively [7].

Morgan et al. proposed a new transformation function which embeds biometric data to the orthogonalized pseudo-random number matrix generated using secret key or token using nonlinear operations which the author has demonstrated to be effective against some attacks like stolen token attack. However, the trade-off between increased computational complexity and performance (authentication time) and template invertibility needs more analysis. The performance is very good as per author's claim an EER equal to 0.4% at GAR 99% [8].

Teoh et al. proposed Bio-phasor which is an improvement of Bio-hashing in which tokenized pseudo-random numbers are mixed iteratively with biometric data [9]. It is more robust to various attacks like stolen Biometric data attack and linkage attacks unlike Bio-hashing even when the BioCode size is low, as shown by Christophe Rosenberger [10]. However, this comes at a cost of increased authentication time due to high complexity to perform the required computation.

Teoh et al. in [11] integrated external data which may be derived from a secret key or physical token with biometric data in terms of a random multi-space quantization (RMQ) process which can be extended to fingerprints.

Now, we present some fixed-length feature vector fingerprint representations proposed in literature.

Jain et al. proposed FingerCode as shown in Fig. 2 which is a novel fixed-length fingerprint representation scheme that captures both global and local features. A reference frame or region of interest is first located in the acquired image, and the frame thus located is divided into sectors. Using the normalization process, each of

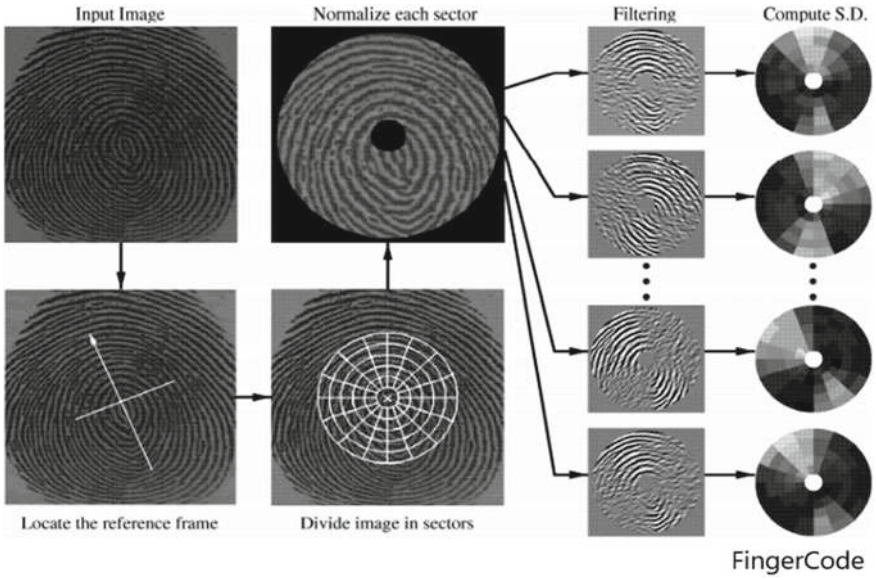


Fig. 2 FingerCode extraction flow diagram

the divided sectors is normalized. The ultimate aim is to decompose the information present in each of the discriminating sectors into separate components. The input imageGabor filters are used in this technique to compute the feature vector from the available texture features by filtering the image in eight different directions. The last step is to calculate the standard deviation of the gray values in the filtered images around the reference point. The author mentions that the limitation of the work is the assumption that the reference frame to tessellate the RoI of fingerprint image can be determined with a reasonable accuracy, and it does not consider part of the fingerprint that remains occluded or obliterated [12].

In [13], Yang et al. suggested a method for converting discrete-coordinate minutiae into a gray-level raster image of fixed size using pixel combinations of linear nature compared to a 2-D texture image that is generated at random. The feature vector fixed in size can be simply retrieved from a fixed-size raster image.

Haiyun et al. represented minutiae set as a fixed-length feature vector in the novel Spectral minutiae representation which is invariant to translation using polar-logarithmic transform.

The author acknowledges that the Spectral minutiae-based fingerprint recognition system underperforms in comparison to commercial fingerprint matchers like VeriFinger with an EER of 3.86% and GAR of 89.7% at FAR = 0% [14].

Capelli et al. proposed "Minutia Cylinder Code (MCC)" a novel minutiae-only feature representation as well as matching scheme for fingerprints. The 3D data structures called cylinders are built from minutiae distances and angles. This 3D cell-based structure, Cylinder, a 3D structure consists of multiple layers, and each

layer represents the density of neighboring minutiae in the corresponding direction. The author has demonstrated the recognition capability of this implementation to be very effective for light architectures. The fingerprint technique version MCC16 using the Local Similarity Assignment with Relaxation (LSA-R) technique has an EER = 0.15% and FMR1000 = 0.18% [15]. Although this scheme has variable-sized representational scheme, it can be transformed into fixed-length representation as done by Luo et al.

Luo et al. in [16] used "Global Minutia cylinder code" which is an extension of MCC with a reference type that needs to be stably detected in fingerprints. The author claims to have achieved an EER: 0.026%.

Teoh et al. in [17] used Fingerprint minutia to produce Binary strings utilizing kernel learning algorithms. This scheme achieves an EER: 4.56%.

3 Proposed Architecture

The proposed framework as shown in Fig. 3 follows the below main stages as:

(a) Fingerprint feature extraction (b) statistical parameter calculation from acquired biometric data (c) fixed-length vector representation of extracted features (d) perform Bio-hashing and introduce randomness using modified Arnold's cat map technique to the orthogonalized pseudo-random vectors (e) perform the final step of Bio-hashing, i.e., thresholding to generate the Bio-hash codes or BioCode.

The acquired fingerprint undergoes the preprocessing and feature extraction using the FingerCode mechanism as described earlier. Here our method uses modified Arnold's cat map technique to introduce randomness in the orthogonalized TRN during Bio-hashing phase. For this purpose, we calculate the statistical parameter from the enrolled biometric features in the form of mean (μ) and use this to form a seed/key for randomizing the orthogonalized pseudo-random number matrix. This key or seed needs not to be stored in the system or database which makes the technique more secure. In past, nonlinear transformations have been used to salt the biometric data, which are not efficient in real time. Our system overcomes these limitations and performs effectively in real-time.

3.1 Bio-hashing (Improved)

The algorithm below outlines the changes introduced in the standard Bio-hashing technique to design the proposed architecture.

Algorithm 1 Reformulated Bio-hashing.

Inputs

1: $b = (b_1, b_2, b_3 \ldots b_p)$: biometric template

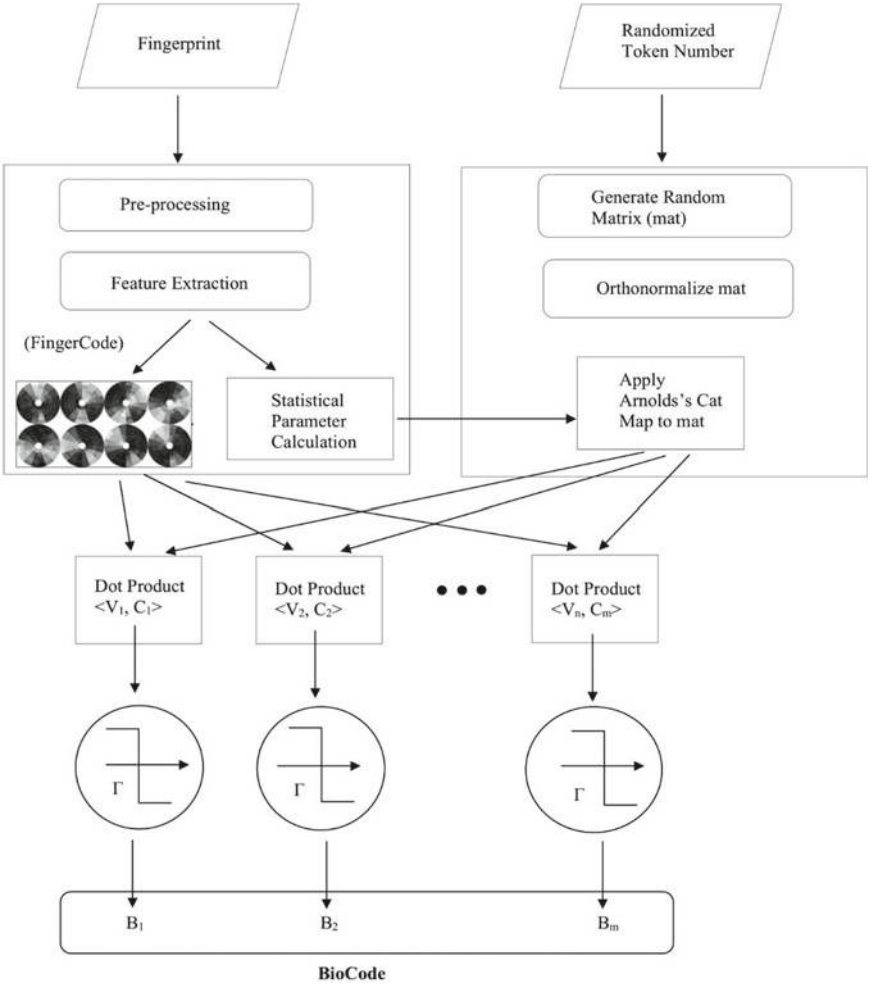


Fig. 3 Architecture of proposed scheme

2: K: seed value or key

Output: $B = (B_1, B_2, B_3 \ldots B_m)$: BioCode of length m bits.

3: Generation with K, n TRN vectors $V_1, V_2, V_3 \ldots V_n$ having length p,
4: Orthogonalization of TRN vectors using Gram–Schmidt algorithm,
5: Apply MAT to the orthogonalized TRN matrix using Arnold keys $\overline{\mu_{\text{bif}}}$ and $\overline{\mu_{\text{ter}}}$ (described in next section)
6: for $j = 1, 2, 3 \ldots n$, compute $x_j = < b, V_j >$.
7: end for
8: Bio-hash/BioCode is computed as below:

$$B_j = \begin{cases} 0 \text{ for all } x_j < \tau \\ 1 \text{ for all } x_j \geq \tau, \end{cases}$$

where τ is the threshold parameter calculated or normally taken as mean.

3.2 *Biometric Statistical Parameter*

Basak et al. have described a new reference point called Virtual Core Point [18] which is used to calculate the statistical parameter mean (μ) of the fingerprint image. The mean (μ) is calculated and then normalized using the below algorithm2. In order to deal with intra-user variation in the minutiae points, we have tweaked the mean (μ) to normalize it to the slight variations in the mean (μ) derived from the fingerprints of the same person.

One important point to note is that the statistical mean from the fingerprint images of the same person does not show much variation, and based upon testing data, the maximum deviation is less than 9%. Figure 4 shows the distribution of mean for fingerprints from the same person which was tested using our method for which the standard deviation comes around 1.96.

Now, these normalized means calculated for two sets (a) bifurcation minutiae points (b) termination minutiae points can be used as keys during MAT.

Algorithm 2: Calculate and normalize mean(μ).

1. Euclidean distances $l_i = \sqrt{(M_{cx} - x)^2 + (M_{cy} - y)^2}$
2. Mean $(\mu_{\text{bif}}) = \sum_{i=1}^{N} \frac{l_i}{N}$
3. Normalized mean$_1$ $\left(\overline{\mu_{bif}}\right) = \text{base}_1^* \left\lfloor \frac{\mu_{\text{bif}}}{\text{base}_1} \right\rfloor + \text{base}_1$

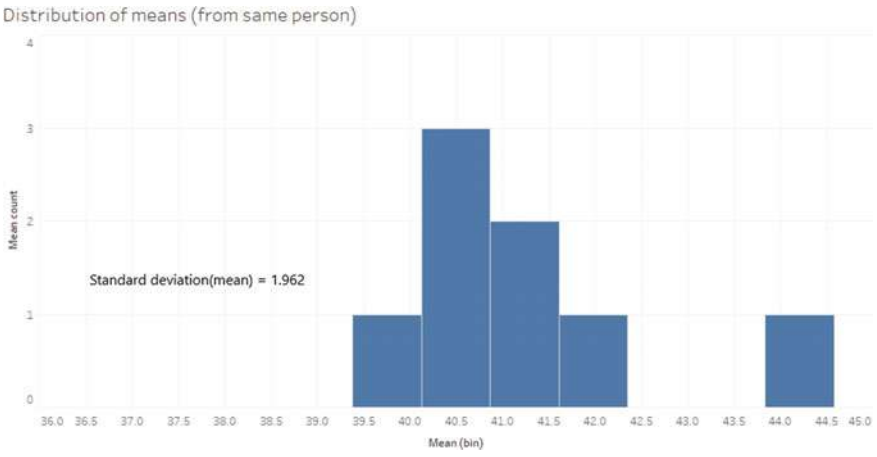


Fig. 4 Distribution of mean for a sample person showing standard deviation

4. Mean (μ_{ter}) $= \sum_{i=1}^{M} \frac{l_i}{M}$
5. Normalized mean$_2$ ($\overline{\mu_{\text{ter}}}$) $= \text{base}_2^* \left\lfloor \frac{\mu_{\text{ter}}}{\text{base}_2} \right\rfloor + \text{base}_2$

where N and M are the total bifurcation and termination minutiae points.

We calculate two different means as below:

Euclidean distances of all the bifurcation minutiae points (x, y) in the fingerprint images from the virtual core point (M_{cx}, M_{cy}) are calculated, and the statistical parameter mean μ_{bif} is calculated by summing up all the Euclidean distances and dividing it by the total count of minutiae points.

We find the nearest multiple of base (this parameter is calculated based upon the maximum variation in mean by testing different fingerprint images from the same person) greater than the mean to form the normalized mean ($\overline{\mu_{\text{bif}}}$).

Similarly, the mean μ_{ter} and normalized mean $\overline{\mu_{\text{ter}}}$ is calculated using the termination minutiae points.

Normalizing the mean is necessary because the normalized mean is used as the modified Arnold's key in the next step which we perform during the Bio-hashing step to introduce randomness in the final orthogonalized pseudo-random vector matrix.

We demonstrate the example for four fingerprint images:

base$_1$ = 23:

The deviation in the mean for the fingerprint images from the same person as shown in Table 1 is less due to which the calculated normalized mean turns out to be same. Here, the base parameter is taken as 23.

The deviation in the mean for the fingerprint images from different person as shown in Table 2 is high due to which the calculated normalized mean vary from each other. Here, the base parameter is taken as 23.

For base$_2$ = 17:

The deviation in the mean for the fingerprint images from the same person as shown in Table 3 is less due to which the calculated normalized mean turns out to be

Table 1 Mean and normalized mean for fingerprint images from same person (base parameter = 23)

Mean (μ_{bif})	Normalized mean ($\overline{\mu_{\text{bif}}}$)
40.8065	46
41.5126	46
39.6532	46
43.2518	46

Table 2 Mean and normalized mean for fingerprint images from different persons (base parameter = 23)

Mean (μ_{bif})	Normalized mean ($\overline{\mu_{\text{bif}}}$)
51.7436	69
33.1376	46
86.7532	92
72.2054	92

Table 3 Mean and normalized mean for fingerprint images from same person (base parameter = 17)

Mean (μ_{ter})	Normalized mean ($\overline{\mu_{\text{ter}}}$)
53.5877	68
57.0112	68
55.6209	68
59.6442	68

Table 4 Mean and normalized mean for fingerprint images from different persons (base parameter = 17)

Mean (μ_{ter})	Normalized mean ($\overline{\mu_{\text{ter}}}$)
31.0046	51
53.1376	68
62.7532	68
37.2054	51

same. Here, the base parameter is taken as 17 which is smaller compared to earlier base parameter.

The deviation in the mean for the fingerprint images from different person as shown in Table 4 is high due to which the calculated normalized mean varies from each other. However, the normalized mean for some cases match with each other due to lower value of base parameter, which is taken as 17 in this case. Use of machine leaning algorithms to fine-tune the base parameters can be taken up in future research.

3.3 Modified Arnold's Cat Map

Vladimir Arnold proposed Arnold's cat map in 1960 which is mostly applied to digital images to randomize the original organization of pixels to make it imperceptible or noisy. One remarkable thing about this transformation is that it has a periodicity of p and when applied to an image equal to its periodicity, i.e., p then original image re-appears.

Arnold's Cat Map equation is given as below:

$$\begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \tag{1}$$

$$x_{\text{new}} = ((2^{*}x) + y) \bmod N$$

$$y_{\text{new}} = \left(\left(1^{*}y\right) + \left(1^{*}x\right)\right) \bmod N$$

We will demonstrate an example using a 2D array of shape (3, 4) by applying Eq. (1) to this array as below:

Let arr be an array of shape (3, 4) given by:

$$\text{arr} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

Applying the above transformation (Eq. (1)) to this 2D array 1, 2, 3, 4 …8 number of times it can be inferred that the periodicity is 8 as shown below:

$$\text{arr}_1 = \begin{bmatrix} 3 & 6 & 2 & 4 \\ 10 & 5 & 7 & 9 \\ 8 & 12 & 1 & 11 \end{bmatrix} \quad \text{arr}_5 = \begin{bmatrix} 10 & 11 & 12 & 4 \\ 3 & 1 & 7 & 9 \\ 8 & 2 & 5 & 6 \end{bmatrix}$$

$$\text{arr}_2 = \begin{bmatrix} 2 & 5 & 6 & 4 \\ 12 & 10 & 7 & 8 \\ 9 & 11 & 3 & 1 \end{bmatrix} \quad \text{arr}_6 = \begin{bmatrix} 12 & 1 & 11 & 4 \\ 2 & 3 & 7 & 8 \\ 9 & 6 & 10 & 5 \end{bmatrix}$$

$$\text{arr}_3 = \begin{bmatrix} 6 & 10 & 5 & 4 \\ 11 & 12 & 7 & 9 \\ 8 & 1 & 2 & 3 \end{bmatrix} \quad \text{arr}_7 = \begin{bmatrix} 11 & 3 & 1 & 4 \\ 6 & 2 & 7 & 9 \\ 8 & 5 & 12 & 10 \end{bmatrix}$$

$$\text{arr}_4 = \begin{bmatrix} 5 & 12 & 10 & 4 \\ 1 & 11 & 7 & 8 \\ 9 & 3 & 6 & 2 \end{bmatrix} \quad \text{arr}_8 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

arr_1, arr_2, arr_3 … arr_8 are the arrays obtained after applying the transformation to the original array 1, 2, 3 …8 number of times, respectively.

We can modify the original Arnold's cat map equation as below to form modified Arnold's cat map equation:

$$\begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \end{bmatrix} = \begin{bmatrix} \text{key} & \text{key}+1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \tag{2}$$

$$x_{\text{new}} = \left(\left(\text{key}^* x\right) + (\text{key}+1)^* y\right) \bmod N$$

$$y_{\text{new}} = \left(\left(1^* y\right) + \left(1^* x\right)\right) \bmod N$$

Applying the above transformation (Eq. (2)) to this 2D array 1, 2, 3, 4 …12 number of times for key = 3, it can be inferred that the periodicity is 12 as shown below:

$$\text{arr}_1 = \begin{bmatrix} 8 & 2 & 9 & 4 \\ 6 & 7 & 12 & 10 \\ 3 & 1 & 11 & 5 \end{bmatrix} \quad \text{arr}_5 = \begin{bmatrix} 10 & 2 & 9 & 4 \\ 6 & 7 & 12 & 1 \\ 3 & 8 & 11 & 5 \end{bmatrix} \quad \text{arr}_9 = \begin{bmatrix} 1 & 2 & 9 & 4 \\ 6 & 7 & 12 & 8 \\ 3 & 10 & 11 & 5 \end{bmatrix}$$

$$\text{arr}_2 = \begin{bmatrix} 10 & 2 & 3 & 4 \\ 7 & 12 & 5 & 1 \\ 9 & 8 & 11 & 6 \end{bmatrix} \quad \text{arr}_6 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 7 & 12 & 5 & 8 \\ 9 & 10 & 11 & 6 \end{bmatrix} \quad \text{arr}_{10} = \begin{bmatrix} 8 & 2 & 3 & 4 \\ 7 & 12 & 5 & 10 \\ 9 & 1 & 11 & 6 \end{bmatrix}$$

$$\text{arr}_3 = \begin{bmatrix} 1 & 2 & 9 & 4 \\ 12 & 5 & 6 & 8 \\ 3 & 10 & 11 & 7 \end{bmatrix} \quad \text{arr}_7 = \begin{bmatrix} 8 & 2 & 9 & 4 \\ 12 & 5 & 6 & 10 \\ 3 & 1 & 11 & 7 \end{bmatrix} \quad \text{arr}_{11} = \begin{bmatrix} 10 & 2 & 9 & 4 \\ 12 & 5 & 6 & 1 \\ 3 & 8 & 11 & 7 \end{bmatrix}$$

$$\text{arr}_4 = \begin{bmatrix} 8 & 2 & 3 & 4 \\ 5 & 6 & 7 & 10 \\ 9 & 1 & 11 & 12 \end{bmatrix} \quad \text{arr}_8 = \begin{bmatrix} 10 & 2 & 3 & 4 \\ 5 & 6 & 7 & 1 \\ 9 & 8 & 11 & 12 \end{bmatrix} \quad \text{arr}_{12} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

arr_1, arr_2, arr_3 … arr_{12} are the arrays obtained after applying the transformation to the original array 1, 2, 3 …12 number of times, respectively.

Key points:

- Arnold periodicities vary for the different values of key for the same array.
- All these transforms differ from each other in their mapping.
- Elements scrambled using a particular Arnold transform cannot be restored using a different Arnold transform [19].

Algorithm 3: Modified Arnold's Cat Map

Input: Orthonormal pseudo-random matrix *arr* of shape (r, c)

Output: Randomized Orthonormal pseudo-random matrix *arr* of shape (r, c)

1. No. of times MAT is applied = $\overline{\mu_{bif}}$
2. count = 0
3. do, until $\overline{\mu_{bif}}$ is greater than the count
4. **for** row = 1 to r **do**
5. **for** col = 1 to c **do**
6. $\text{row}_{\text{new}} \longleftarrow (\overline{\mu_{ter}} * row + (\overline{\mu_{ter}} + 1) * col) \,\%\, row$
7. $\text{col}_{\text{new}} \longleftarrow (1 * row + 1 * col) \,\%\, col$
8. temp $\longleftarrow$ $arr\,[row][\,col]$
9. $arr\,[row][\,col] \longleftarrow arr\,[\text{row}_{\text{new}}][\,\text{col}_{\text{new}}]$
10. $arr\,[\text{row}_{\text{new}}][\,\text{col}_{\text{new}}] \longleftarrow$ temp
11. **end for**
12. **end for**
13. count $\longleftarrow$ count + 1
14. **end**

Table 5 Comparisons under different parameters for ideal and token stolen scenario

Dataset	Parameters	(Token not stolen scenario)		
		EER (%)	FAR (%)	FRR (%)
FVC2004 DB2B	$base_1 = 29$, $base_2 = 31$	0	0	0
	$base_1 = 23$, $base_2 = 19$	0	~0	~0
FVC2004 DB2B	$base_1 = 23$, $base_2 = 19$	2.7	3.8	0.8
	$base_1 = 19$, $base_2 = 17$	1.8	1.7	4.1

3.4 Results

The datasets DB1, DB2, DB3, and DB4 contain each 80 fingerprint images from 10 persons. Generally, the count of minutiae varies from 60 to 80 for a fingerprint of good quality [20], but the count could vary from one fingerprint to another and the deliberate perturbations in the fingerprint images make it difficult to make the recognition highly accurate. The proposed technique was tested against dataset DB2 only. Table 5 shows the results under different parameter selection for the ideal case and token stolen scenario. Under ideal scenario, the proposed technique achieves an EER of 0%. Also, under the token stolen scenario the proposed technique performs better until the bad quality and distorted fingerprint images are encountered, those certain fingerprint images act like outlier and significantly increase the FRR to 4.1% at higher values of setting parameters. In cases when the statistical parameters derived from both genuine and impostor fingerprint images are close to each other, the FAR increases to 3.8% which is not good for a real-world system.

For lower values of setting parameters, the FRR increases significantly due to outliers.

Sample Results

This section presents the results of experiments obtained using the suggested method. Accuracy of the algorithm is checked by performing testing on fingerprint images that had been enrolled and their BioCodes saved into the database against the probed fingerprints. Few sample results of the proposed work are shown below:

Matching Result for fingerprints from same person:

[0100001101101100111111110100 0
1011100111001111001101010011011
1011000000011011100001011000000
0010111011011000100011001010000
0000011100011110010101001000001
1111000001000000001000000010100
1011111001101001011101010000001
1001111111111010111010101011001
0001000000110100]

Enrolled BioCode1

[0100000100101100011111111100 01
01110011100111100110001111 1011
10110000001101010000101100001
00011111111101100011001110100000
00000111100101100101010010 0001
11110000110000000010000100110100
11111010011010010111010100000 0
10011101101101011101010101100 1
10010000000100100]

Probed BioCode2.
Hamming distance between two BioCodes = d (BioCode1, BioCode2).
= 26.
Result: **Match.**
Matching Result for fingerprints from different person (token not stolen):

[11100000101111110011110000 1110
10101111101011110011100 0100011
11010011000010010101000011111011
11100101100000001001000111111011
00010100001100001111100011 0010
01010110000110011111000000 01000
10011001110011111011010 0101010
00010011101001001001001010 0011
01111110100 01111]

Probed BioCode3 from impostor.
Hamming distance between two BioCodes = d (BioCode1, BioCode3).
= 134.
Result: **Not Match.**
Matching Result for fingerprints from different person (token stolen scenario):

[01011001000000100 1101011111010
11110110100 0101110010101101101
01011101000101100 0111110 001001
01100101011000 0101100000000010
01101001101101010000 0010 01110

0101001011001000000000011011110
0001010111001110111111011110101
0000011011011110111001010010
1101101111011101]

Probed BioCode4 from impostor.
Hamming distance between two BioCodes = d (BioCode1, BioCode3).
= 133.
Result: **Not Match.**

4 Discussion and Conclusion

In this research, improved Bio-hashing technique using modified Arnold's cat map has been used to recognize or authenticate the fingerprints (with different perturbations) which can help overcome the vulnerabilities of the standard Bio-hashing technique. The technique has been evaluated after analyzing the results for different performance metrics EER, FAR, and FRR. It is found that proposed technique showed the best results when the fingerprint's quality is good to moderate and when the quality is poor the FRR and FAR increases significantly due to loss of minutiae points. An EER of 2.7% is achieved at FRR = 0.8% which is not acceptable under real-life scenarios. Our system has limitations while representing the minutiae into fixed-length feature vector, recently some work has been done and several fixed-length feature vector representational schemes have been proposed.

Future Work considering various minutiae-based representational schemes to design an entropy-maximized recognition system can be considered for this study. Also, investigating other statistical properties derived from minutiae or other invariant fingerprint features would help in refining the parameter settings to achieve better recognition performance.

References

1. Nachiappan R, Karthikeyan S (2011) An evaluation of fingerprint security using noninvertible biohash. Int J Netw Secur Its Appl 3. https://doi.org/10.5121/ijnsa.2011.3411
2. Krivokuća V, Marcel S (2020) On the recognition performance of biohash-protected finger vein templates. In: Uhl A, Busch C, Marcel S, Veldhuis R (eds) Handbook of vascular biometrics. advances in computer vision and pattern recognition. Springer, Cham
3. Jin ATB, Hui LM (2010) Cancelable biometrics. Scholarpedia 5(1):9201
4. Jin ATB, Ling DNC, Goh A (2004) Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recogn 37(11):2245–2255. ISSN 0031–3203. https://doi.org/10.1016/j.patcog.2004.04.011

5. Topcu B, Erdogan H, Karabat C, Yanikoglu B (2013) Biohashing with fingerprint spectral minutiae. In: 2013 ınternational conference of the BIOSIG special ınterest group (BIOSIG), pp 1–12
6. Jin Z, Hwang JY, Lai Y, Kim S, Teoh ABJ (2018) Ranking-based locality sensitive hashing-enabled cancelable biometrics: index-of-max hashing. IEEE Trans Inf Forensics Secur 13(2):393–407. https://doi.org/10.1109/TIFS.2017.2753172
7. Onifade O, Olayemi K, Isinkaye F (2020) A fingerprint template protection scheme using arnold transform and bio-hashing. Int J Image Graph Signal Process 12:28–36. https://doi.org/10.5815/ijigsp.2020.05.03
8. Ghammam L, Barbier M, Rosenberger C (2018) Enhancing the security of transformation based biometric template protection schemes. CyberWorlds, Oct 2018, Singapour, Singapore. ffhal-01862157f
9. Teoh ABJ, Ngo DCL (2006) Cancellable biometrics realization through biophasoring. In: Proceedings of 9th IEEE international conference on control, automation, robotics and vision (ICARCV'06)
10. Christophe R (2018) Evaluation of biometric template protection schemes based on a transformation. pp 216–224. https://doi.org/10.5220/0006550502160224
11. Goh TA, Ngo D (2006) Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. IEEE Trans Pattern Anal Machine Intell 28(12):1892–1901
12. Anil J, Salil P, Lin H, Pankanti S (1999) FingerCode: a filterbank for fingerprint representation and matching. In: Proceedings/CVPR, IEEE computer society conference on computer vision and pattern recognition. IEEE Computer society conference on computer vision and pattern recognition. vol 2. pp 193. https://doi.org/10.1109/CVPR.1999.784628
13. Yang B, Chen Z, Busch C (2011) Raster image representation of fingerprint minutiae. In: Proceedings of the 2011 ACM symposium on applied computing (SAC '11). Association for Computing Machinery, New York, NY, USA, pp 8–12. https://doi.org/10.1145/1982185.1982188
14. Xu H, Veldhuis RNJ, Bazen AM, Kevenaar TAM, Akkermans TAHM, Gokberk B (2009) fingerprint verification using spectral minutiae representations. IEEE Trans Inf Forensics Secur 4(3):397–409. https://doi.org/10.1109/TIFS.2009.2021692
15. Cappelli R, Ferrara M, Maltoni D (2010) minutia cylinder-code: a new representation and matching technique for fingerprint recognition. IEEE Trans Pattern Anal Mach Intell 32:2128–2141. https://doi.org/10.1109/TPAMI.2010.52
16. Luo Y, Feng J, Zhou J (2014) Fingerprint matching based on 16 global minutia cylinder code. In: IEEE ınternational joint conference on biometrics
17. Jin Z, Lim MH, Teoh ABJ, Goi BM, Tay YH (2016) Generating fixed-length representation from minutiae using kernel methods for fingerprint authentication. IEEE Trans Syst Man Cybernet: Syst 46(10):1415–1428
18. Basak S, Islam M, Amin M (2012) A new approach to fingerprint detection using a combination of minutiae points and invariant moments parameters. J Info Process Syst 8:421–436. https://doi.org/10.3745/JIPS.2012.8.3.421
19. Minati M, Ashanta R, Sunit K (2014) High security ımage steganography with modified arnold's cat map. Int J Comput Appl 37.https://doi.org/10.5120/4636-6685
20. Jain AK, Nandakumar K, Nagar A (2008) Biometric template security. EURASIP J Adv Signal Process 113:17. https://doi.org/10.1155/2008/579416

Chapter 13
A Framework for Smart Traffic Controller by Improved Cooperative Multi-agent Learning Algorithms (ICMALA)

Deepak A. Vidhate and Parag Kulkarni

1 Introduction

Congestion is prevalent these days, and it is worsening because of rapidly rising number of vehicles on the road. Typically, they occur in urban areas, where traffic lights make up the maximum of the management strategy. A high number of automobiles are spread throughout a broad and densely populated urban area. This makes it difficult and complicated to successfully manage such a large-scale, dynamic, and complex system [1]. Present road conditions are frequently forced to its limits, and future extension is not viable due to environmental and economical constraint. As a result, one of the very essential goals is to enhance the outcome of traffic signal management in order to achieve effective traffic management. Despite the reality that volume of road traffic on the road is increasing in large cities, most current traffic management technologies do not take use of intelligent traffic light regulation [2]. The most of traffic light controllers, in fact, rely on simple traffic procedures that simply switch red and green signals for a set period of time. Time span may change for the period of the busy duration, but it is not improved in any other way. Many scientists have discovered how to use machine learning to investigate more effective traffic light controllers automatically. In most metropolitan places, it has been found that intelligent traffic regulation and improving the deployment efficacy of roadways are an efficient and cost-effective methods of resolving the urban traffic crisis [3]. The traffic signal light controller technique gives a critical aspect of the intelligent transportation system [4]. There are a plethora of variables that influence traffic control scheme. Since quick and erratic traffic flow, static control is hardly a solution. The paper proposes a reinforcement learning-based dynamic traffic control framework

D. A. Vidhate (✉)
Dr. Vithalrao Vikhe Patil College of Engineering, Ahmednagar, Maharashtra, India
e-mail: dvidhate@yahoo.com

P. Kulkarni
iKnowlation Research Laboratory, Shivajinagar, Pune, Maharashtra, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_14

[5]. Reinforcement learning may be a crucial step in resolving the aforementioned issues. It is effective in resolving a variety of issues [6]. In reinforcement learning, an agent learns from previous experience through trial and error and communication with the surroundings. As a result, it is a good fit for the situation of traffic light control's dynamic decision-making. Every action provides direct reinforcement as well as new interpretation about the state of environment. The agent has a control strategy that makes best utilization of long-term reinforcements it is likely to receive. Consider the situation as a multi-agent strategy that each agent watches a single traffic signal using reinforcement learning [7, 8]. Because each agent just watches its immediate surroundings and chooses actions that depend on a single traffic light signal, the single agent's activities are not communicated. Hence, agents may pick their own activities that appear to be useful to the single but, when combined, cause a severe disruption. Different traffic signal control kinds are defined as action choose in the reinforcement learning framework, while the number of vehicles coming and the density of vehicles at an intersection are recorded as environment conditions. As received rewards, signal management metrics such as delay time, the number of halting cars, and overall vehicle density are given. The paper uses cooperative learning and explicit communication between agents to use reinforcement learning technology to traffic signal planning. An agent has an effect merely by having a direct impact on its network, implying that it is located nearby in the environment. Using this idea, the general cooperation problem can be observed as an integration of specific cooperation problems that can be solved using reinforcement learning approaches [9]. Because of system's time constraints, an efficient approach for achieving the best combined actions is required. Joint action learning [10] is used to establish the optimum possible joint actions between linked agents.

This paper adds to a number of previous findings. First, the current reinforcement learning framework for traffic control is improved by allowing sufficient cooperation between adjacent traffic signals. Second, it applies the basic method to collaborative action learning to a large-scale problem, demonstrating its effectiveness in real-world scenarios. Third, it gives a unique perspective on the features that a traffic light control system must possess in order for such collaboration to be beneficial, as well as evidence that joint action learning outperforms alternatives to traffic light management with similar characteristics.

The article is divided into different sections. Section 2 provides information on relevant work in the field of traffic signal control. The parameters for traffic evaluation are described in Sect. 3. Section 4 suggests A Framework for Smart Traffic Controller by Improved Cooperative Multi-agent Learning Algorithms (ICMALA). The system model for smart traffic signal controller is explained in Sect. 6 including definitions for the state, action, and reward functions. The experimental result investigations are explained in Sect. 7, followed by final comments.

2 Literature Survey

There are two primary types of traffic control systems: offline and online. Theoretical moves toward control optimization are used in offline approaches. Online approaches dynamically adjust the time of the traffic regulator based on current traffic circumstances. Many academics have addressed and concentrated on traffic control approach and joint traffic flow control direction. Fuzzy logic [11], neural networks [12], and evolutionary algorithms [13] are among the methodologies developed. These methods are only applicable to a smaller system with a very less number of controls. Various traffic control learning algorithms, like Sarsa [14] or Q-learning [15], have already been applied. Nonetheless, all techniques face the same issue of being unable to scale to a large environment due to the rapid expansion of state space. Such methods are either limited to a little context or used to prepare consistent traffic signal regulation. Model-oriented reinforcement learning [16] is a well-defined method in which state change and reinforcement are calculated that depend on information and then used to estimate a viable strategy. All states are separated into individual states using this system, which is dependent on each vehicle. The *Q*-value connects the position of one vehicle at one time stage to its position at the next time stage. During peak hours, a number of states increase sequentially, potentially creating a large environment. The approach from [17] is the transportation industry approach. The traffic congestion issue was partially overcome using F-B technique approaches [17]. Following that, various improved approaches based on the F-B approach emerged [18]. The success of the time allotment method presented in [19] was estimated using the driving reimbursement coefficient and delay time. The method appears to be crisp and sensible because it decreases the length of time spent waiting. This strategy could only just tackle the heavy traffic problem; hence, a new proper methodology is required. Vidhate and Kulkarni [20] used intelligent traffic control to alleviate traffic blocking, but congestion difficulties between surrounding intersections necessitated a more sophisticated approach. The local synchronization resulted in an excellent answer to the problem mentioned in [21]. As of unexpected and sophisticated type of the problem, constructing an exact arithmetic form of a traffic classification in advance is extremely difficult [22]. It has proven to be a popular method of resolving traffic issues by utilizing computer knowledge and machine learning [23]. Reinforcement learning is one of many machine intelligence technologies utilized to fine-tune the control of a transportation system [24]. The Q-learning algorithm [25] was used in a study to regulate online traffic. The technique was capable of selecting the best coordination model under various traffic scenarios. Some systems [26] that use the Q-learning algorithm have achieved remarkable results. A paper offered an online traffic control system based on the Q-learning algorithm, which made a better attempt at dealing with normal traffic congestion [27].

3 Traffic Estimation Parameters

The signal light control system is in charge of a critical function in traffic management. In normal circumstances, a sensible time allocation mechanism guarantees that traffic flows smoothly. The delay time, the number of automobiles halt at intersections, and the number of newly incoming automobiles are all commonly used traffic estimation parameters [28].

3.1 Delay Time

The defer/delay time is the contrast within the ongoing and hypothetically determined time for a vehicle to leave a sign. We can ascertain the time contrast by setting aside the complete postpone effort for a specific time frame and the normal defer season of a cross. The more extended the postpone time, the more slow the normal speed of a vehicle leaving a sign.

3.2 Number of Vehicles Stopped

The quantity of vehicles ended is dictated by the quantity of vehicles holding up behind the stop line to leave the street signal. The marker [29] is utilized to survey the street's perfection just as the progression of traffic. It is characterized as

$$\text{stop} = \text{stopG} + \text{stopR}$$

where stopR denotes the number of cars stopped before a red light and stopG denotes the number of cars halted before a green light.

3.3 Number of Vehicles Newly Arrived

Signal saturation is calculated as the relationship of real traffic movement to extreme available traffic movement. The value of the freshly arrived car is computed as follows:

$$S = \frac{\text{traffic flow}}{\left(d_r \times s_f\right)}$$

where s_f is traffic flow of the signal and d_r is the relation of red-light period to green-light period.

3.4 Traffic Movement Capability

Traffic movement capability indicates the largest volume of cars transient over the signal. The indication displays the outcome of the signal control method. The period of traffic signals and the ability of traffic volume are linked.

4 Cooperative Nash-Q Learning Framework

Signal agents recognize events and replies in traffic circumstances, learn data using learning algorithms, and then develop accomplishment choices based on the information they have gathered. The aim of a traffic indicator controller is to improve traffic flow and reduce average delay time. Signals at one crossroads synchronize with signals at another crossroads in this traffic setup. Signals at each crossroads create a cooperative approach to make the most of their own profit throughout the process. Agent cooperation is achieved by splitting partial details of states among neighboring agents. To obtain the finest promising traffic signal regulations, A Framework for Smart Traffic Controller by Improved Cooperative Multi-agent Learning Algorithms (ICMALA) is created for smart traffic controller approaches toward dynamic conditions of real-world traffic situations [29, 29]. Q learning to the multi-agent learning area is extended with Nash equilibrium concept. To get used to Q learning to the multi-agent framework, the initial move is to identify the necessity to view joint actions, instead of only independent actions. Q-function for multi-agent turns into $Q(s, a^1 \ldots a^n)$, instead of single agent Q-function [30].

Proposed Cooperative Nash-Q learning framework

According to the analysis above, the multi-agent Markov game reinforcement algorithm is summarized as follows by taking agent k for example.

1. Choose action a_t^k under joint actions of other.
2. Observe $r_t^1 \ldots r_t^k \ldots r_t^n$ and $s_{t+1} = s'$.
3. Agent k updates value about other agent's actions according to.

$$p\left(a^i | a^k, s_{t+1}\right) = \frac{p(s_{t+1} | a^i, a^k)\, p\left(a^i\right)}{p(s_{t+1} | a^k)}$$

4. Q table update Q_{t+1}^k as

$$Q_{t+1}^k(s_t, a_t^1 \ldots a_t^k \ldots a_t^n) = (1 - \alpha_t)\, Q_t^k(s_t, a_t^1 \ldots a_t^k \ldots a_t^n)$$

$$+\alpha_t\left[r_t^k + \alpha P\left(a^i \ldots a^n | a^k, s_{t+1}\right) Q_t^k(s_{t+1})\right]$$

Convergence

It is to demonstrate the meeting of Q_t^i to equilibrium Q_*^i for the learning agent i. Joint approaches of all agents determine the value of Q_*^i. It means agent needs to study Q-values of all the agents and develop policies after it. The aim of study is $(Q_*^1 \ldots Q_*^n)$, and it is to prove the convergence of $(Q_t^1 \ldots Q_t^n)$ to $(Q_*^1 \ldots Q_*^n)$ [31, 32].

Convergence Proof

Basic assumption is required for convergence proof, i.e., decrease in learning rate.

Assumptions 1 For any multi-agent system, its Nash equilibrium has the following properties [33].

1. Each state $s \in S$ and action $a^k \in A^k$ for $k = 1 \ldots n$ are reached repeatedly. Agent that takes the policy of Nash equilibrium will get more rewards when the other agents deviate from Nash equilibrium, and Nash equilibrium does not reach a global optimum

$$\Pi\pi^j(s) \cdot \sum Q^i(s) \leq \sum Q^i(s) \cdot \pi^j(s) \cdot \Pi\pi^j(s) \tag{1}$$

2. Nash equilibrium reaches a global optimum as

$$\Pi\pi^j(s) \cdot \sum Q^i(s) \geq \sum Q^i(s) \cdot \Pi\pi^j(s) \tag{2}$$

Assumption 2 Learning rate α_t fulfills the subsequent criterion for all s, t, a^1, …, a^n [33, 34].

1. $0 \leq \alpha_t \leq 1$
   It indicates that Q learning algorithm changes with possibility 1 to the best Q-function.
2. $\alpha_t\left(s, a^1, \ldots a^n\right) = 0$ if $\left(s, a^1, \ldots, a^n\right) \neq (s_t, a_t^1 \ldots a_t^n)$.
   It shows that the agent modifies simply the Q-value field relevant to present actions $a_t^1 \ldots a_t^n$ and present state s_t.

Lemma 1 *Under assumption 2, iterative process* $Q_{t+1} = (1 - \alpha_t) Q_t + \alpha_t w_t$ *will converge to* $E(w_t | h_t, \alpha_t)$ *with probability 1, where* h_t *is the historical states and policies at the time* t [34, 35].

Lemma 2 *Assume that* $(\pi^1(s), \pi^2(s), \ldots, \pi^N(s))$ *is the Nash equilibrium point for the system* $(Q^1(s), Q^2(s), \ldots, Q^N(s))$*, its Nash equilibrium reward is* $(v^1(s, \pi^1, \pi^{-1}), v^2(s, \pi^2, \pi^{-2}), \ldots, v^N(s, \pi^N, \pi^{-N}))$*, and then, the agent i's Nash equilibrium reward under joint actions is* [35, 35].

$$Q^i\left(s, a^1 \ldots a^N\right) = r\left(s, a^1 \ldots a^N\right) + \gamma \sum P(s' | s, a^1 \ldots a^N) \cdot v^i\left(s', \pi^1 \ldots \pi^i \ldots \pi^N\right)$$

Lemma 3 *Given that* $P_t^i\ Q^i(s) = r_t^i + \gamma\ Q^i(s)\ \Pi\pi^i(s)$ *where* $(\pi^1(s), \pi^2(s), \ldots, \pi^N(s))$ *is the mixed Nash equilibrium policy for the system* $(Q^1(s), Q^2(s), \ldots, Q^N(s))$, $P_t = (P_t^1 \ldots P_t^N)$ *is a contract mapping* [35–37].

Considering the above assumptions and lemmas, proposed method converges to Q-values of equilibrium point with probability 1.

Theorem *The system sequence converges to Q-value of equilibrium point* $(Q^{1*} \ldots Q^{N*})$ *with probability 1, if the system sequence* $(Q^1(s) \ldots\ Q^N(s))$ *for each* $s \in S$ $(Q^1(s) \ldots Q^N(s))$ *satisfies* $Q_{t+1}^i(s, a^1 \ldots a^N) = (1 - \alpha_t) Q_t^i(s, a^1 \ldots a^N) + \alpha_t\ [r_t^j + \gamma\, Q_t^i(s')\ \prod \pi^j(s')]$, *and Q-value meets*

$$Q^{i*}(s, a^1 \ldots a^N) = r^i(s, a^1 \ldots a^N) + \gamma \sum P(s'|s, a^1 \ldots a^N) \cdot v^i(s', \pi^{1*} \ldots \pi^{i*} \ldots \pi^{N*})$$

where $(\pi^1(s'), \pi^2(s'), \ldots, \pi^N(s'))$ *is the mixed policy for the system* $(Q^1(s'), Q^2(s'), \ldots, Q^N(s'))$ *at state* s'. $(\pi^{1*} \ldots \pi^{i*} \ldots \pi^{N*})$ *is the Nash equilibrium point for the system* [36–38].

Proof Under Lemma 2

$$\left\| P_t^i\, Q^i - P_t^i\, Q^{i*} \right\| \leq \left\| Q^i - Q^{i*} \right\|$$

Under Lemma 1

$$Q_{t+1}^k(s, a^1 \ldots a^N) = (1 - \alpha_t)(s, a^1 \ldots a^N) + \alpha_t [r_t^i + \gamma\, Q_t^i(s') \prod \pi^j(s')]$$

Converges to

$$E[r_t^i + \gamma\, Q_t^i(s') \prod \pi^j(s')] = \sum P(s'|s, a^1 \ldots a^N).(r_t^i(s, a^1 \ldots a^N) + \gamma\, Q_t^i(s') \prod \pi^j(s'))$$

Defining mapping M^i as

$$(M^i\, Q^i)(s'|s, a^1 \ldots a^N) = \sum P(s'|s, a^1 \ldots a^N).(r_t^i(s, a^1 \ldots a^N) + \gamma\, Q_t^{i*}(s') \prod \pi^j(s'))$$

then

$$M^i\, Q^i = \sum P(s'|s, a^1 \ldots a^N).P_t\, Q^i(s).$$

Under Lemma 3, as P_t is the contract according of Q^i and $P(s'|s, a^1 \ldots a^N) \geq 0$, M^i is also the contract mapping of Q^i and Q^{i*} is the fixed point of mapping M^i [39, 40].

Consider

$$\begin{aligned}\left(M^i Q^{i*}\right).(s, a^1 \ldots a^N) = \sum P(s'|s, a^1 \ldots a^N).(r_t^i(s, a^1 \ldots a^N) \\ + \gamma Q_t^{i*}(s') \prod \pi^{j*}\left(s'\right)) = r_t^i(s, a^1 \ldots a^N) \\ + \gamma \sum P(s'|s, a^1 \ldots a^N).Q_t^{i*}(s') \prod \pi^{j*}\left(s'\right))\end{aligned}$$

Under Lemma 4,

$$Q_t^{i*}(s') \prod \pi^{j*}\left(s'\right) = v^i(s', \pi^{1*} \ldots \pi^{i*} \ldots \pi^{N*})$$

So $Q^{i*} = M^i Q^i$.

Therefore the iterative formula

$$Q_{t+1}^k(s, a^1 \ldots a^N) = (1 - \alpha_t) Q_t^k(s, a^1 \ldots a^N) + \alpha_t [r_t^i + \gamma Q_t^{i*}(s') \prod \pi^{j*}\left(s'\right)]$$

Converges to $M^i Q^i = Q^{i*}$ with probability 1.

5 Cooperative Learning Model

Traffic streams of four signs with eight stream bearings are read for advancement in reality. There are four crossroads at each signal agent, as in Fig. 1, i.e., SignalAgent 1, SignalAgent 2, SignalAgent 3, and SignalAgent 4 for Ja, Jb, Jc, and Jd, respectively.

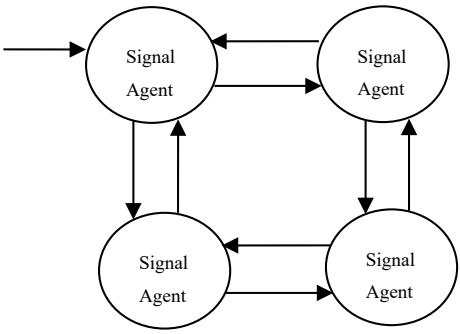


Fig. 1 Traffic run and organizes of four crossroads with flow directions

The synchronization within intersection is considered as Markov process, presented with *s*, *a*, and *r*, where *S* is the state of the intersection, a traffic control action, and *r* control agent's return.

States of System: Each agent receives real-time traffic information. It returns a traffic management judgment to represent state of road. To represent situation of road traffic, necessary data such as the number of cars incoming and the number of cars now halted at a signal are employed.

Number of cars newly incoming $= X_{\max} =$ x1, x2, x3, x4 $= 10$.
Number of cars at present halted at $J = I_{\max} =$ i1, i2, i3, i4 $= 20$.

To make the algorithm easier to understand, we will discretize a number of cars that have just arrived and a number of cars currently halted at a junction. Agent 1's state changes to (x1, i1), e.g., (5, 0), which indicates that 5 new cars are coming at agent 1 while 0 vehicles are stopped at junction 1. Agent 2's state becomes (x2, i2), agent 3's state becomes (x3, i3), and agent 4's state becomes (x3, i4) (x4, i4). Current situation of the system becomes Input as (xi, ii). Here, it can get together 200 potential states by consolidating greatest 10 showing up vehicle and limit of 20 vehicles halted at signal ($10 * 20 = 200$).

Actions of System: In the reinforcement learning framework, each strategy specifies the learning agent actions at a certain moment. In reinforcement learning, reinforcements are achieved by mapping the situation to the action. It has an impact not only on the following scene, but also on direct prizes, affecting all subsequent payouts. Distinct applications have completely different states and operations [41]. Traffic light control operations were classified into three categories in the study: no adjustment of sign span, expanding signal term, and diminishing sign length.

Value	Action
1	no adjustment of sign span
2	expanding signal term
3	diminishing sign length

Action set for signal agent 1 is $A_1 = \{1, 2, 3\}$, action set for signal agent 2 is $A_2 = \{1, 2, 3\}$, and action set for signal agent 3 is $A_3 = \{1, 2, 3\}$.

Each one is for one of the real-life traffic situation listed below. In situation of regular traffic flow, where the light control rules do not change, the approach of no change in signal length is applied [42]. The method of lengthening the signal is most commonly utilized when traffic stream on one route is regular but not on the other. There are two scenarios: increasing the signal duration to enhance traffic flow and decreasing the signal time when traffic flow on one route is lower than on others. Other routes' waiting times are lowered as a result of the reduced signal light, allowing vehicles to pass through the intersection more quickly.

Reward and Return: The problem's target is described by the reward function in reinforcement learning. The environment's visible state is linked to a value, reinforcement, and the state's internal demands. The eventual aim of a reinforcement learning agent is to get the most out of overall reinforcement over time [43]. In

this, the agent builds signal management judgments under a variety of traffic conditions and returns an action progression, with goal of reducing amount of road traffic congestion displayed by the actions of the road traffic jamming. Furthermore, in a certain traffic state, the model presents enhanced traffic management method. The traffic value display is used to assess traffic flows in this case.

The reward is determined as: current state $i = (\text{xi, ii})$ and next state $j = (\text{xj, ij})$.

1. [xi, ii] → [xi, ii−1] $[X_{\max} = 10, I_{\max} = 20] \rightarrow [X_{\max} = 10, I_{\max} = 19]$
   One of the halted cars is passing across the intersection.
2. [xi, ii] → [xi+ 1, ii−1] $[X_{\max} = 9, I_{\max} = 20] \rightarrow [X_{\max} = 10, I_{\max} = 19]$
   One car has just arrived at the junction, while another is passing by.
3. [xi, ii] → [xi, ii−3] $[X_{\max} = 10, I_{\max} = 20] \rightarrow [X_{\max} = 10, I_{\max} = 17]$
   Several stopped cars are passing across the intersection.
4. [xi, 0] → [xi+ 1, 0] $[X_{\max} = 9, I_{\max} = 0] \rightarrow [X_{\max} = 10, I_{\max} = 0]$

   There is only one new car on its way, and no cars are stopped at the intersection.

The reward is estimated by

$$\begin{aligned} \text{Reward is rp(i, p, j)} &= 1 \quad \text{if x`1} = x1 + 1 \ldots \text{for 4} \\ &= 2 \quad \text{if i`1} = \text{i1} - 1 \ldots \text{for 1} \\ &= 3 \quad \text{if i`1} = \text{i1} - 3 \ldots \text{for 2 and 3} \end{aligned}$$

Because only one presently halted car is allowed to pass the junction in case 1, the reward value for cases 2 and 3 is higher than for cases 1 and 2 because in case 1 only one currently stopped vehicle is allowed to pass the junction without regard for the arrival of the vehicle. In this scenario, however, the second agent must allow the vehicle to pass even if the first vehicle has just arrived at the intersection. The signal agent must be able to manage the vehicle's arrival and departure smartly [42, 42].

6 Results and Discussion

A controller with a learning rate of 0.5, a discount rate of 0.9 and a learning rate of 0.6 is considered. The cost was revised 1000 times with 6000 episodes during the learning process.

Figure 2 depicts delay time versus a number of states obtained by simple Q learning (without collaboration), joint action learning, and Nash-Q learning (with collaboration). In a multi-agent scenario, the delay time obtained by using cooperative approaches, such as joint action learning and Nash-Q learning, is more as that of it obtained by using simple Q learning. Sharp variation in delay is seen for simple Q learning, whereas Nash-Q and joint action learning does not have it. Delay range for joint action learning is consistent across all states, as seen in the graph.

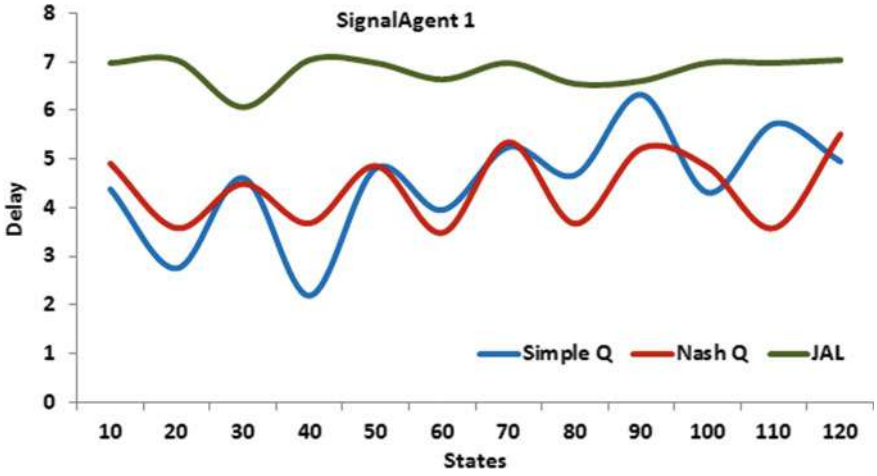


Fig. 2 States versus delay time for traffic SignalAgent 1

Figure 3 shows average rewards versus states calculated by using simple Q learning (without collaboration), joint action learning, and Nash-Q learning (with collaboration) methods for traffic signal agent 1. Average rewards received by using cooperative methods, i.e., Nash-Q and joint action learning, are more as compared to that of without cooperation method, i.e., simple Q learning for agent 1. However, within cooperation methods, it is observed that joint action learning gives slightly more rewards as compared with Nash-Q learning for the agent 1.

Figure 4 depicts delay time versus a number of states obtained by simple Q learning (without collaboration), joint action learning, and Nash-Q learning (with collaboration) for traffic signal agent 1. Delay time obtained by using cooperative

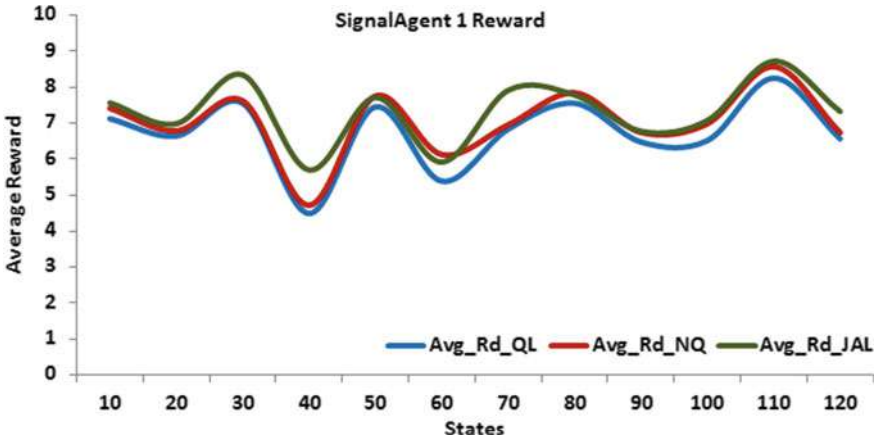


Fig. 3 Average rewards versus states for traffic SignalAgent 1

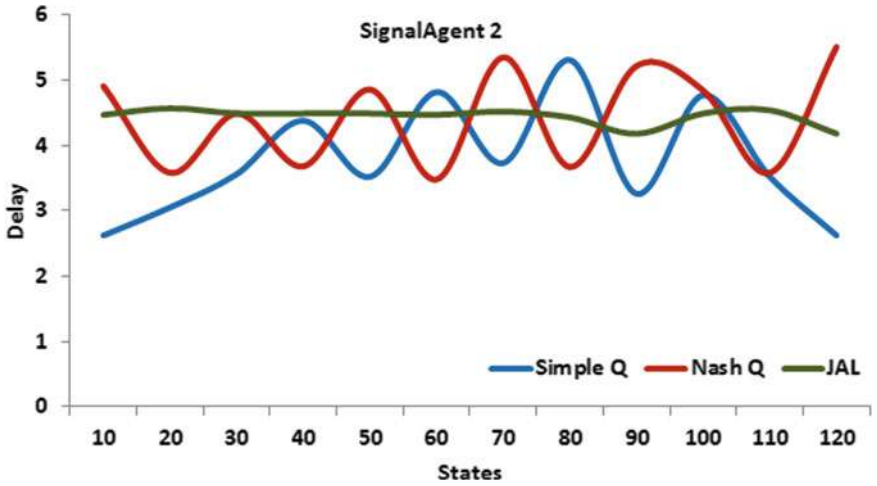


Fig. 4 States versus delay time for traffic SignalAgent 2

approaches, such as joint action learning and Nash-Q learning, is more as that of it is obtained by using simple Q learning for agent 2 in the multi-agent scenario.

Figure 5 shows average rewards versus states calculated by using simple Q learning (without collaboration), joint action learning, and Nash-Q learning (with collaboration) methods for traffic signal agent 2. Average rewards received by using cooperative methods, i.e., Nash-Q and joint action learning, are more as compared to that of without cooperation method, i.e., simple Q learning for agent 1. However,

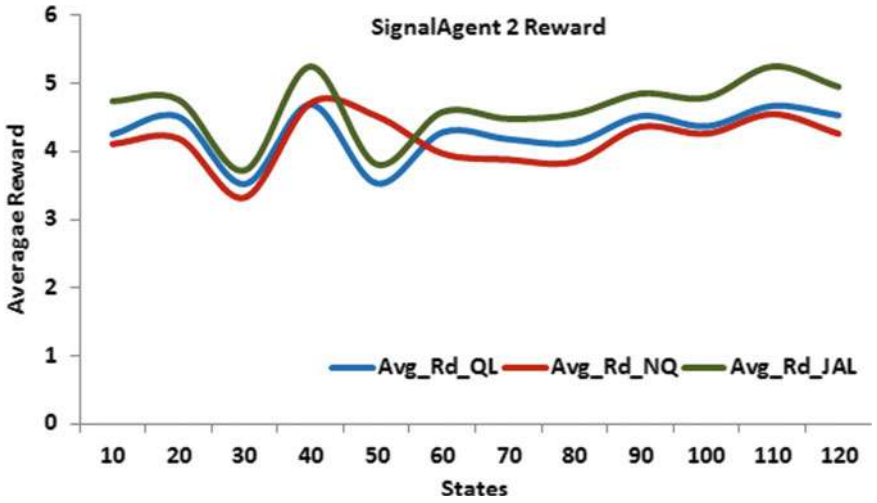


Fig. 5 Average rewards versus states for traffic SignalAgent 2

within cooperation methods, it is observed that joint action learning gives slightly more rewards as compared with Nash-Q learning for the agent 2.

Figure 6 shows delay time versus a number of states given by using simple Q learning (without collaboration), joint action learning, and Nash-Q learning (with collaboration) methods. Delay time obtained by cooperative methods, i.e., joint action learning, is much less as compared to that of without cooperation method, i.e., simple Q learning for agent 3 in the multi-agent scenario.

Figure 7 shows average rewards versus states calculated by using simple Q learning (without collaboration), joint action learning, and Nash-Q learning (with

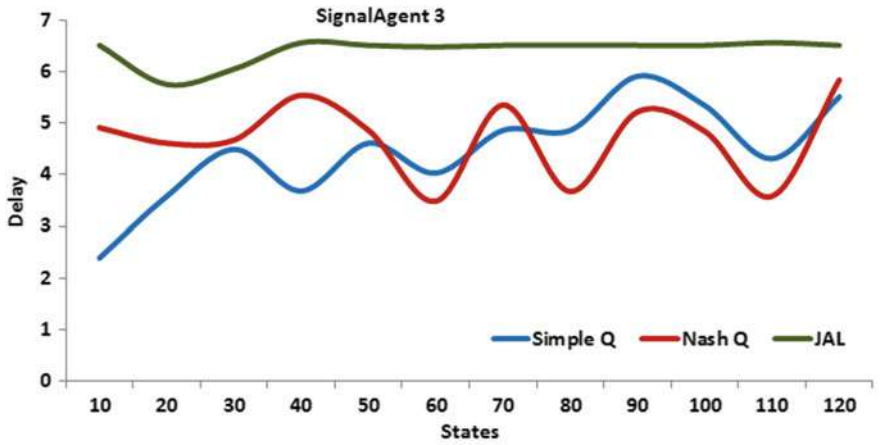


Fig. 6 States versus delay time for traffic SignalAgent 3

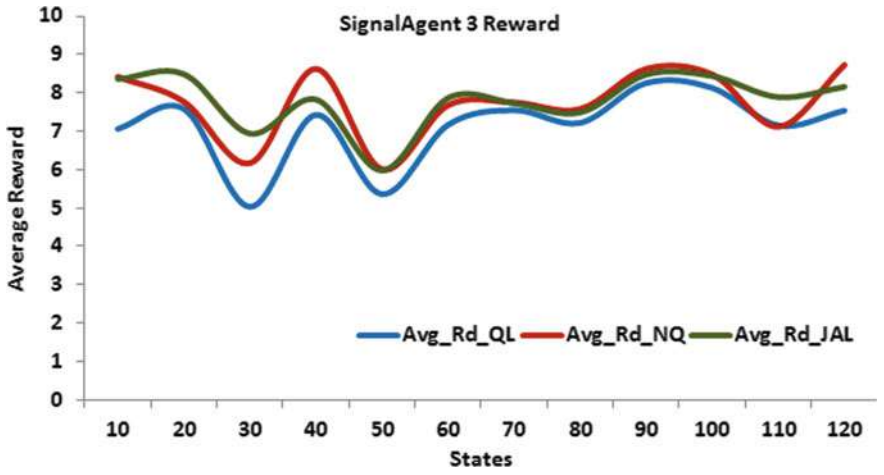


Fig. 7 Average rewards versus states for traffic SignalAgent 3

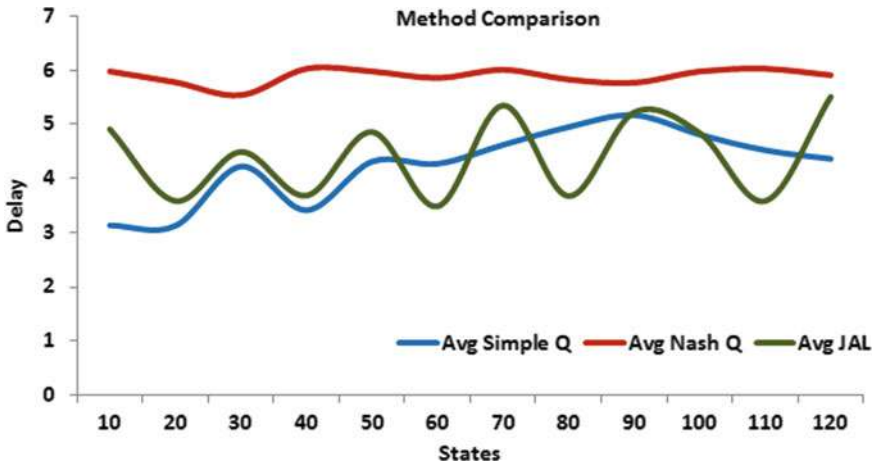


Fig. 8 State versus delay time study for multi-agent system

collaboration) methods. Average rewards received by cooperative methods, i.e., joint action learning and Nash-Q learning, are more as compared to that of without cooperation method, i.e., simple Q learning for agent 3.

Figure 8 shows states versus average delay time study for Multi-agent System of Intelligent Traffic Light Control considering all signals a system as whole by using simple Q learning, joint action learning, and Nash-Q learning methods for 120 states. Average delay time given by Nash-Q and joint action learning is more as compared to that of simple Q learning for an entire multi-agent system of traffic light control. For individual agent, Nash-Q learning gives less delay time but in cooperation with other agents Nash-Q gives significant delay time. Average delay for the multi-agent system by simple Q learning is low in initial states and gradually increases as increase in number of states. Average delay given by Nash-Q learning is stable throughout all states.

Figure 9 shows average reward analysis for Multi-agent System of Intelligent Traffic Light Control considering all signals by using simple Q learning, joint action learning, and Nash-Q learning methods for 120 states. Average rewards received by Nash-Q and joint action learning are more as compared to that of simple Q learning for an entire multi-agent system of traffic light control. There are very little variation in received rewards for entire system by applying Nash-Q and joint action learning, whereas average received by simple Q learning (without cooperation) are low in first half of the states and increased thereafter.

The findings show a link between the high volume of traffic and the importance of cooperative learning. For each traffic signal control, the Nash-Q learning and joint action learning strategy consistently produces positive results. These results make it evident in which situations the cooperative method works best. Furthermore, the findings support the hypothesis that cooperation is extremely beneficial in high traffic situations. Despite the intense traffic, the Nash-Q learning joint action learning

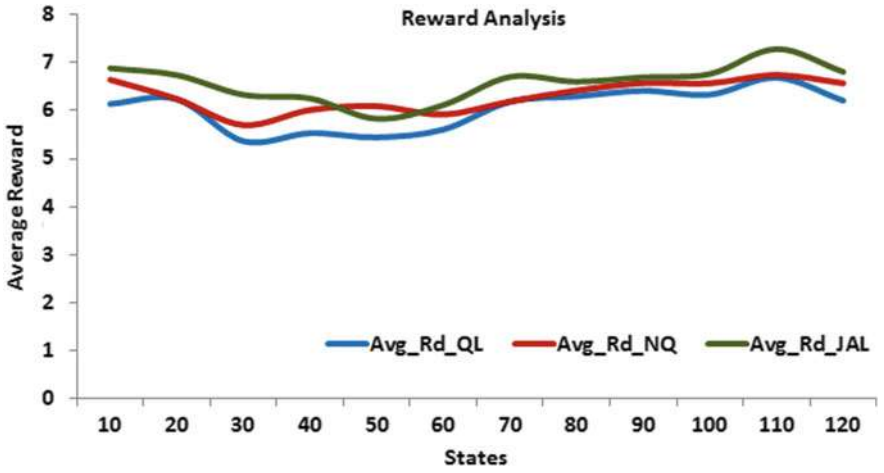


Fig. 9 Average reward analysis for multi-agent system

method achieves the same results, but at a slower rate of convergence. These techniques appear to be significantly more robust, as it can execute fine in a variety of conditions. The performance further demonstrates that combined action learning is realistic in practical scenarios by evaluating both heavy and light traffic. In fact, the result difference between cooperative methods, i.e., Nash-Q learning, joint action learning, and earlier approaches, was always greater in high traffic situations. To put it another way, as the number of agents grows, need for cooperation grows as well for intelligent traffic management.

7 Conclusion

The traffic control system is extremely complex and dynamic. Without cooperation with a preset strategy, managing traffic jams and abrupt traffic incidents for the Q learning model is difficult. Need for merging appropriate and smart traffic controller strategies with actual road traffic is growing. Reinforcement learning gathers data by maintaining open lines of communication with situations. It has strong learning skill to a complicated system, allowing it to control unidentified critical states successfully, despite the fact that it takes a long time to learn. The use of reinforcement learning algorithms in the field of traffic controller is causing increasing concern. A Framework for Smart Traffic Controller by Improved Cooperative Multi-agent Learning Algorithms (ICMALA) for smart traffic controller optimization is proposed in the paper. In the algorithm, each cooperative agent anticipates its own individual action policy by looking at the actions of other cooperative agents in the past and then makes the same decision to get the best joint action policy. Two approaches are determined, multi-agent Q learning by Nash-Q and joint action learning, and came up with a

promising result. The results show that the Nash-Q learning and joint action learning method congregates to a single policy rather quickly, resulting in higher performance.

References

1. Zhu F, Ning J, Ren Y, Peng J (2012) Optimization of image processing in video-based traffic monitoring. Elektronika ir Elektrotechnika 18(8):91–96
2. de Schutter B (1999) Optimal traffic light control for a single intersection, vol 3. In: Proceedings of the American control conference (ACC '99), June 1999, pp 2195–2199
3. Findler N, Stapp J (1992) A distributed approach to optimized control of street traffic signals. J Transp Eng 118(1):99–110
4. Baskar LD, Hellendoorn H (2012) Traffic management for automated highway systems using model-based control. IEEE Trans Intell Transp Syst 3(2):838–847
5. Vidhate DA, Kulkarni P (2017) A framework for improved cooperative learning algorithms with expertness (ICLAE). In: International conference on advanced computing and communication technologies advances in intelligent systems and computing, vol 562. Springer Singapore, pp 149–160
6. Artificial intelligence in transportation information for application, transportation research circular, Number E–C 113, Transportation on Research Board of the National Academies, January 2007
7. Abdulhai B et al (2003) Reinforcement learning for true adaptive traffic signal control. ASCE J Transp Eng 129(3):278–285
8. Wiering M (2000) Multi-agent reinforcement learning for traffic light control. In: Proceedings of 17th international conference on machine learning, pp 1151–1158
9. Guestrin C, Lagoudakis MG, Parr R (2002) Coordinated reinforcement learning. In: Proceedings nineteenth international conference on machine learning, pp 227–234
10. Kok JR, Vlassis N (2006) Collaborative multi-agent reinforcement learning by payoff propagation. J Mach Learn Res 7:1789–1828
11. Chiu S (1992) Adaptive traffic signal control using fuzzy logic. In: Proceedings of the IEEE intelligent vehicles symposium, pp 98–107
12. Spall JC (1997) Traffic-responsive signal timing for system-wide traffic control in transportation research part c: emerging technologies, 5(3):153–163
13. Foy MD, Benekohal RF, Goldberg DE Signal timing determination using genetic algorithms. Transportation Research Record No. 1365, pp 108–115
14. Shoufeng M et al (2002) Agent-based learning control method for urban traffic signal of the single intersection. J Syst Eng 17(6):526–530
15. Thorpe TL, Andersson C (1996) Traffic light control using SARSA with three state representations. Technical report, IBM Corporation
16. Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton, NJ
17. Vidhate DA, Kulkarni P (2017) Expertise based cooperative reinforcement learning methods (ECRLM). In: International conference on information and communication technology for intelligent system, Springer book series Smart innovation, systems and technologies (SIST) , vol 84. Springer Cham, pp 350–360
18. Mase K, Yamamoto H (1990) Advanced traffic control methods for network management. IEEE Mag 28(10):82–88
19. Baskar LD, de Schutter B (2011) Traffic control and intelligent vehicle highway systems: a survey. IET Intell Transp Syst 5(1):38–52
20. Vidhate DA, Kulkarni P (2016) Innovative approach towards cooperation models for multi-agent reinforcement learning (CMMARL). In: International conference on smart trends for information technology and computer communications. Springer, Singapore, pp 468–478

21. Choi W, Yoon H, Kim K, Lee S (2002) A traffic light controlling FLC considering the traffic congestion. In: Proceedings of the international conference on fuzzy systems (AFSS '02), pp 69–75
22. Vidhate DA, Kulkarni P (2016) New approach for advanced cooperative learning algorithms using RL methods (ACLA). In: VisionNet'16 proceedings of the third international symposium on computer vision and the internet. ACM DL pp 12–20
23. Zegeye S, De Schutter B, Hellendoorn J, Breunesse EA, Hegyi A (2012) A predictive traffic controller for sustainable mobility using parameterized control policies. IEEE Trans Intell Transp Syst 13(3):1420–1429
24. Vidhate DA, Kulkarni P (2017) Enhanced cooperative multi-agent learning algorithms (ECMLA) using reinforcement learning. In: International conference on computing, analytics and security trends (CAST). IEEE Xplorer, pp 556–561
25. Chin YK, Wei YK, Teo KTK (2012) Q-learning traffic signal optimization within multiple intersections traffic network. In: Proceedings of the 6th UKSim/AMSS European symposium on computer modeling and simulation (EMS '12), pp 343–348, November 2012
26. Vidhate DA, Kulkarni P (2017) Multi-agent cooperation models by reinforcement learning (MCMRL). Int J Comput Appl 176(1):25–29
27. Prashanth LA, Bhatnagar S (2011) Reinforcement learning with function approximation for traffic signal control. IEEE Trans Intell Transp Syst 12(2):412–421
28. Vidhate DA, Kulkarni P (2016) Performance enhancement of cooperative learning algorithms by improved decision-making for context-based application. In: International conference on automatic control and dynamic optimization techniques (ICACDOT). IEEE Xplorer, pp 246–252
29. Chin YK, Lee LK, Bolong N, Yang SS, Teo KTK (2011) Exploring Q-learning optimization in traffic signal timing plan management. In: Proceedings of the 3rd international conference on computational intelligence, communication systems and networks (CICSyN '11), July 2011, pp 269–274
30. Vidhate DA, Kulkarni P (2016) Enhancement in decision making with improved performance by multiagent learning algorithms. IOSR J Comput Eng 1(18):18–25
31. Ben-Akiva M, Cuneo D, Yang Q (2003) Evaluation of freeway control using a microscopic simulation laboratory. Transp Res C 11(1):29–50
32. Vidhate DA, Kulkarni P (2016) Implementation of multi-agent learning algorithms for improved decision making. Int J Comput Trends Technol (IJCTT) 35(2)
33. Al-Khatib AM (2011) Cooperative machine learning method. World Comput Sci Inform Technol J (WCSIT) 1:380–383
34. Vidhate DA, Kulkarni P (2016) A step toward decision making in diagnostic applications using single agent learning algorithms. Int J Comput Sci Inform Technol (IJCSIT) 7(3):1337–1342
35. Choi Y-C, Ahn H-S (2010) A survey on multi-agent reinforcement learning: coordination problems. In: IEEE/ASME international conference on mechatronics and embedded systems and applications, pp 81–86
36. Vidhate DA, Kulkarni P (2014) Multilevel relationship algorithm for association rule mining used for cooperative learning. Int J Comput Appl 86(4):20–27
37. Prabuchandran KJ, Kumar ANH, Bhatnagar S (2014) Multi-agent reinforcement learning for traffic signal control. In: 17th IEEE international conference on intelligent transportation systems (ITSC), pp 2529–2534
38. Vidhate DA, Kulkarni P (2014) A novel approach to association rule mining using multilevel relationship algorithm for cooperative learning. In: Proceedings of 4th international conference on advanced computing and communication technologies (ACCT-2014), pp 230–236
39. Camara M, Bonham-Carter O, Jumadinova J (2015) A multi-agent system with reinforcement learning agents for biomedical text mining. In: Proceedings of the 6th ACM conference on bioinformatics, computational biology, and health informatics, BCB'15. ACM, NY, USA, pp 634–643
40. Vidhate DA, Kulkarni P (2014) To improve association rule mining using new technique: multilevel relationship algorithm towards cooperative learning. In: International conference on

circuits, systems, communication and information technology applications (CSCITA). IEEE pp 241–246
41. Iima H, Kuroe Y (2015) Swarm reinforcement learning methods improving certainty of learning for a multi-robot formation problem. CEC, May 2015, pp 3026–3033
42. Vidhate DA, Kulkarni P (2019) Performance comparison of multi agent cooperative reinforcement learning algorithms for dynamic decision making in retail shop application. Int J Comput Syst Eng 5(3):169–178. Inderscience Publishers (IEL)
43. Vidhate DA, Kulkarni P (2019) A framework for dynamic decision making by multi-agent cooperative fault pair algorithm (MCFPA) in retail shop application. In: Information and communication technology for intelligent systems. Springer, Singapore, pp 693–703

Chapter 14
Effective Diagnosis of Cervical Cancer Using Balanced Random Forest Classifier

Asif Newaz, Farhan Shahriyar Haq, and Nadim Ahmed

1 Introduction

Cervical cancer is a type of cancer that affects the cells of the cervix and is caused primarily by different strains of the Human Papillomavirus (HPV) [1]. Other factors include smoking, HIV infection, organ transplantation, having a family history of cervical cancer, continuous oral contraceptive intake, etc. [2]. It is the fourth most commonly diagnosed cancer, as well as the fourth leading cause of cancer-related deaths in females around the globe [3]. There was a total of 604,127 new cases of cervical cancer in 2020 alone, as well as 341,831 deaths [3]. However, when diagnosed, cervical cancer is one of the most successfully treatable forms of cancer [4]. Consequently, rapid, and accurate early detection is fundamentally associated with the survival of patients.

Artificial Intelligence (AI) is becoming increasingly popular in the healthcare domain due to its enhanced ability to accurately anticipate the presence of disease in patients. A wide range of techniques has been introduced to deal with different scenarios that arise in the prediction task. Proper utilization of such techniques can yield higher accuracy with a low misclassification rate. Imbalanced data is one of the major challenges in the classification task [5]. It refers to having an unequal number of instances in different classes. Training machine learning classifiers on such imbalanced data results in biased prediction towards the majority class. Consequently, a high misclassification rate is observed in the prediction of the minority

A. Newaz (✉) · F. Shahriyar Haq · N. Ahmed
Islamic University of Technology, Gazipur 1704, Bangladesh
e-mail: asifnewaz@iut-dhaka.edu

F. Shahriyar Haq
e-mail: farhanshahriyar@iut-dhaka.edu

N. Ahmed
e-mail: nadimahmed@iut-dhaka.edu

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_15

class instances. However, accurate prediction of the rare class instances is generally more important and can have adverse consequences if predicted wrongly. In the medical domain, the minority class generally represents the diseased patients, and correctly identifying these cases is of crucial importance as the consequence of predicting an ill person as healthy can be severe. To resolve the problem of class imbalance, researchers have proposed a variety of solutions that can be broadly classified into two categories: Data-level approach and Algorithmic-level approach. The algorithmic level approach refers to altering the original algorithm in such a way as to make them cost-sensitive [6]. A higher misclassification cost is assigned to the minority class instances to shift the decision boundary away from the majority class, leading to reduced bias. The data level approach refers to sampling techniques where the original distribution of the data is modified to balance the number of instances between the classes. This is done by oversampling the minority class or by undersampling the majority class or by hybridization of the two. Many different techniques have been proposed to perform sampling. Random Oversampling (ROS), Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic (ADASYN), Random Undersampling (RUS), Edited Nearest Neighbors (ENN) are some popular sampling methods [7–9]. Each one undertakes a different strategy to perform sampling.

The effectiveness of the sampling approach relies largely on the way the data is modified. ROS and RUS are the two simplest strategies to resample the data. However, simple duplication of the minority class instances with ROS results in overfitting, while randomly removing samples from the majority class causes loss of information. Heuristic sampling techniques like SMOTE or ENN have been proposed to alleviate such problems. These heuristic approaches have been proven to be useful in many imbalance classification problems but they still have certain limitations [10, 11]. These approaches either oversample the minority class or exclude instances from the majority class entirely in order to get a balanced distribution in the data. The creation of a substantial number of synthetic instances results in minority class overfitting while eliminating instances from the majority class in this manner leads to information loss, since a significant portion of the majority class is not utilized throughout the training process. To prevent such loss of information or overfitting, the sampling process can be merged with the ensemble learning framework [12]. A number of such techniques such as bagging or boosting-based ensemble have been proposed by researchers [13, 14]. Ensemble learning is a machine learning approach in which various models, termed "Weak learners," are integrated to enhance their individual performance. In general, weak learners have a high bias or a high variance. Aggregating multiple models can reduce these effects yielding higher prediction accuracy. However, ensemble models by themselves still remain susceptible to the imbalanced learning problem. Integrating sampling strategies into the ensemble learning framework can mitigate this problem and provide a balanced prediction performance.

In this study, an effective diagnostic tool has been developed for the prediction of cervical cancer using a benchmark cervical cancer dataset collected from the UCI Machine Learning repository [15]. The dataset comprises medical records, habits,

history of sexual life, and demographic information of 858 individuals. The number of positive cases is only 55 compared to 803 negative cases. Such an imbalance scenario is quite common in medical datasets as the number of patients with the disease is generally much lower. However, building a predictive model on such imbalanced data requires some careful considerations. To balance the dataset, a large number of synthetic samples need to be produced using SMOTE or similar approaches. On the other hand, a large number of majority class samples need to be removed using undersampling techniques to achieve sufficient balance. These lead to overfitting or loss of information reducing the predictive power of the machine learning model. To alleviate this problem and to ensure the quality of the resampled data upon which the machine learning model would be trained on, we propose to merge the undersampling process with the construction of an ensemble classifier, specifically the Random Forest (RF) classifier [13]. Generating bootstrap samples from the original dataset is referred to as bagging. In an RF classifier, a large number of trees are grown, each of which is trained on a bootstrap sample of the original data. Decision trees are extremely prone to overfitting and are often biased in the presence of imbalanced data. By creating an ensemble of such trees this way mitigates the problem of overfitting. To reduce the effect of class imbalance, undersampling was performed on each bootstrap sample. The sampling rate was determined using the grid-search technique. The training set for each tree is constructed using this balanced bootstrap data. Thus, trees are trained on a balanced random subset of the data in this manner. By aggregating the prediction from all the trees in the forest, A reliable ensemble model may be constructed that overcomes the issue of class imbalance while simultaneously mitigating the impact of undersampling. As a result, our proposed Balanced Random Forest (BRF) classifier is capable of delivering superior predictive performance while maintaining a high degree of accuracy in diagnosing cervical cancer compared to other popular approaches.

2 Literature Review

Diagnosis of cervical cancer using pap-smear images is the conventional approach. Different machine learning techniques have been utilized for the automatic analysis of pap-smear images, resulting in a quicker and more consistent screening process [16, 17]. Researchers have also used MRI (Magnetic Resonance Imaging) images for the detection of cervical cancer [18]. The advancement of computer-assisted technologies, such as molecular dynamics simulation methods, has allowed for a better understanding of new molecular processes in diseases like cervical cancer. Comprehensive molecular and integrative analysis has allowed for the identification of several new genomic and proteomic features of cervical cancer subtypes [19].

Although the procedures listed above have revolutionized cervical cancer diagnosis, there are major concerns over the complexity and expense involved with them along with the non-availability that may be present in low-income countries where 95% of cervical cancer fatalities occur [3]. In this regard, diagnosing cervical cancer

by evaluating risk factors identified during preliminary screening tests can provide a cheaper and flexible alternative. In 2017, a dataset comprised of risk factors for cervical cancer collected from the Hospital Universitario de Caracas in Venezuela was made public. The dataset contains medical records, habits, history of sexual life, and demographic information of 858 female individuals. Several researchers have attempted to build a predictive model for cervical cancer using this information. Ijaz et al. [20] developed a detection method that uses SMOTE and SMOTE with Tomek Link for balancing the data and Random Forest for classification. Geetha et al. [21] also utilized SMOTE to resolve the class imbalance, and then applied the feature reduction techniques—Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) to predict the existence of cervical cancer. Wu et al. [22] proposed two approaches: SVM-RFE and SVM-PCA for the diagnosis of cervical cancer. Moldovan et al. [23] utilized the Chicken Swarm Optimization (CSO) technique to select the hyperparameters of the SVM classifier. In a recent study, Lu et al. [24] proposed an ensemble method for predicting cervical cancer. Their technique integrates a logical data correction mechanism with a probability-based voting strategy that utilizes the output probability of each base classifier in order to diagnose cervical cancer. They reported a recall score of 28.35% and an accuracy of 83.16%. However, due to the extremely imbalanced nature of the dataset, this high accuracy is too optimistic. The recall score, which indicates the classifier's effectiveness in predicting positive instances, is significantly lower. A better screening tool is required that ensures an effective distinction between the patients and healthy individuals. In this regard, we have developed a reliable decision support system for the accurate diagnosis of cervical cancer using the BRF classifier. Our proposed approach effectively handles the class imbalance present in the data and provides a balanced prediction performance to properly distinguish cancer patients from healthy ones.

3 Materials and Methods

3.1 Dataset Description

To build the diagnostic model for cervical cancer, this study utilizes the Cervical Cancer Risk Factor dataset available at the UCI repository [15]. The data is originally collected from Hospital Universitario de Caracas in Venezuela. The dataset contains medical records, habits, history of sexual life, and demographic information of 858 female individuals. There is a total of 32 distinct features. 4 different tests were performed on the individuals to diagnose cervical cancer. The result of the biopsy is considered the gold standard and has been utilized in this study. A description of the dataset is provided in Table 1.

Owing to privacy concerns, some chose not to answer certain questions. Hence the data contains missing entries. The number of missing entries is also reported

Table 1 Description of the dataset

Number	Feature information	Summary statistics	Number of missing entries
1	Age	Range: 13–84 Mean: 26.821	0
2	Number of sexual partners	Range: 1–28 Mean: 2.528	26
3	First sexual intercourse	Range: 10–32 Mean: 16.995	7
4	Number of pregnancies	Range: 0–11 Mean: 2.276	56
5	Smokes	0 = (722 patients) 1 = (123 patients)	13
6	Smokes (years)	Range: 0–37 Mean: 1.220	13
7	Smokes (packs/yeas)	Range: 0–37 Mean: 0.453	13
8	Hormonal contraceptives	0 = (269 patients) 1 = (481 patients)	108
9	Hormonal contraceptives (years)	Range: 0–30 Mean: 2.256	108
10	IUD (Intrauterine device)	0 = (658 patients) 1 = (83 patients)	117
11	IUD (years)	Range: 0–19 Mean: 0.515	117
12	STDs (Sexually transmitted diseases)	0 = (674 patients) 1 = (79 patients)	105
13	STDs (number)	0 = (674 patients) 1 = (34 patients)	105
14	STDs: condylomatosis	0 = (709 patients) 1 = (44 patients)	105
15	STDs: cervical condylomatosis	0 = (753 patients) 1 = (0 patient)	105
16	STDs: vaginal condylomatosis	0 = (749 patients) 1 = (4 patients)	105
17	STDs: vulvo-perineal condylomatosis	0 = (710 patients) 1 = (43 patients)	105
18	STDs: syphilis	0 = (735 patients) 1 = (18 patients)	105
19	STDs: pelvic inflammatory disease	0 = (752 patients) 1 = (1 patient)	105
20	STDs: genital herpes	0 = (752 patients) 1 = (1 patient)	105
21	STDs: molluscum contagiosum	0 = (752 patients) 1 = (1 patient)	105

(continued)

Table 1 (continued)

Number	Feature information	Summary statistics	Number of missing entries
22	STDs: AIDS	0 = (753 patients) 1 = (0 patients)	105
23	STDs: HIV	0 = (735 patients) 1 = (18 patients)	105
24	STDs: Hepatitis B	0 = (752 patients) 1 = (1 patient)	105
25	STDs: HPV	0 = (751 patients) 1 = (2 patients)	105
26	STDs: Number of diagnosis	0 = (787 patients) 1 = (68 patients)	0
27	STDs: Time since first diagnosis	Range: 1–22 Mean: 6.141	787
28	STDs: Time since last diagnosis	Range: 1–22 Mean: 5.817	787
29	Dx: Cancer	0 = (840 patients) 1 = (18 patients)	0
30	Dx: CIN	0 = (849 patients) 1 = (9 patients)	0
31	Dx: HPV	0 = (840 patients) 1 = (18 patients)	0
32	Dx	0 = (834 patients) 1 = (24 patients)	0

in Table 1. Two features in particular, STDs: Time since first diagnosis and STDs: Time since last diagnosis, have a total number of 787 missing records. These two features were not included in this study due to the substantial number of missing entries (>90%). Moreover, a total number of 105 patients have missing entries on 18 features. These patients were also excluded from the study because more than half of their feature information was missing. The rest of the missing entries were imputed using the Multivariate Imputation by Chained Equations (MICE) algorithm, which uses multivariate regression for data imputation [25]. After handling the missing data, the final dataset obtained contained information of 753 patients with 30 features. The number of positive and negative cases in the data is 53 and 700, respectively.

3.2 Proposed Methodology

To develop an effective diagnostic model, the quality of the data needs to be good. The presence of missing entries in the data might impair the prediction power of a classification algorithm as well as produce biased estimates, leading to invalid conclusions. Therefore, the dataset was first carefully processed for missing entries.

Using a stratified tenfold cross-validation approach, the dataset was then divided into training and testing sets. This technique divides the data arbitrarily into ten folds while maintaining the proportion of examples for each class on the training and testing folds. This guarantees that each fold retains the same data distribution as the original dataset. Following that, a total of N bootstrap samples was generated from each training set. The class imbalance was also present in the bootstrap samples as was in the original set. To achieve equal class distribution, random undersampling was applied to the majority class. The decision trees were then trained on the balanced bootstrap samples and the forest was grown. The construction of this BRF classifier is described in detail in Sect. 3.3. The performance of the proposed BRF classifier was tested on each testing fold and the average of the 10 results was taken. Four distinct quantifiable metrics—Accuracy, Sensitivity, Specificity, and G-mean score—were used to evaluate the proposed method's performance. Figure 1 depicts the outline of the proposed framework.

3.3 Balanced Random Forest Classifier (BRF)

The construction of the BRF classifier is described below:

Step 1: The original dataset is used to produce N number of bootstrap samples. In this study, N was set to 100.

Step 2: To balance the class distribution in the data, random undersampling is applied to the majority class of each bootstrap sample. The sampling rate (ratio of the number of instances in the minority class to the number of instances in the majority class after sampling) was taken as 0.9. The grid search technique was utilized to obtain the most suitable sampling rate.

Step 3: N decision trees are then trained on each balanced bootstrap sample. Since undersampling is performed on each bootstrap sample separately, not on the entire dataset, the problem of loss of information due to undersampling is largely reduced.

Step 4: A random subset of features is chosen at each node of each tree for evaluation. The Gini Index is measured to determine which feature maximizes the metric to go in that node. Growing uncorrelated trees this way ensures better generalization which is a prime aspect of the random forest classifier. The process is repeated until the entire tree has been created

Step 5: Steps 2 to 4 are repeated for all the trees in the forest. As a result, a large forest with a diverse range of trees is developed.

Step 6: Each tree makes a decision for every sample in the test set in order to produce a prediction. The final prediction is the one that the trees in the forest make the most frequently.

The construction process is illustrated in Fig. 2.

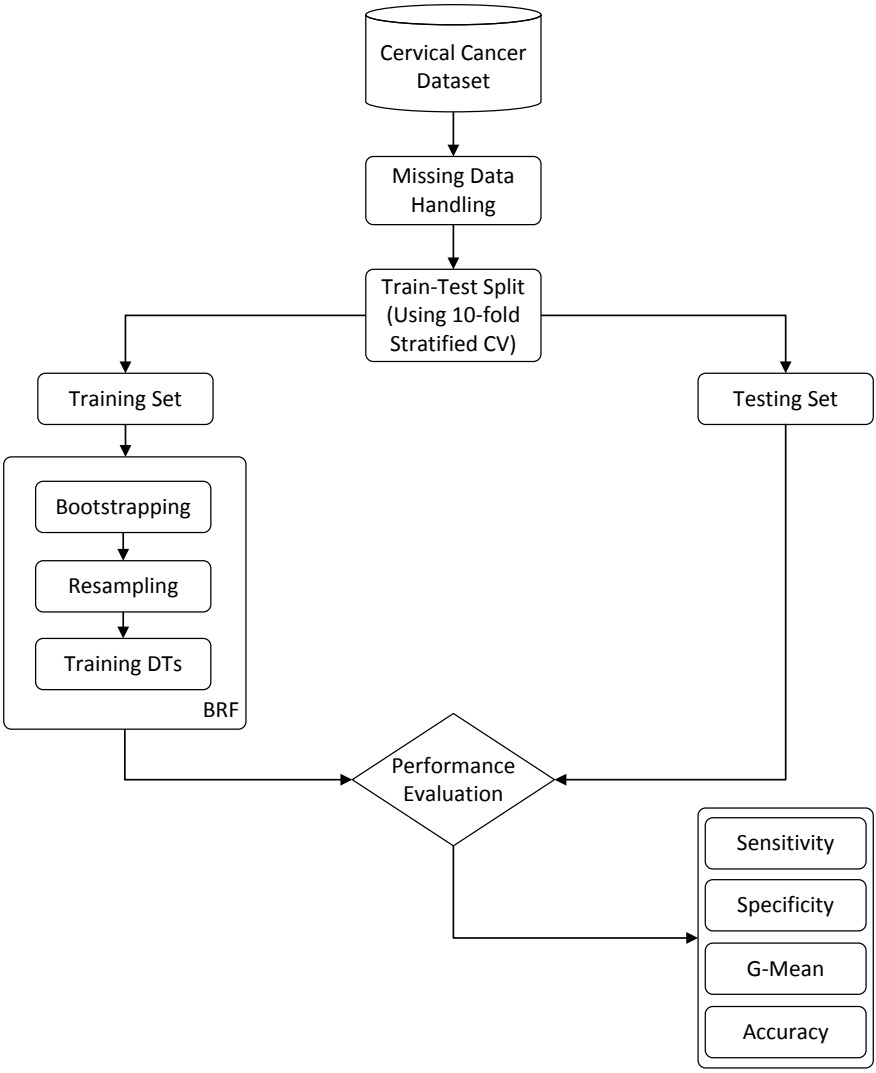


Fig. 1 Outline of the proposed framework

3.4 Performance Metrics

Performance indicators like classification accuracy get biased towards the majority class if the data is skewed. Class-specific performance metrics are required to properly assess the classifier's performance. Therefore, four different metrics: sensitivity, specificity, g-mean score, and accuracy were considered for evaluation. Sensitivity measures minority class performance, while specificity assesses how effectively the

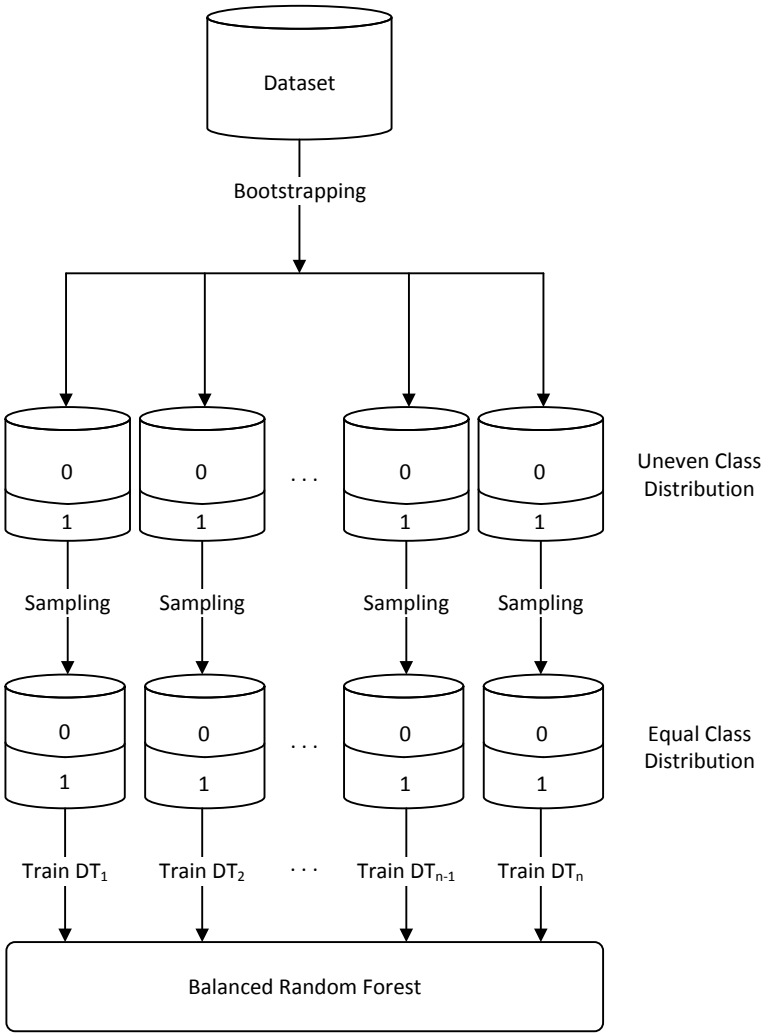


Fig. 2 Construction of BRF classifier

negative class was predicted. The g-mean score is a combination of the two and in order to have a high score, it is necessary to have high sensitivity and specificity.

4 Results and Discussion

Class imbalance poses a major challenge in classification tasks. Conventional classification algorithms are not built to cope with situations when there is a significant

Table 2 Performance of RF classifier

	Random forest (RF)
Sensitivity (%)	0.00
Specificity (%)	99.71
G-mean (%)	0.00
Accuracy (%)	92.70

imbalance in the data. The prediction becomes biased towards the majority class due to the disparity, resulting in a high minority class misclassification rate. The problem intensifies when the imbalance ratio is large and the number of minority class samples is lower. Special strategies need to be adopted while dealing with such datasets to avoid biasing towards to majority class and attain a high accuracy on the minority class prediction.

The dataset that we have utilized in this study contains a considerable amount of disparity between the positive and negative classes (53 positive samples and 700 negative samples). Under such conditions, the standard Random Forest (RF) classifier performs very poorly as can be observed from Table 2. Sensitivity and specificity values are found to be 0.00 and 99.71% respectively, which elucidates that the standard RF classifier completely fails to correctly identify any minority class samples. It is also evident that for an imbalanced dataset like this, the accuracy measure is quite irrelevant. The accuracy value obtained is 92.70%, which is clearly over-optimistic. The G-mean value, which provides a balanced measure between majority and minority classes, is found to be 0.00%. It indicates that the model is completely biased due to the high imbalance present in the data.

To address such an imbalance scenario, sampling strategies can be adopted. SMOTE is a widely used sampling method for imbalanced classification. For balancing the skewed dataset, this method oversamples the minority class instances. After that, the classifier is trained on the balanced data, which eliminates the bias introduced by the majority class. The performance measures obtained with the RF classifier when the training data was first resampled using the SMOTE algorithm are provided in Table 3. The RF classifier performs marginally better when trained on resampled data. However, the sensitivity score is still extremely low, only 5.67%. This low value of sensitivity even after applying SMOTE is due to the fact that the number of minority class samples is comparatively much lower than the dominant class. To balance the dataset using oversampling requires the generation of a

Table 3 Performance of RF classifier after applying SMOTE

	SMOTE + RF
Sensitivity (%)	5.67
Specificity (%)	96.43
G-mean (%)	12.77
Accuracy (%)	90.04

considerable number of augmented samples. This overfits the data and therefore the classifier is performing poorly on the unseen data.

Another popular approach while dealing with imbalanced data is to randomly undersample (RUS) the majority class. One major drawback of this strategy is the loss of information due to the removal of majority class samples. The performance measures obtained with the RF classifier when the majority class of the training data was first randomly under-sampled are provided in Table 4. This approach appears to be quite effective, providing a sensitivity score of 52%. However, the specificity score dropped to 62.29% as a substantial number of majority class samples were removed. The g-mean score obtained is 48.87%.

Although this is a considerable improvement compared to the previous two approaches, performance can be further improved by incorporating the sampling strategy into the construction of the RF classifier. When RUS is applied to the whole majority class, a significant amount of information is discarded. This may be prevented by undersampling each bootstrap sample generated during the RF classifier's model development process. The issue of information loss due to undersampling is mitigated this way since samples are drawn with replacement, that is, samples are returned to the population after being utilized to construct a sample set before the next unit is drawn. This method provides better generalizability and minimal information loss while also eliminating bias induced by the imbalanced nature of data. Table 5 shows the performance metrics acquired using this method. As seen in Table 5, when our suggested BRF classifier was implemented, performance increased substantially. The sensitivity score achieved is 58%, which is a remarkable increase over the standard RF classifier's sensitivity value of 0.0%. The sensitivity is much higher than the other two popular sampling methods as well. Furthermore, a specificity score of 70.86% indicates that our model improves the minority class performance without significantly compromising the performance on the majority class. This implies that the prediction performance is balanced, which is desirable in a dependable decision support system.

Table 4 Performance of RF classifier after applying RUS

	RUS + RF
Sensitivity (%)	52.00
Specificity (%)	62.29
G-mean (%)	48.87
Accuracy (%)	61.51

Table 5 Performance of BRF classifier

	Balanced random forest (BRF)
Sensitivity (%)	58.00
Specificity (%)	70.86
G-mean (%)	59.36
Accuracy (%)	70.00

Table 6 Performance of RUSBoost classifier

	RUSBoost
Sensitivity (%)	45.67
Specificity (%)	61.29
G-mean (%)	47.03
Accuracy (%)	60.14

Comparison of our presented model with other popular approaches used in imbalance classification such as SMOTE or RUS has already been presented before in Tables 3 and 4. Another popular ensemble method for imbalance classification is the RUSBoost classifier [14]. It works in a similar manner but uses a boosting framework for prediction. Table 6 highlights the performance metrics achieved with this classifier. As is readily apparent, the g-mean score obtained is 47.03%, which is comparatively lower than the proposed BRF classifier (59.36%). Therefore, it is evident that our proposed method outperforms all the other popular approaches making it more suitable for the diagnosis of cervical cancer.

Figure 3 compares the G-mean scores acquired by the BRF classifier to those obtained by the standard RF classifier, RF with over-sampled data (SMOTE), RF with under-sampled data (RUS), and RUSBoost classifier.

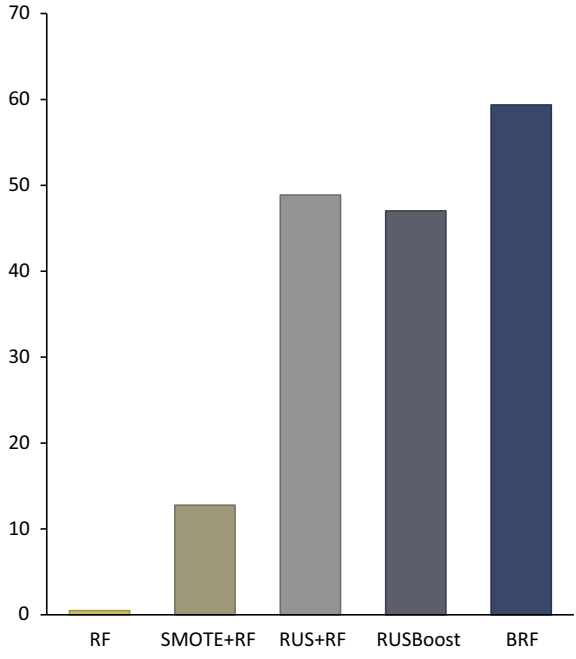


Fig. 3 Performance comparison of our proposed approach with other techniques in terms of G-mean score

5 Conclusion

This paper presents a robust decision-support system for the early diagnosis of cervical cancer using clinical features. The undersampling process is incorporated into the construction of the random forest classifier to effectively handle the large class imbalance present in the data which is a frequent scenario in many medical diagnosis tasks. The developed model provides a more generalizable and balanced prediction performance on both the majority and minority classes. This ensures an effective distinction between the patients and healthy individuals. In the future, we would like to extend our work and employ different feature selection techniques to identify the key risk factors for the prediction of cervical cancer. Using the most representative subset of features might also increase the classification algorithm's prediction ability.

References

1. Cervical cancer—Symptoms and causes—Mayo Clinic (2020). https://www.mayoclinic.org/diseases-conditions/cervical-cancer/symptoms-causes/syc-20352501. Last Accessed 12 Sep 2021
2. Hillemanns P, Soergel P, Hertel H, Jentschke M (2016) Epidemiology and early detection of cervical cancer. Oncol Res Treatm 39(9):501–506
3. Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, Bray F (2021) Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer J Clinicians 71(3):209–249
4. World Health Organization (WHO) (2021) WHO health topics: cervical cancer. https://www.who.int/health-topics/cervical-cancer#tab=tab_1. Last Accessed 12 Sept 2021
5. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
6. Ling CX, Sheng VS (2008) Cost-sensitive learning and the class imbalance problem. Encyclopedia of Machine Learn 2011:231–235
7. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. J Artif Intel Res 16:321–357
8. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, pp 1322–1328
9. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern 3:408–421
10. Fotouhi S, Asadi S, Kattan MW (2019) A comprehensive data level analysis for cancer diagnosis on imbalanced data. J Biomed Inform 90:103089
11. Sun T, Zhang R, Wang J, Li X, Guo X (2013) Computer-aided diagnosis for early-stage lung cancer based on longitudinal and balanced data. PLoS ONE 8(5):e63559
12. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. Pattern Recogn 48(5):1623–1637
13. Chen C, Liaw A, Breiman L (2004) Using random forest to learn imbalanced data. University of California, Berkeley 110(1–12):24
14. Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A (2009) RUSBoost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybernetics-Part A: Syst Humans 40(1):185–197

15. UCI Machine Learning Repository (2021) https://archive.ics.uci.edu/ml/index.php. Last Accessed 29 July 2021
16. Wang P, Wang L, Li Y, Song Q, Lv S, Hu X (2019) Automatic cell nuclei segmentation and classification of cervical Pap smear images. Biomed Signal Process Control 48:93–103
17. Su J, Xu X, He Y, Song J (2016) Automatic detection of cervical cancer cells by a two-level cascade classification system. Anal Cellular Pathol
18. Sneha K, Arunvinodh C (2016) Cervical cancer detection and classification using texture analysis. Biomed Pharmacol J 9(2):663–671
19. HLA-B N (2017) Integrated genomic and molecular characterization of cervical cancer. Nature 543:16
20. Ijaz MF, Attique M, Son Y (2020) Data-driven cervical cancer prediction model with outlier detection and over-sampling methods. Sensors 20(10):2809
21. Geetha R, Sivasubramanian S, Kaliappan M, Vimal S, Annamalai S (2019) Cervical cancer identification with synthetic minority oversampling technique and PCA analysis using random forest classifier. J Med Syst 43(9):1–19
22. Wu W, Zhou H (2017) Data-driven diagnosis of cervical cancer with support vector machine-based approaches. IEEE Access 5:25189–25195
23. Moldovan D (2020) Cervical cancer diagnosis using a chicken swarm optimization based machine learning method. In: 2020 international conference on e-health and bioengineering (EHB), pp 1–4.https://doi.org/10.1109/EHB50910.2020.9280215
24. Lu J, Song E, Ghoneim A, Alrashoud M (2020) Machine learning for assisting cervical cancer diagnosis: an ensemble approach. Futur Gener Comput Syst 106:199–205
25. Raghunathan TE, Lepkowski JM, Van Hoewyk J, Solenberger P (2001) A multivariate technique for multiply imputing missing values using a sequence of regression models. Surv Methodol 27(1):85–96

Chapter 15
Asthma Detection System: Machine and Deep Learning-Based Techniques

Mohammed Tawfik, Nasser M. Al-Zidi, Ibraheam Fathail, and Sunil Nimbhore

1 Introduction

Asthma is a lung condition affecting the airways, i.e., the tubes that carry air in and out of the lungs, which results in difficulty in breathing with wheezing and irritation in the airways [1]. The syndrome of that disease differs from one person to another, and its acuteness is normally affected by the variable of age. Chest tightness, coughing, lack of breath, and wheezing are the most frequent symptoms. Infections, allergies to some food, smoking and dust, humidity, and chemical fumes inhaled allergens, such as pollen from grass and trees are highly connected to the intensity of the symptoms. According to the World Health Organization (WHO), approximately 262 million people were affected by asthma in 2019 [2]. Each year, more than 3 million people die of COPD, which is known as chronic obstructive pulmonary disease (COPD) [2]. In India, about 15–20 million people are asthmatics while in USA Asthma disease affects 9.6% of children. According to Global Burden of Disease (GBD) in 2008–2010, the number of people who had Asthma around the world reached 334 million in the USA. The total cost of asthma medications was estimated to be $56 billion in 2008. European study in 2011 estimated actual worth of Asthma to be around €19.3 billion [3].

The most symptom of asthma is wheezing [4] that is produced when moving air, they react with an airway. However, breathing sounds may provide valuable details on the condition of the lungs. It is become possible for the doctor's experience

M. Tawfik (✉) · I. Fathail · S. Nimbhore
Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University Aurangabad, Aurangabad, India
e-mail: kmkhol01@gmail.com

M. Tawfik
Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen

N. M. Al-Zidi
Faculty of Administrative and Computer Sciences, Albaydha University, Albaydha, Yemen

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_16

and precision to have a great impact on the diagnosis bronchial tubes have several bronchial areas, back, and lateral, and bronchial counterparts. The doctor must place the stethoscope accurately to differentiate the sounds of the patient's internal organs from environmental noises. From this, it can be assumed that the electronic stethoscope, which can record and store lung sounds, not only stores the data but helps to interpret diseases with the help of Machine learning and deep learning. Wheezing is a respiratory noise that is continuously Superposed on the normal breath. The wheezing has different significant features than Normal breath. The more precise type of waveform analysis to identify that wheezing is a sinusoidal wave with a domain frequency between 100 and 5000 Hz [5] with a peak-to-peak time interval between at least 400 ms.

Many studies have been conducted to scientifically describe wheezing, extract the important features, and process them by computerized signal processing in the frequency domain, time–frequency, and analysis. Some previous studies developed machine learning and deep learning models for predicting asthma through lung sounds and wheezing. Still, there were some limitations in prediction accuracy, and those models did not detect asthma in real-time.

Contrary to previous studies, our study presents a novel system that can diagnose asthma diseases with high accuracy using artificial intelligence (AI). We have implemented the deep learning technique on respiratory sounds as waveforms collected from several asthmatics and non-asthmatics and classified normal and abnormal sounds. Then we deployed that model to make it capable of predicting the new sound in real-time as a web interface and the developed mobile app, thus can help doctors diagnose patients with high accuracy.

This paper is organized as follows: Sect. 2 narrates some recent works related to ours; Sect. 3 provides an overview of the system and architecture; Sect. 4 describes the data acquisition and pre-processing; Sect. 5 details System Implementation and Model deployment; Sect. 6 shows the results and performance evaluation of the deep learning model; while Sect. 7 reports the conclusion.

2 Related Work

Breath sound analysis has, to date, been aided by various signal processing and machine learning techniques. In this section, a few of the most prominent studies on breath sound analysis will be reviewed.

Radwa et al. [6] have used Time-dependent frequency on a public dataset that consists of 143 lung sound Wav files collected by stethoscope, some of the files are mono channel and Multichannel, the proposed approach performed using and consists of a two-part respiration detection and wheeze detector, then identify the wheezing Monophonic or Polyphonic the obtain result was the accuracy of 91% with sensitivity of 91 and specificity of 90%.

Some studies used Spectral and Power spectrum analysis for audio files; Gadge, Rode [7, 8] have developed a System for Wheeze Detection based on Spectral Analysis for sounds signals that collected from trachea location using a microphone called Sony ECM-77B for 110 subjects 100 Normal and 10 Asthmatics and represent spectrogram on time domain and time–frequency (TF), with sampling rate 10,000 Hz.

Most studies often used first to turn the file into a spectrogram image for the classification of sounds and then use a neural network to process it. However, Himadri et al. [9] used The Linear Predictive Cepstral Coefficients for features extracted from audios files on Dataset called (ICBHI17), after divided each segment into frames each one consists of 256 samples and 100 point articulation/overlap each frame, applied The hamming window and calculate the sum of Linear prediction coefficients (LPC) in a frequency ranges for the frames and feed The Multi-layer Perceptron classifier (MLP), The achieved accuracy of 99.22%.

Class the wheezing based on severity levels Fizza et al. [10] based on integrated power (IP), The data collected for 55 asthmatic subjects using a wireless digital stethoscope; the sounds are taken from the left and right lower lung base and trachea, and represent each sound using spectrogram, segmented the spectrogram with criteria by a physician to find the severity level of asthma, wheezing labeled according. The wheeze by a Fast Fourier Transform (FFT) To 512 hamming windows to get power spectrum in the range between 100–1600 Hz, calculate the sum of absolute value for each power spectrum, and have applied statistical methods called multivariate analysis MANOVA to find the level of severity. The overall performance for Machine learning algorithms was as the following are the severity levels for mild, moderate, and severe 100% (KNN Classifier), 92% (SVM Classifier), and 94 Totally. The study conducted by Rajkumar et al. [10] has used the wavelet packet transform for feature extraction on audio collected from subject's total of 72:29 female and 43 males, evaluated those features using statistical method one-way analysis of variance (ANOVA). The obtained accuracies of 98.25 and 99.25% using conventional validation (CV).

3 Asthma Diagnosis System

3.1 Overview of the System

Researchers have put much effort into developing healthcare applications in the last few years, that make individuals better monitor and manage health through healthcare applications [11]. Healthy living is about much more than simply managing and monitoring one's health; people also get benefit from it. Machine learning, deep learning, cloud computing, and IoT [12] are the heart of this development in healthcare applications, rapidly spreading throughout the medical industry. Nowadays, they are changing the form of healthcare filed. These technologies are interconnected so that each of these techniques offers something unique to the medical industry, which is changing the way medical professionals manage patient care and their own

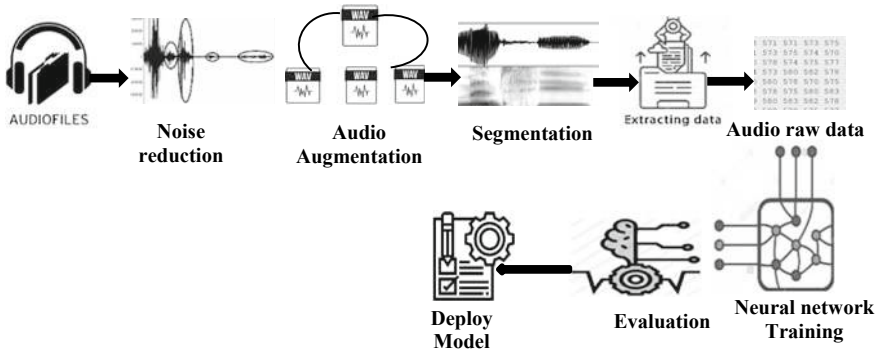


Fig. 1 Deep learning model architecture

roles. While Machine learning and Deep learning are the most well-known of technical terms, it can analyze large amounts of data much faster and more accurately than humans. one of the applications of healthcare is Asthma monitor or diagnosis, which utilizes a stethoscope or Mic for diagnosing and assessing the severity level of asthma. We implemented an asthma detection system on sound records collected from subjects from different locations on the chest and trachea on inhaling and exhale situation, we processed these files and removed the noise, segment audio to slice (clip), performed features extraction to get raw data from these audios, feed it to the deep neural network, we have used a stethoscope with a mic to be digital after deployed the model as a web interface. This framework can help the physician to detect asthma directly. The following phases show system details (Fig. 1).

4 Proposed System

4.1 Data Acquisition

This paper evaluates the proposed system on data collected from Chronic Obstructive Pulmonary Disease subjects [13-15] The data was collected from the chest zone, upper, middle, and lower, inhale and exhale situation, and trachea by digital stethoscope and mic. The subjects for both males and females were 85 asthmatics and 42 healthy people ranging in age from 21 to 90. The recorded files were in Wav format with an average duration of 6–25 s.

Pre-processing phase

The respiratory datasets contained various subjects related to lung details. We have selected Asthmatic subjects that are characterized by wheezing and crackling during a breath. It is oblivious that breathing sounds contain noises such as heartbeat and other sounds. We processed it using an audio processing program called “Audacity”

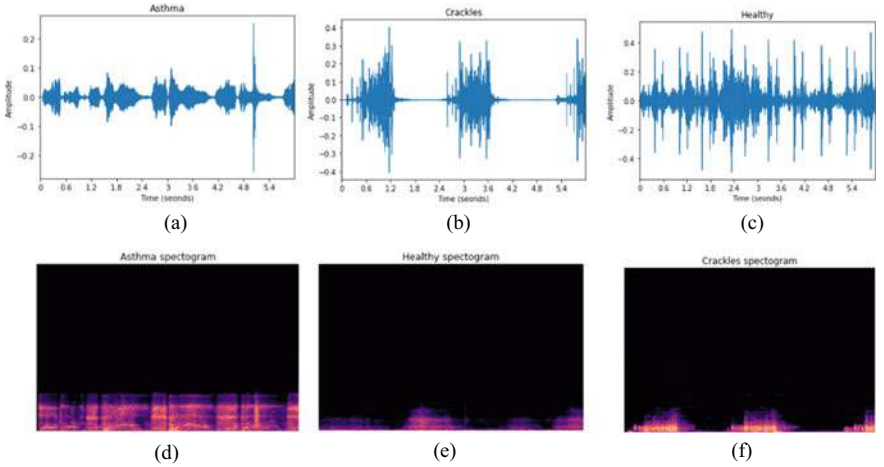


Fig. 2 **a–c** shows amplitude representation, **d–f** spectrogram for asthma, healthy and crackles sound

by adding Noise Gate, De-clicker plugins. We performed augmentation for audio with the help of the Librosa library; Time shifting, Speed tuning, Mix background. Each audio file was divided into six-second intervals. This period can identify Asthma, which is a step performed with the help of the WavePad Sound Editor program. The obtained results after the augmentation and segmentation process are 315 recorded for Asthmatics and 132 healthy records (Fig. 2).

4.2 Feature Extraction

In the collected records, wheeze and crackle signals were characterized by irregularity and asymmetric, that making it tough to describe and identify the key feature from the spectrum. Hence, we have adopted two types of feature extraction methods to convert the audio into an understandable format, to differentiate between the Normal and the Asthmatic, and to make it feedable to the deep-learning model. The extracted features used the numeric data from audio and spectrogram. We used two types of feature extraction to Evaluate that help improve the accuracy of the deep-learning model. We have used 6 types of sound features; we extract them using the Librosa package in python after changing the audio to the mono channel.

1 The chromatogram typically used to represent audio as a vector contains 12 elements [16], indicating the degree of energy of each pitch class, then the standard deviation of the 12 coefficients was applied.
2. Root Mean Square (RMS) [17] is like the amplitude-modulated, but it is more computationally efficient and robust.

3. Spectral Centroid is used to characterize the spectrum [18] that indicates the center of mass location and calculate as the mean of the represented frequency in the sound.
4. Spectral Rolloff Spectral represents the frequency [18] below and defines each frame as the center frequency for a spectrogram bin such the default percent is 0.85for the spectrum's energy.
5. Spectral Bandwidth finds the average deviation of the chart around the Centroid [19], commonly used to define the state of audio at a particular time.
6. Mel-Frequency Cepstral Coefficients (MFCC) is widely used for feature extraction, especially in speech recognition [20]. It was performed in six steps; The pre-emphasis step used higher-frequency sounds to balance the high-frequency spectrum, which had a noticeable roll-off, framing, and windowing step; divided the signal into short frames every 10 M.S. to investigate temporal aspects of audio and efficiently track, 20 M.S. for analysis window to provide perfect spectral resolution of the sound, Compute Fast Fourier transform FFT for every window, Discrete Fourier transform (DFT) spectrum: For each window frame calculates the magnitude spectrum using DFT. The Fourth step was mapping the spectrum's powers that computed the Mel spectrum by passing the Fourier transformed (F.T.) signal by band-pass filters called Mel-filter bank and applying triangular filters, which were usually 40 filters. The Fifth step is to Calculate Discrete Cosine Transform (DCT) after computing the energy levels in neighboring bands must decorrelate these filter banks, the log Mel spectrum was inversed into the spatial domain, saving the amount of MFCC. In This paper, we used the number 20 of MFCC coefficients (Fig. 3).

For the second type of feature extraction from audio as an image(spectrum), Features were extracted in two different styles to better performance of the deep learning model (Fig. 4).

After pre-processing phase for the first model, we have applied K-Fold cross-validation for splitting data into 10 groups that help to eliminate the overfitting problem on training phase (Fig. 5).

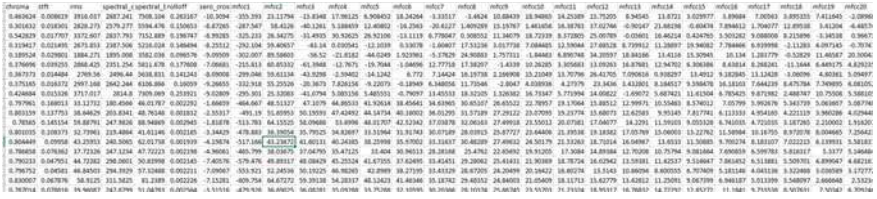


Fig. 3 MFCC extracted features

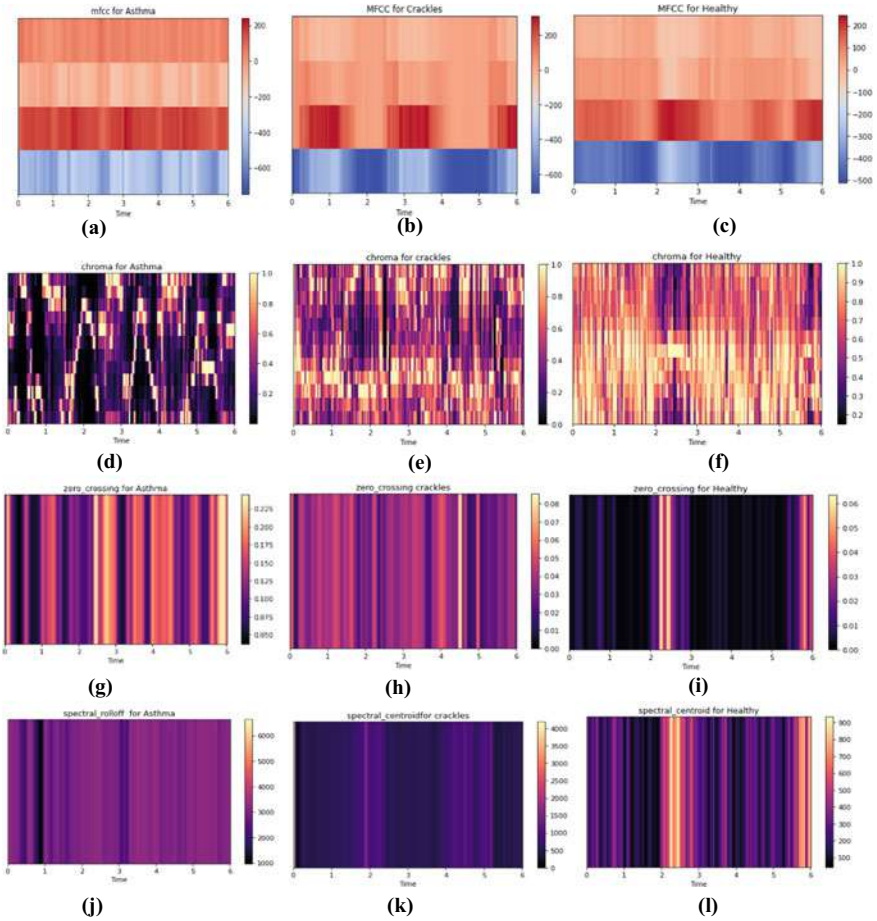


Fig. 4 **a–c** are the representation for MFCC extracted features, **d–f** Chroma extracted features, **g–i** spectrogram for zero-crossing rate (ZCR) features, **j–l** Spectral centroid and spectral_rolloff features

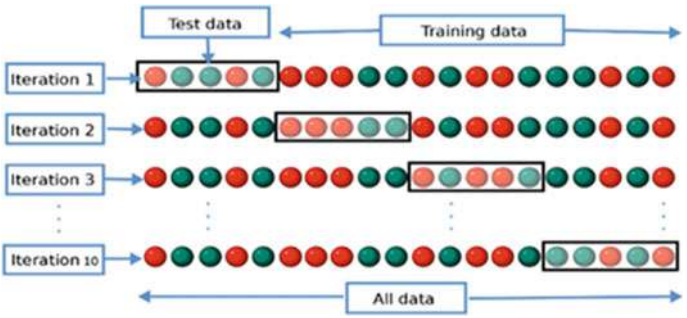


Fig. 5 K-fold splitting

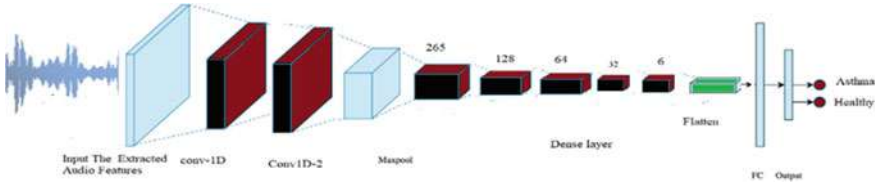


Fig. 6 Show convolutional neural network architecture

5 System Implementation

5.1 Deep Learning Model

The deep learning pipelines model implemented in our classification smart application incorporates the convolutional neural network illustrated in Fig. 5. CNN is a Feedforward neural network capable of processing many structured data, such as images, speech recognition, natural language processing, pattern recognition, and time series [21]. CNN Architecture consists of Input Layer takes the input data that will feed to the neural network and reshape it to the appropriate dimensional, Convo Layer performs convolution operation and apply the filter, Pooling Layer that can be two types max-pooling or average pooling used to reduce the volume of input data, Fully connected layer It connects layers of neurons in one neural pathway to layers of neurons in another path, and the SoftMax layer is the final layer used for binary classification or multi-classification. We implement two models of CNN, the first one on raw features as showing in Fig. 3 and the second on the image (spectrogram) as shown in Fig. 4, for evaluating the performance and select the higher accurate model (Fig. 6).

5.2 Model Deployment

To utilize machine learning and deep learning models in real environments and decision-making [22], the model should be integrated into the existing production environment and receives input and returns the output. The TensorFlow with Keras provide a total solution for the production deployment of machine-learning models. We have built a portable system that supports a web interface and mobile application use Classification pipeline Model to Diagnose whether the sound signal contains Asthma or not. In the first phase, we designed a stethoscope with a mic to get digitized data. In the Second phase using TensorFlow and Keras, we deploy the model as an h5 extension. We used a web framework called Flask and made a web service; we coded it with python to receive an audio file using the post method and the pipelined for features extraction, reshape data, and the trained model. Finally, we designed a web interface and Android application to send the recorded file, receive the classification

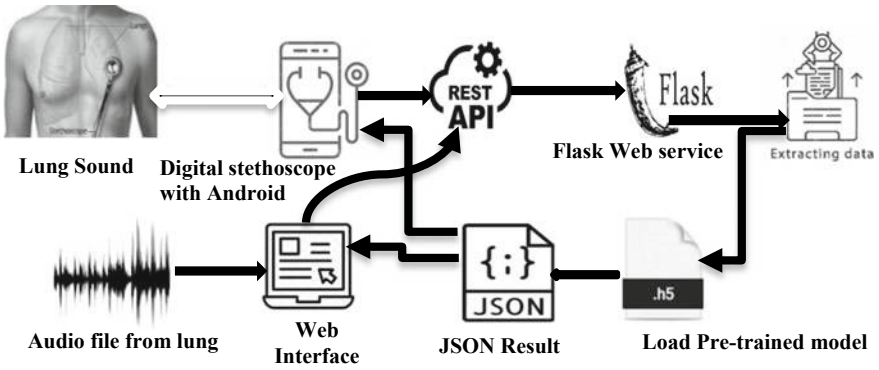


Fig. 7 Deep learning model deployment process and result

result and show it on the interface. Figure 7 provides a high-level representation of the intended system, including the web and application deployment.

6 Results

In our conducted experiment, the audio files collected from various resources [13–15]. We selected records that contain wheezing, and crackles related to Asthma only, to be two classes Asthma and Healthy. We built two models; the first worked on numeric features extracted from audio as mentioned in Fig. 3 whereas the second worked on spectrogram representation as shown Fig. 4. The convolutional neural network (CNN) was used with xgbosst to compare the obtained accuracy using the first model with Adam optimizer as shown in Fig. 8 and Table 1. Hence, we deployed a model that depends on the first model, and we built a web interface and Mobile App, which is a portable system that helps doctors to diagnose Asthma with high accuracy (Fig. 9).

7 Conclusion

Physicians use the stethoscope most frequently to assess lung function based on sound. Methods for processing lung sounds are beneficial if they are simple to use and capable of accurately recognizing additional lung sound classes. Our present work aimed at aiding clinical decision-making, to specifically diagnose a patient as asthmatic or no in a way that helps in overcoming the traditional auscultation with the electronic stethoscope and the Application of AI. A total of 447 asthmatic and Healthy records were used in this paper to train the CNN classification model, based on audio features such as Chromatogram, RMS, Spectral Centroid, Spectral Rolloff,

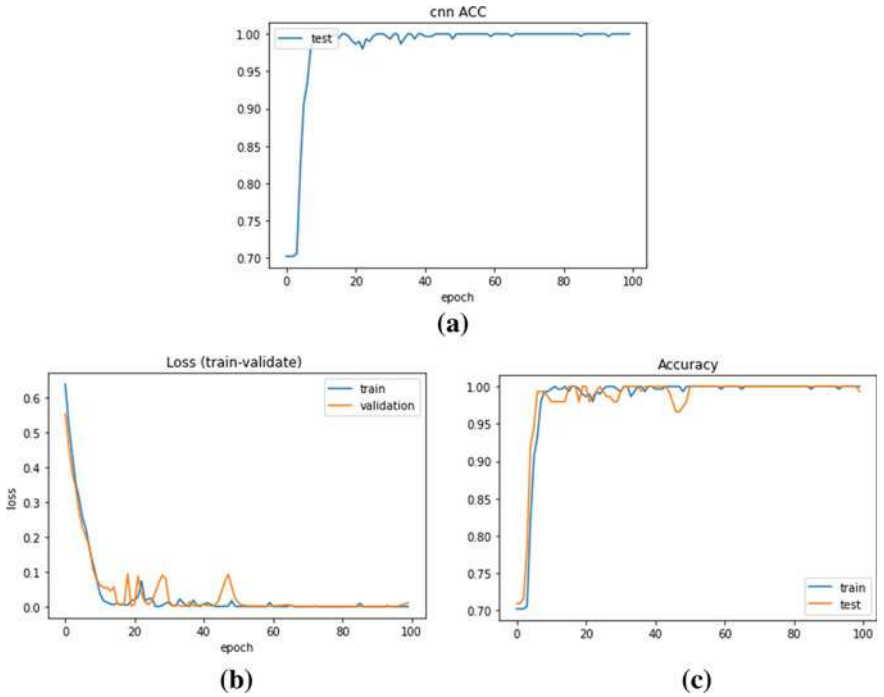
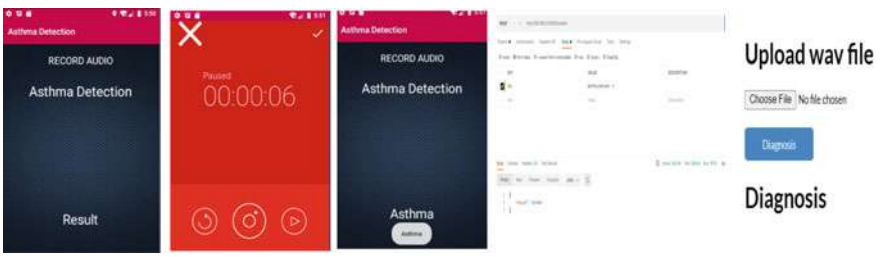


Fig. 8 Model performance **a** Testing accuracy, **b** losing train versus validation, **c** Accuracy comparison between train versus test Model performance

Table 1 Show Performance Evaluation with different optimizers

Model	Optimizer	Accuracy%	Sensitivity%	Specificity%	F-Score%
Raw data model	RMSprop	99.7	100	100	100
	Adam	99.8	99	100	99
	Adagrad	99.6	99	99	99
	Xgbosst Classifier	99.2	99	99	99
Spectrogram model	Adam	97.2	96	95.6	96



F

Fig. 9 Android app result with web interface

and MFCCs, in which the accuracy reached 99.8%. This work was the first deep learning model that can be used in a real environment because we built a web and mobile app to classify asthma in real-time accurately.

References

1. Yernault JC, Lenclud C (1989) Wheezing and asthma. In: Progress in asthma and COPD: proceedings of the symposium "Progress in asthma and COPD". ICS849. Elsevier Science Publishers B.V., pp 61–69
2. Dharmage SC, Perret JL, Custovic A (2019) Epidemiology of asthma in children and adults, /pmc/articles/PMC6591438/
3. Nunes C, Pereira AM, Morais-Almeida M (2017) Asthma costs and social impact. Asthma Res Pract 3
4. O'byrne PM, Postma DS (1999) The many faces of airway inflammation asthma and chronic obstructive pulmonary disease
5. Patel PH, Mirabile VS, Sharma S, Marr C (2021) Wheezing (Nursing). StatPearls Publishing, Treasure Island (FL), Saint Louis University Hospital
6. Rady RM, Akkary IMEl, Haroun AN, Fasseh NAE, Azmy MM (2015) Respiratory wheeze sound analysis using digital signal processing techniques. In: Proceedings—7th international conference computing intelligence communications systems networks, CICSyN, pp 162–165
7. PB G, SV R (2016) Automatic wheeze detection system as symptoms of asthma using spectral power analysis. J Bioeng Biomed Sci 6
8. Aviles-Solis JC, Storvoll I, Vanbelle S, Melbye H (2020) The use of spectrograms improves the classification of wheezes and crackles in an educational setting. Sci Reports 101(10):1–8
9. Mukherjee H, Sreerama P, Dhar A, Obaidullah SM, Roy K, Mahmud M, Santosh KC (2021) Automatic lung health screening using respiratory sounds. J Med Syst 45
10. Nabi FG, Sundaraj K, Lam CK (2019) Identification of asthma severity levels through wheeze sound characterization and classification using integrated power features. Biomed Signal Process Control 52:302–311
11. Senbekov M, Saliev T, Bukeyeva Z, Almabayeva A, Zhanaliyeva M, Aitenova N, Toishibekov Y, Fakhradiyev I (2020) The recent progress and applications of digital technologies in healthcare: a review
12. Bolhasani H, Mohseni M, Rahmani AM (2021) Deep learning applications for IoT in health care: a systematic review. Inform Med Unlocked 23:100550
13. Rocha BM, Filos D, Mendes L, Vogiatzis I, Perantoni E, Kaimakamis E, Natsiavas P, Oliveira A, Jácome C, Marques A, Paiva RP, Chouvarda I, Carvalho P, Maglaveras N (2018) A respiratory sound database for the development of automated classification. In: IFMBE proceedings. Springer, pp 33–37
14. Fraiwan M, Fraiwan L, Khassawneh B, Ibnian A (2021) A dataset of lung sounds recorded from the chest wall using an electronic stethoscope. vol 2
15. Rocha BM, Filos D, Mendes L, Serbes G, Ulukaya S, Kahya YP, Jakovljevic N, Turukalo TL, Vogiatzis IM, Perantoni E, Kaimakamis E, Natsiavas P, Oliveira A, Jácome C, Marques A, Maglaveras N, Pedro Paiva R, Chouvarda I, De Carvalho P (2019) An open access database for the evaluation of respiratory sound classification algorithms. Physiol Meas 40:035001
16. Nepal A, Kumar Shah A (2022) Chroma feature extraction related papers the KUSC classical music dataset for audio key finding. Int J Multimedia Its Appl (IJMA)-ERA Indexed A Robust Mid-Level Represent Harmonic Content in Music Signals
17. Seryasat OR, Aliyari Shoorehdeli M, Honarvar F, Rahmani A (2010) Multi-fault diagnosis of ball bearing using FFT, wavelet energy entropy mean and root mean square (RMS). In: Conference proceedings—IEEE international conference on systems, man and cybernetics, pp 4295–4299

18. Kos M, Kačič Z, Vlaj D (2013) Acoustic classification and segmentation using modified spectral roll-off and variance-based features. Digit Signal Process A Rev J 23:659–674
19. Decker AG, Malthus TJ, Wijnen MM, Seyhan E (1992) The effect of spectral bandwidth and positioning on the spectral signature analysis of inland waters. Remote Sens Environ 41:211–225
20. Appendix A MFCC Features
21. Sun Y, Xue B, Zhang M, Yen GG (2020) Completely automated CNN architecture design based on blocks. IEEE Trans Neural Netw Learn Syst 31:1242–1254
22. Chen Z, Yao H, Lou Y, Cao Y, Liu Y, Wang H, Liu X (2021) An empirical study on deployment faults of deep learning based mobile applications, pp 674–685

Chapter 16
Comparison of Machine Learning Algorithms and Neural Network for Breast Cancer Prediction

Ibraheam Fathail, Vaishali Bhagile, Mohammed Tawfik, Nasser M. Al-Zidi, and Talal A. Aldhaheri

1 Introduction

World Health Organization says that the most prevalent malignancy is breast cancer with more than 2.2 million cases in 2020. Breast cancer affects around 1 in every 12 women at some point in their lives. Breast cancer is the number one cause of cancer death among women, and nearly 685,000 women died from it in 2020. There are significant differences between high-income and low- and middle-income countries, where the 5-year survival rate after breast cancer exceeds 90% in high-income countries, while it does not exceed 66% in India and 40% in South Africa [6]. Symptoms of breast cancer include the appearance of a tumor in the breast, nipple retraction or indentation, change in the size or features of the breast, flattening or indentation of the skin covering the breast, the appearance of redness, or what appears to be wrinkled skin on the surface of the breast, similar to the peel of an orange [7].

Causes of breast cancer: Inheritance only 5–10% of breast cancer cases are due to genetic causes and other genetic defects such as the ataxia–telangiectasia mutation gene and the gene number (P53) responsible for curbing tumors. Breast cancer risk factors include age, exposure to radiation, excess weight, menstruation at a relatively early age, taking birth control pills, smoking, and high density of breast tissue on Mammography [8].

So it is very important to have early detection of breast cancer before reaching high-risk state. Machine learning algorithms are an essential topic in the healthcare business because they may improve disease prediction accuracy and allow for

I. Fathail (✉) · M. Tawfik · T. A. Aldhaheri
Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India
e-mail: abumarina2223@gmail.com

N. M. Al-Zidi
Faculty of Computer Sciences, Albaydha University, Albaydha, Yemen

V. Bhagile
Deogiri Institute of Technology and Management Study, Aurangabad, Maharashtra, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_17

a deeper examination of medical data [9]. In this article, we executed eight supervised machine learning algorithms and neural network from artificial intelligence on the breast cancer dataset, and comparing them, we obtained the good results of all algorithms for classification of the breast cancer.

2 Related Work

Asri et al. [1] employed machine learning methods such as SVM, NB, DT, and k-NN to predict breast cancer and make comparison between the algorithms to the right evaluation for classifying data in terms of accuracy, precision, sensitivity, and specificity. The dataset used in this work is Wisconsin Breast Cancer dataset (WBCD). The result which has been obtained from this classification displays that SVM gained the highest accuracy (97.13%). Gupta et al. [2] applied six machine learning algorithms, which are k-NN, DT, LR, RF, SVM, and radial function kernel, and Adam Gradient Descent from deep learning on breast cancer dataset. The results of comparison show that the deep learning algorithm (Adan Gradient Descent) gives the highest accuracy (98.24%). Sharma et al. [3] executed the popular machine learning algorithms (RF, k-NN, NB, etc.) to predict breast cancer. The dataset used in this work is Wisconsin Diagnosis Breast (WDBD) which contains 33 features with 569 records. The experimental results from this work depending on RF, k-NN, and NB are 94.74, 95.90, and 94.47, respectively.

Shamrat et al. [4] used supervised machine learning algorithms such as SVM, NB, k-NN, RF, DT, and LR for breast cancer prediction. They estimated the classification depending on specificity, f1-score, sensitivity, and precision. The result which has been obtained from this work shows that Support Vector Machine gives the highest accuracy of 97.07%. Li et al. [5] searched about the relationship between the attributes and the breast cancer to reduce the risk of this disease. They used five classification algorithms (LR, DT, SVM, RF, and NN) to classify two datasets (WBCD and BCCD) and compared these algorithms in terms of f1-score and accuracy. The results from experiment of BCCD depending on DT, SVM, RF, LR, and NN are 0.683, 0.763, 0.785, 0.737, and 0.748, respectively, and the results of WBCD depending on DT, SVM, RF, LR, and NN are 0.963, 0.977, 0.989, 0.981, and 0.985, respectively. Chaurasia et al. [11] used the application of data mining methods to the identification and prediction of breast malignant tumors was thoroughly explored in this paper. The techniques used in this work include SVM, k-NN, DT, RF, LDA, and GBC. The result which has been gained is 97.66% from all techniques.

The major goal of Gopal et al. [12] is to look at IoT devices and machine learning techniques for predicting breast cancer. The classifier's minimal error rate was discovered to be 34.21%, 45.82%, and 64.47%, respectively. MAR, RMSE, and RAE are acronyms for Mean Absolute Error, Root Mean Square Error, and Relative Absolute Error. The MLP classifier outperforms the LR and RF classifiers in terms of accuracy and error rate, as demonstrated by the data. Precision, recall, f1-score, and

Table 1 Features description of WBCD

Feature	Description
Diagnosis	Breast tissue diagnostic (M = malignant, B = benign)
radius_mean	The average of the distances between the center and peripheral points
texture_mean	Gray-scale value standard deviation
perimeter_mean	The tumor's average size
smoothness_mean	Local difference in radius lengths mean
compactness_mean concavity_mean	Mean of perimeter^2/area—1.0 The average severity of the concave regions of the contour
Concave points_mean	The average number of components with concave contours

accuracy for the suggested classifier were all 98%, 97%, 96%, and 98%, respectively. Bhusare et al. [13] used and assessed the overall performance of each model and forecasted optimal accuracy in this article machine learning classifier, Nearest neighbor training accuracy, SVM, SVM classifier, Gaussian Naive Bayes, Decision tree, and Random Forest Algorithm. After analyzing the performance of various machine learning models that give higher accuracy on breast cancer datasets, the performance of all different machine learning models on breast cancer datasets may be analyzed. Sample Heading (Forth Level). The contribution should contain no more than four levels of headings. The following Table 1 gives a summary of all heading levels.

3 Methodology

3.1 Dataset (WBCD)

The data utilized in the trials is from the Wisconsin Breast Cancer Dataset (WBCD), which has already been classified as malignant or benign. The dataset is made up of 30 features that were calculated using fine-needle aspiration (FNA) of a breast lump. The majority of cancer datasets are in the form of photographs. The features have four significant digits in their values. There are 569 records in total, 357 of which are benign and 212 of which are malignant. We will explain the important features which effect on breast cancer in Table 1 (Fig. 1).

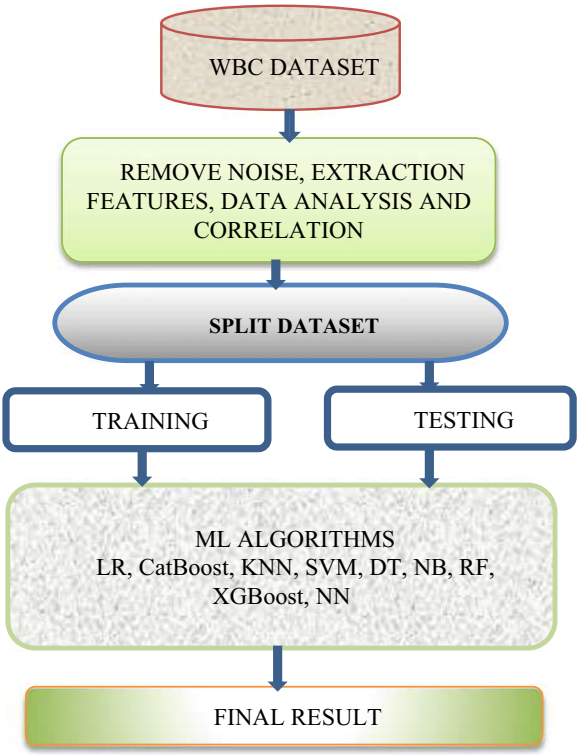


Fig. 1 The proposed method our work

3.2 Remove Noise, Extraction Features, Data Analysis, and Correlation

In this stage, we removed the NaN values from the features, and these noisy value make the algorithms frequently miss trends in the data. Data that is noisy is useless data. Feature Extraction is a method of decreasing the number of features in a dataset by creating new ones from old ones; in our dataset, we dropped (Id, Unnamed) features. Data analysis means analyzing the features and knowing the count, min, mean, std, and max of the variables. Correlation is a measure of the degree to which two variables are connected, and there are three type of correlation:

1. Positive Correlation: When two variables increase and decrease at the same time, they have a nice relationship. The number '1' denotes a perfect positive connection.
2. Negative Correlation: When one variable rises and the other falls at the same time.
3. Zero Correlation: When two variables appear to be unrelated in any way, a perfect negative correlation is a value of zero. The correlation in the WBC dataset is shown in Fig. 2.

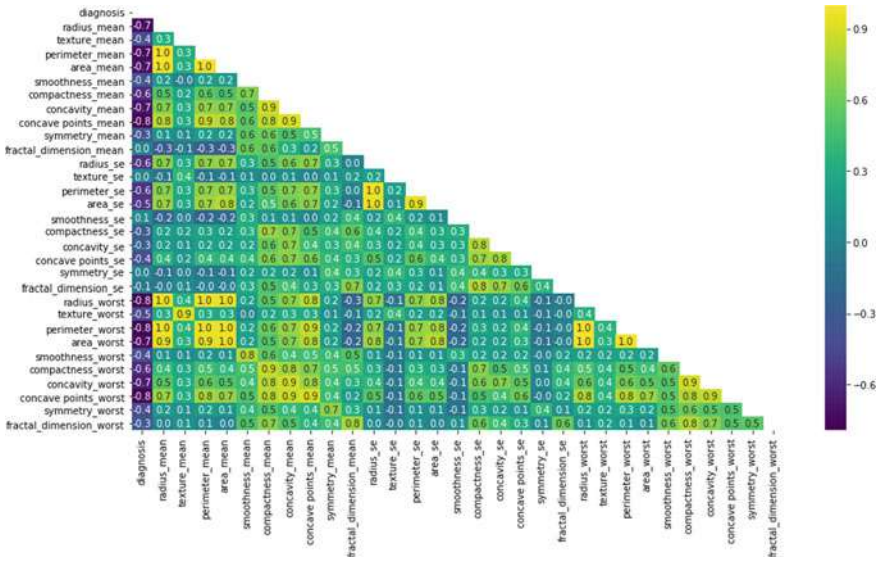


Fig. 2 Correlation between the features of WBCD

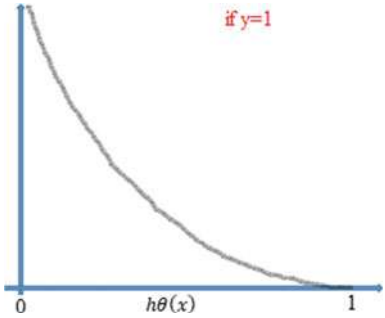
3.3 Split Dataset

Split data to training and testing data: in this stage, we split the dataset into two forms: first form we spliced it depending on test_size = 0.11 and random_state = 0, we obtained the train data 506 records of 30 features and test data 63 records of 30 features, and this divided was useful with some of the classification algorithms which have given us the high accuracy, but some of them like NB, SVM, and NN with this split not give high accuracy like the split which depends on test_size = 0.30 and random_state = 42 which obtained on the train data 398 records of 30 features and test data 171 records of 30 features. However, the results for all algorithms were above 93% in both states.

3.4 Model Fitting

In this part, we used eight machine learning algorithms and neural network technique to classify and predict the breast cancer. The purpose of this research is to determine which model has the best prediction accuracy as the primary model. We compared between the algorithms to know which one gives highest accuracy to select it for classification and prediction breast cancer. In this work, there is no need to explain the classification algorithms, but we will explain or mention about the equation mathematical which used for classification. First of all, the classification means prediction by results of group of data, for example, prediction about the results of

Fig. 3 If $y = 1$



students who is pass or fail in examination depending on the features of student such as the hours of reading and monthly examination results and else the features.

The equation math which is used in the classification. In reality, two equations are used for classification. First, if $y = 1$, use this equation:

$$\text{cost}(h\theta(x), y) = -\log(h\theta(x)) \quad \text{if } y = 1 \tag{1}$$

- The horizontal axis has the value of *h,* and the vertical axis has the value of *J*.
- When the value of *h* decreases to zero, the error value is large, because it is far from the true value 1.
- When the value of *h* approaches 1, the error is small and reaches zero when it is equal to 1, because the real value is zero. Coordinate chart of this state is shown in Fig. 3.

The second state if $y = 0$ used this equation:

$$\text{cost}(h\theta(x), y) = -\log(1 - h\theta(x)) \quad \text{if } y = 0 \tag{2}$$

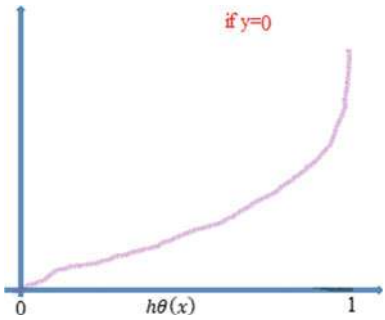
- The horizontal axis has the value of *h* and the vertical axis has the value of *J*.
- When the value of *h* decreases to zero, the error value is small, because it is close to the true value 1.
- When the value of *h* approaches 1, the error is large, because the real value is one; coordinate chart of this state is shown in Fig. 4.

The two equations are grouped together. In both cases, when *y* is zero and one, in the following form:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h\theta\left(x^{(i)}\right)\right)\right] \tag{3}$$

When *y* is equal to zero, the first part of the equation disappears and becomes like this

Fig. 4 If $y = 0$



$$J(\theta) = -\frac{1}{m}\left[\left(1 - y^{(i)}\right)\log\left(1 - h\theta\left(x^{(i)}\right)\right)\right] \tag{4}$$

When y is equal to one, the second part of the equation disappears and is like this:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h\theta\left(x^{(i)}\right)\right] \tag{5}$$

3.5 Experimental Performance

In this stage, we compare the classification algorithms in terms of five metrics:

a. **Confusion matrix:**

The confusion matrix is array used to assess the performance of classification models for a given set of test data. It can only be determined if the true values for the test data are known. The matrix is divided into two dimensions, actual predicted values and actual values as given in Table 2:

- True Positive (TP): Positive was properly predicted by the model, and the actual value matched.
- True Negative (TN): Negative was anticipated by the model, and Negative was predicted by the real or actual value.

Table 2 Confusion matrix

Predicted		
Actual	Positive	Negative
Positive	True positive	False positive
Negative	False negative	True negative

Table 3 Confusion matrix of algorithms

	Benign	Malignant	Class
LR	36 0	2 25	Benign Malignant
CatBoost	37 1	1 24	Benign Malignant
Naive Bayes	24 4	1 34	Benign Malignant
SVM	36 0	2 25	Benign Malignant
K-NN	37 2	1 32	Benign Malignant
DT	35 0	3 25	Benign Malignant
RF	38 0	0 25	Benign Malignant
XGBoost	37 0	1 25	Benign Malignant
NN	37 0	1 25	Benign Malignant

- False Positive (FP): Positive was predicted by the model, but the value was really negative.
- False Negative (FN): The model anticipated a negative outcome, while the actual outcome was a positive one.

The results which has obtained from algorithms for confusion matrix depend on test_size = 0.11 and random_state = 0 divided, and we obtained the train data 506 records of 30 features and test data 63 records of 30 features as given in Table3.

- ***Logistic regression:***

Only two observations are misclassified as benign, and none are misclassified as malignant, according to the confusion matrix of logistic regression, with an accuracy of 96.83%.

- ***CatBoost:***

Only one observation is misclassified as benign, and one is misclassified as malignant, according to CatBoost's confusion matrix, with an accuracy of 96.83%.

- ***Naive Bayes:***

Only one observation is misclassified as benign, and four are misclassified as malignant, according to the Naive Bayes confusion matrix, with an accuracy of 94.15%.

- ***Support Vector Machine:***

The Support Vector Machine confusion matrix seems intriguing. Only two observations are misclassified as benign, while none are misclassified as malignant, resulting in an accuracy of 96.83%.

- ***K-Nearest Neighbors:***

The K-Nearest Neighbor confusion matrix seems intriguing. Only one observation is misclassified as benign, while two are misclassified as malignant, resulting in an accuracy of 95.24%.

- ***Decision Tree:***

The Decision Tree confusion matrix seems intriguing. Only three observations are misclassified as benign, while none are misclassified as malignant, resulting in an accuracy of 95.24%.

- ***Random Forest:***

The Random Forest's confusion matrix seems intriguing. None misclassified as benign, and none erroneously labeled as cancer, resulting in an accuracy of 99.50%.

- ***XGBoost:***

The XGBoost confusion matrix seems intriguing. Only one observation is misclassified as benign, while none are misclassified as malignant, resulting in an accuracy of 98.41%.

- ***Neural Network:***

The Neural Network confusion matrix seems intriguing. Only one observation is misclassified as benign, while none are misclassified as malignant, resulting in an accuracy of 98.41%.

b. **Accuracy**

It is one of the most essential factors in determining how accurate categorization issues are. It indicates how often the model predicts the outcome accurately

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{6}$$

c. **Precision**

It may be defined as the percentage of all positive classes that the model correctly predicted being true or the number of valid outputs provided by the model. The following formula may be used to compute it:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

d. **Recall**

It is the percentage of positive classes out of a total of positive classes that our model correctly predicted. It is important to have the highest possible recall rate.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

e. **F-measure**

We may use this score to evaluate both recall and accuracy at the same time. The F-score is maximal when recall and accuracy are equal. It may be calculated using the following formula:

$$\text{F-score} = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}} \tag{9}$$

The results of our experimental show that the Random Forest gives the highest accuracy, recall, precision, and f1-score (100%, 100%, 100%, and 100%) as given in Table 4.

Figure 5 shows the accuracy measures (Accuracy, Precision, recall and F1 score) for the classification algorithms as mentioned in Table 4.

Table 4 Comparative between classification algorithms for prediction breast cancer

Classification model	LR	CatBoost	NB	SVM	K-NN	DT	RF	XGBoost	NN
Accuracy	96.83	96.83	94.15	96.83	96.49	95.24	99.50	98.41	97.07
Recall	94.74	97.37	96.30	94.74	98.15	92.11	99.50	97.37	97.22
Precision	100	97.37	94.54	100	96.36	100	100	100	98.13
F1-score	97.30	97.37	95.41	97.30	97.25	95.89	99.50	98.67	97.67

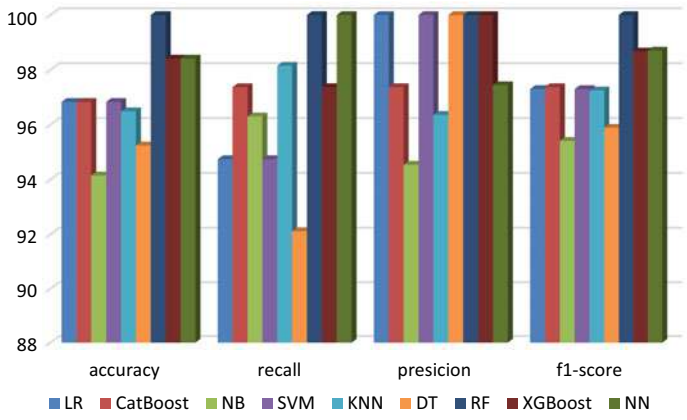


Fig. 5 Shows the performance measures (accuracy, precision, recall, and f1-score)

4 Conclusion

Breast cancer is the most common kind of cancer found in women. A woman selected at random has a twelve percent probability of being diagnosed with the disease [10]. As a result, early diagnosis of breast cancer can save many lives. The workings of nine machine learning models for the Wisconsin Breast Cancer Dataset are presented in this study using their hyper-parameters. Different machine learning techniques have been used to classify cancerous and benign cells. The accuracy found by Random Forest is highest 100%.

References

1. Asri H, Mousannif H, Al Moatassime H, Noel T (2016) Using machine learning algorithms for breast cancer risk prediction and diagnosis. Procedia Comput Sci 83:1064–1069
2. Gupta P, Garg S (2020) Breast cancer prediction using varying parameters of machine learning models. Procedia Comput Sci 171:593–601
3. Sharma S, Aggarwal A, Choudhury T (2018, December) Breast cancer detection using machine learning algorithms. In: 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS). IEEE, pp 114–118
4. Shamrat FJM, Raihan MA, Rahman AS, Mahmud I, Akter R (2020) An analysis on breast disease prediction using machine learning approaches. Int J Sci Technol Res 9(02):2450–2455
5. Li Y, Chen Z (2018) Performance evaluation of machine learning methods for breast cancer prediction. Appl Comput Math 7(4):212–216
6. DeSantis CE, Bray F, Ferlay J, Lortet-Tieulent J, Anderson BO, Jemal A (2015) International variation in female breast cancer incidence and mortality rates. Cancer Epidemiol Prev Biomarkers 24(10):1495–1506
7. Zaki NF, Sabri YM, Farouk O, Abdelfatah A, Spence DW, Bahammam AS, Pandi-Perumal SR (2020) Depressive symptoms, sleep profiles and serum melatonin levels in a sample of breast cancer patients. Nat Sci Sleep 12:135
8. Ilbawi AM, Velazquez-Berumen A (2018) World health organization list of priority medical devices for cancer management to promote universal coverage. Clin Lab Med 38(1):151–160
9. Fathail I, Bhagile VD (2020, October) IoT based machine learning techniques for healthcare applications. In: 2020 international conference on smart innovations in design, environment, management, planning and computing (ICSIDEMPC). IEEE, pp 248–252
10. U.S. Breast Cancer Statistics. [Online]. Available https://www.breastcancer.org/symptoms/understand_bc/statistics
11. Chaurasia V, Pandey MK, Pal S (2021, March) Prediction of presence of breast cancer disease in the patient using machine learning algorithms and SFS. IOP Conf Ser Materials Sci Eng 1099(1):012003. IOP Publishing
12. Gopal VN, Al-Turjman F, Kumar R, Anand L, Rajesh M (2021) Feature selection and classification in breast cancer prediction using IoT and machine learning. Measurement 178:109442
13. Bhusare K, Deshmukh P, Bodkhe B (2021) Breast cancer prediction using multiple models based on machine learning and deep learning

Chapter 17
Exploring Energy Poverty Indicators Through Artificial Neural Networks

Lefkothea Papada and Dimitris Kaliampakos

1 Introduction

1.1 The Problem of Energy Poverty

Energy poverty is one of the current crucial socioeconomic problems, as it deprives people of basic standards of life quality. Access to electricity, as well as to clean, affordable, and modern energy services remains difficult for millions of people living in the developed countries (750 million and 2.5 billion people, respectively) [1]. The notion of energy poverty is differentiated in the developed world, as mainly being related to affordability of energy, i.e., the inability of people to adequately cover their energy needs at an affordable cost. The problem has intensified in Europe, as well as in Greece during the last decade, as a result of the broader financial crisis. More specifically, large parts of the population are unable to maintain an adequate level of heating at home, are on arrears on their energy bills, spend large amounts of money on energy compared to their incomes, reduce drastically energy consumption in the home, or are forced to cut on other basic needs in order to meet their energy needs.

As a multidimensional phenomenon, energy poverty is not approached through a single indicator. On the contrary, different objective and subjective indicators tend to be used by the scientific community in order to measure and analyze the problem within certain geographical areas. The prevalent objective indicator of energy poverty has been the "10% rule", introduced by the United Kingdom [2]. According to this, a household is considered energy poor if, in order to achieve an adequate level of thermal comfort at home, it is required to spend more than 10% of its income on

L. Papada (✉) · D. Kaliampakos
Lab of Mining and Environmental Technology, School of Mining and Metallurgical Engineering, National Technical University of Athens, Zografou Campus 15780, Athens, Greece
e-mail: lefkipap@metal.ntua.gr

D. Kaliampakos
e-mail: dkal@central.ntua.gr

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_18

energy uses, as defined in Eq. 1.

$$\text{Fuel Poverty Ratio} = (\text{Modelled fuel costs(i.e. modelled consumption} \times \text{price})/\text{Income}) > 10\% \quad (1)$$

However, the difficulty of modeling required energy consumption—as suggested by the definition—has forced most researchers toward the use of actual costs instead of required ones in calculations, as data on actual costs are easier to obtain through primary surveys and national statistical surveys. As a result, the "10% rule" with the use of actual energy costs is the one that has informally been widely recognized as the typical, objective indicator of energy poverty in Europe [3, 4]. Based on this indicator, it has been found that 58% of households in Greece are energy poor [5], while the problem is even greater in the mountainous areas of the country due to certain geographical and climatic characteristics, with the energy poverty rate escalating to 73.5% [6].

Many other objective indicators have been also used in literature for measuring energy poverty, such as, indicatively, the indicators 2M and M/2 introduced by the European Observatory of Energy Poverty [7], which measure the ratio of energy costs to income and absolute energy costs, respectively, with respect to the national median numbers. An interesting indicator developed in Papada and Kaliampakos [8] has been the "Degree of Coverage of Energy Needs" (DCEN), which aimed to capture hidden energy poverty. According to this indicator, the compression of energy needs (CEN) was defined according to Eq. 2, considering that when a household actually spends on energy less than 80% of the respective required energy cost, it cannot meet adequately its energy needs.

$$\text{CEN} = (\text{Actual energy cost}/\text{Required energy cost}) < 0.80 \quad (2)$$

In addition to objective indicators, energy poverty is also approached through subjective indicators, in a complementary way. This kind of indicators reflects the subjective perception of energy poverty, based on households' personal assessment of thermal comfort conditions in the home. The most widely used subjective indicators at European level are "inability to keep home adequately warm", "arrears on utility bills", and "dwelling with a leaking roof, damp walls, floors or foundation, or rot in window frames of floor" [9, 10]. Still, other subjective indicators have been also suggested over time, for example "health problems linked to poor heating conditions" and "restriction of other essential needs" [5].

Subjective indicators are considered essential for an integrated approach to the energy poverty problem, as identifying aspects that are not captured by objective indicators [3, 11]. Given that the relevant indicators refer to different objective and subjective characteristics but, even more, to different human behaviors and decisions, it is particularly difficult to reveal the internal relationship between the indicators (objective–subjective) with typical mathematical tools. Relevant research [5, 6] has shown that simple correlations cannot accurately reflect the way indicators

are connected, if being actually connected. In fact, the answer to "how an individual reacts to his/her financial inability to meet his/her energy needs" and "what are the factors affecting this decision" is a product of a complicated decision process, which cannot result from a simple algorithmic relation. Indicatively, it has been reported that the connection between the "10% rule" and subjective indicators is not strong enough [12, 13], but this has been only theoretically inferred so far.

1.2 Introduction to Artificial Neural Networks

The development of artificial neural networks (ANNs) began as an attempt to understand the function of human brain and mimic its evaluation capabilities. In other words, the aim of ANNs is the capability to decide and act under conditions of uncertainty, or even manage situations with limited prior experience [14].

ANNs present a strong resemblance to the biological brain and, hence, a large part of their terminology comes from neuroscience. They are mathematical models that consist of interconnected processing nodes (neurons) with a predefined topology (layers). Their main feature is the capability of mass parallel processing of input data, contrary to the typical mathematical models that are based on a serial execution of mathematical and logical functions [15]. ANNs present a number of advantages over other methods (numerical and non-numerical), which make them important tools in terms of dealing with a variety of problems. Their logic is based on the collection of data in the form of cause–effect (input–output), which are entered into the network in order to be trained. From that moment, the neural network is able to "discover" the existing relationships within the data and "learn" them, with the appropriate adjustment of connection weights [14]. In this way, the network is trained and taught through "experience" and, hence, it can give estimates for uncertain conditions or for incomplete data [16].

Based on the above, with the use of ANNs, it is possible to assess whether there are specific "trends" in the way an individual reacts to the problem of energy poverty, without knowing the relationships between the variables in question. The tool of ANNs has not been generally applied in the field of energy poverty so far, with few recent exceptions. In particular, Rajic et al. [17] suggested a model for forecasting energy demand of Serbian households and, particularly, two consumption variables for electricity and heating (outputs), based on 15 input parameters at macroeconomic level, e.g., population, transport, unemployment, average wage, industry, etc., with the use of a neural network called "endocrine NARX neural network (ENARX)" in MATLAB. Additionally, beyond ANNs but close to this method, Longa et al. [18] used another machine learning classifier and, in particular, a type of decision tree (XGBoost/gradient boosting decision tree), to predict the risk of household energy poverty in the Netherlands based on a range of socioeconomic parameters (inputs): economic value of the house, property status, house age, household size, and average population density. Characteristically, they reported that machine learning could be

used as an effective tool for monitoring energy poverty that could substantially help toward designing and implementing appropriate policy measures.

In the present paper, the use of artificial intelligence and, particularly, the use of ANNs for exploring the relationships between energy poverty indicators, is attempted for the first time, to the best of authors' knowledge. In this context, several models are developed aiming to predict "objective" energy poverty (objective indicators) based on the "subjective" perception of the problem (subjective indicators) for the case of Greece.

2 Materials and Methods

A database was initially set up, composed of data of five distinct primary surveys on energy poverty in Greece. The five surveys consisted of a first one with reference to the whole country (Greece) [5], a second one with reference to the mountainous areas of Greece [6], two surveys with reference to the town of Metsovo, Greece, and a last one with reference to the settlement of Agrafa, Greece. The last two areas were selected as representative case studies of mountainous areas being hardest hit by energy poverty in Greece.

All the above surveys were conducted at household level during the years 2016–2020, with random sampling method, at 95% confidence interval. The database included a total of 1754 series of data, i.e., 400 cases of households living in Greece, 400 households living in mountainous areas of Greece, 643 households living in Metsovo, and 311 households living in Agrafa. From the wide range of data of the above surveys, including housing and living conditions, heating systems, energy costs, incomes, etc., certain objective and subjective indicators were selected and calculated for the current analysis.

More specifically, the analysis included the prediction of objective indicators based on subjective indicators through training a neural network. The three objective indicators selected as output variables were (a) "10%_actual", i.e., based on actual expenses, (b) "10%_required", i.e., based on required expenses, and (c) "CEN", i.e., compression of energy needs. The specific objective indicators were selected as being considered the most decisive ones, according to the authors' opinion. Particularly, the "10% rule" is the most widely used indicator of energy poverty in Europe, capturing the side of energy cost, one of the key components of energy poverty. On the other hand, the "CEN" indicator captures the rather neglected but critical aspect of energy poverty, that of compression of energy needs, according to which a household does not spend a lot on energy, or else, consumes less energy than required to meet energy needs. This last aspect is largely invisible in most common indices; yet, it constitutes a significant side of energy poverty. The five subjective indicators selected as input variables were (a) inability to keep home adequately warm (IW), (b) arrears on energy bills (AB), (c) damp problems/leakages (DL), (d) health problems (HP), and (e) restriction of other essential needs (RN).

With respect to the calculation of the indicators, the subjective indicators were directly obtained by the primary surveys, while the objective indicators needed to be calculated based on specific data of the surveys. More specifically, the indicator 10%_actual was calculated based on data of expenditure and disposable income of households. The calculation of indicators 10%_required and CEN was more complicated and was based on the methodologies developed in Papada and Kaliampakos [8, 19], respectively. For the implementation of the above-mentioned methodologies, certain parameters, i.e., climatic and building variables, had to be calculated per household, namely, heating degree days, cooling degree days, heat transfer coefficient, performance factors of the heating and the cooling systems used.

After setting the datasets, the next basic step was the training of a neural network, aiming at predicting the energy poverty risk as defined by objective indicators, based on different subjective indicators. For this reason, the machine learning tool/interface "WEKA" (Waikato Environment for Knowledge Analysis) was used, an application developed at the Waikato University in New Zealand, which includes a collection of machine learning algorithms and a variety of data preprocessing [20]. The multilayer perceptron, i.e., a neural network that uses back propagation for training, was selected for the current analysis.

On the whole, 93 different basic tests were performed in terms of all different combinations of subjective indicators, i.e., (a) all five subjective indicators with each one of the objective indicators, (b) five different combinations of four subjective indicators with each of the objective indicators, (c) ten different combinations of three subjective indicators with each of the objective indicators, (d) ten different combinations of two subjective indicators with each of the objective indicators, and (e) each subjective indicator with each of the objective indicators. For each distinct model examined, 70% of the dataset was used for training and 30% for testing.

Certain techniques were adopted to improve the performance of the models provided by the neural network. Indicatively, the number of hidden layers and the number of nodes were selected per case, in order to get the best results per distinct model (e.g., in some cases, networks with two hidden layers fitted better than one). In case that classes of the output variable were considerably unbalanced, a filter called synthetic minority oversampling technique (SMOTE) was applied, to avoid any bias in the provided results. SMOTE was used to adapt the relative frequency between majority and minority classes through oversampling the minority class, i.e., creating synthetic instances, with the use of K-nearest neighbor approach. Hence, numerous tests had to be performed per case (per basic test), changing certain elements of the neural network in order to get the best possible performance of the model.

In order to evaluate the performance of the models trained and tested, certain metrics were taken into account, such as the accuracy score, along with basic performance metrics (detailed accuracy by class), i.e., "precision", "recall", "F-measure", and "ROC area". In more detail, "precision" refers to the ratio of correct positive predictions to the total positive (correct and incorrect) predictions of the class, while "recall" refers to the ratio of correct positive predictions to the sum of correct positives and incorrect negatives of the class. "F-measure" expresses the combination of "precision" and "recall" and is used as a general metric representing the costs of

false predictions (false positives and false negatives). "ROC area" (Receiver operator characteristic area under the curve) is another measure of accuracy, indicating the degree of random prediction of the model and is preferred to be the highest possible. In fact, ROC area reveals whether the neural network is really appropriate or not. Moreover, the confusion matrix per model was constructed, indicating the portions of correct and incorrect predictions, i.e., the diagonal elements of each row represent the correctly predicted instances (correct positives and correct negatives) per class category of the output variable. The evaluation of a model's performance depends on a combination of all the above elements.

3 Results and Discussion

Tables 1 and 2 present the performance of the neural network models tested, using as input variables all different combinations of subjective indicators and as output values the three objective indicators under study (10%_actual, 10%_required, and CEN). From the 93 different tests performed—in terms of different combinations of subjective indicators—only the more meaningful ones are presented, i.e., the model predicting the indicator 10%_required is not presented at all, as not providing neither a valuable performance, nor any meaningful model.

3.1 Prediction of Indicator 10%_Actual Based on Subjective Indicators

As regards the prediction of indicator 10%_actual, the accuracy ranges from 55.89 to 56.67% (Table 1). The lowest accuracy (55.89%) is shown in the case of the combination IW-DL-RN as input variables, while the highest one (56.67%) in the case of four subjective indicators: IW-AB-DL-RN. In other words, the use of only three indicators, i.e., inability to keep home adequately warm, damp problems/leakages, and restriction of other essential needs can correctly predict the indicator of 10%_actual at the percentage of 55.89%, while the addition of the indicator arrears on energy bills increases the accuracy of the prediction by just 0.8% points. The further addition of the indicator of health problems to the above model reduces its performance by 0.2% points.

The models using as input variables only the indicator RN, or the following three combinations: IW-RN, AB-RN, and IW-AB-RN, or even all the subjective variables, present the same accuracy (56.51%), with the first four cases presenting the same confusion matrix as well. This shows that the indicator restriction of other essential needs, either alone or combined with various other indicators, can accurately predict the indicator of 10%_actual at the percentage of 56.51%. Hence, the indicator restriction of other essential needs appears to be the key subjective indicator affecting

Table 1 Prediction of indicator 10%_actual and confusion matrices for selected combinations of subjective indicators (test sets presented)

Input variables	Prediction of indicator 10%_actual							
	Precision	Recall	F-measure	ROC area	Class	Accuracy (%)	Confusion matrix (%)	
IW-AB-DL-HP-RN	0.653	0.557	0.601	0.576	Yes	**56.51**	**56**	44
	0.476	0.576	0.522	0.576	No		42	**58**
	0.580	0.565	0.569	0.576	(Weighted avg)			
IW-AB-DL-RN	0.643	0.592	0.617	0.560	Yes	**56.67**	**59**	41
	0.476	0.531	0.502	0.560	No		47	**53**
	0.575	0.567	0.569	0.560	(Weighted avg)			
IW-AB-RN	0.624	0.659	0.641	0.547	Yes	**56.51**	**66**	34
	0.469	0.431	0.449	0.547	No		57	**43**
	0.560	0.565	0.562	0.547	(Weighted avg)			
IW-DL-RN	0.642	0.565	0.601	0.568	Yes	**55.89**	**57**	43
	0.469	0.550	0.506	0.568	No		45	**55**
	0.571	0.559	0.562	0.568	(Weighted avg)			
IW-RN	0.624	0.659	0.641	0.548	Yes	**56.51**	**66**	34
	0.469	0.431	0.449	0.548	No		57	**43**
	0.560	0.565	0.562	0.548	(Weighted avg)			
AB-RN	0.624	0.659	0.641	0.545	Yes	**56.51**	**66**	34
	0.469	0.431	0.449	0.545	No		57	**43**
	0.560	0.565	0.562	0.545	(Weighted avg)			
RN	0.624	0.659	0.641	0.545	Yes	**56.51**	**66**	34
	0.469	0.431	0.449	0.545	No		57	**43**
	0.560	0.565	0.562	0.545	(Weighted avg)			

the output (indicator 10%_actual) in all models tested. In other words, households that tend to restrict other essentials for the sake of heating needs, tend to spend a disproportionally high percentage of their income on energy, hence are considered energy poor both from an objective and a subjective point of view.

Regarding the best result of the above tested models, deriving from the combination IW-AB-DL-RN as input variables, 59% of the energy poor households and 53% of the non-energy poor households are predicted correctly (the diagonal elements of the confusion matrix). Moreover, the F-measure (0.569) and the ROC area (0.560)

Table 2 Prediction of indicator CEN and confusion matrices for selected combinations of subjective indicators (test sets presented)

Input variables	Prediction of indicator CEN							
	Precision	Recall	F-measure	ROC area	Class	Accuracy (%)	Confusion matrix (%)	
IW-AB-DL-HP-RN	0.596	0.564	0.579	0.581	Yes	**55.15**	**56**	44
	0.504	0.536	0.520	0.581	No		46	**54**
	0.554	0.551	0.552	0.581	(Weighted avg)			
IW-AB-DL-RN	0.595	0.511	0.550	0.581	Yes	**54.18**	**51**	49
	0.495	0.579	0.534	0.581	No		42	**58**
	0.550	0.542	0.542	0.581	(Weighted avg)			
IW-AB-DL	0.584	0.507	0.543	0.536	Yes	**53.20**	**51**	49
	0.485	0.562	0.521	0.536	No		44	**56**
	0.539	0.532	0.533	0.536	(Weighted avg)			
IW-AB-RN	0.613	0.606	0.610	0.589	Yes	**57.48**	**61**	39
	0.530	0.536	0.533	0.589	No		46	**54**
	0.575	0.575	0.575	0.589	(Weighted avg)			
IW-DL-RN	0.607	0.553	0.579	0.588	Yes	**55.92**	**55**	45
	0.512	0.567	0.538	0.588	No		43	**57**
	0.564	0.559	0.560	0.588	(Weighted avg)			
IW-RN	0.607	0.663	0.634	0.585	Yes	**58.06**	**66**	34
	0.541	0.481	0.509	0.585	No		52	**48**
	0.577	0.581	0.577	0.585	(Weighted avg)			

for the weighted average exceed 0.5, implying that the examined model is marginally satisfactory.

The neural network of the selected model included one hidden layer, composed of three nodes. The filter SMOTE was also activated to adapt the relative frequency between majority and minority classes, i.e., double the incidents of minority classes, with the use of K-nearest neighbor approach (15 nearest neighbors selected). The above neural network is illustrated in Fig. 1.

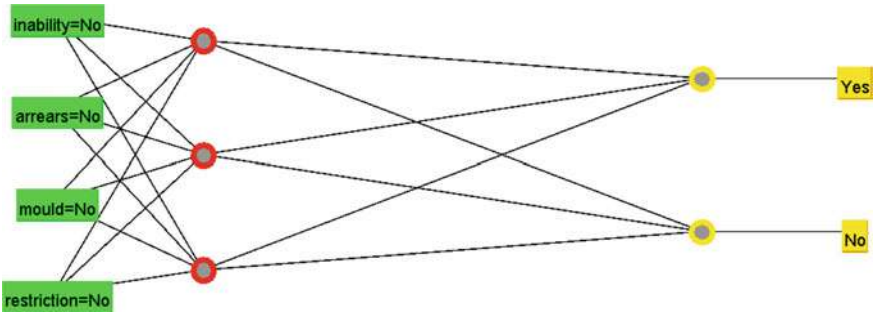


Fig. 1 Neural network of the model with the best performance, predicting indicator 10%_actual (IW-AB-DL-RN: input values—one hidden layer)

3.2 Prediction of Indicator CEN Based on Subjective Indicators

Regarding the prediction of the indicator CEN, the accuracy ranges from 53.20 to 58.06% (Table 2). The lowest accuracy (53.20%) is shown in the case of the combination IW-AB-DL as input variables, while the highest one (58.06%) in the case of IW-RN. Hence, it appears that the use of only two indicators, i.e., inability to keep home adequately warm and restriction of other essential needs can accurately predict the indicator CEN at the percentage of 58.06%, while the addition of any other indicator reduces the accuracy of the prediction.

In the best result of the tested models (IW-RN as input variables), 66% of the energy poor households are predicted correctly, but only 48% of the non-energy poor households are predicted correctly. For this reason, the best model can be considered the one of the combination IW-AB-RN as input values, which, although presenting an accuracy slightly lower than the previous one (57.48%), the prediction of both energy poor and non-energy poor households exceeds 50% (61% and 54%, respectively). The values of F-measure and ROC area for the weighted average (0.575 and 0.589, respectively) exceed 0.5 for both cases, implying that the model is marginally satisfactory. In any case, the indicator inability to keep home adequately warm appears to be the key subjective indicator affecting the output (indicator CEN) in all models tested, i.e., the neural network of the selected model included two hidden layers, composed of four and three nodes, respectively (Fig. 2).

4 Conclusions

The present paper explores the relationships between energy poverty indicators in Greece with the use of artificial intelligence and, particularly, with the use of artificial neural networks (ANNs), an endeavor attempted for the first time, to the best of

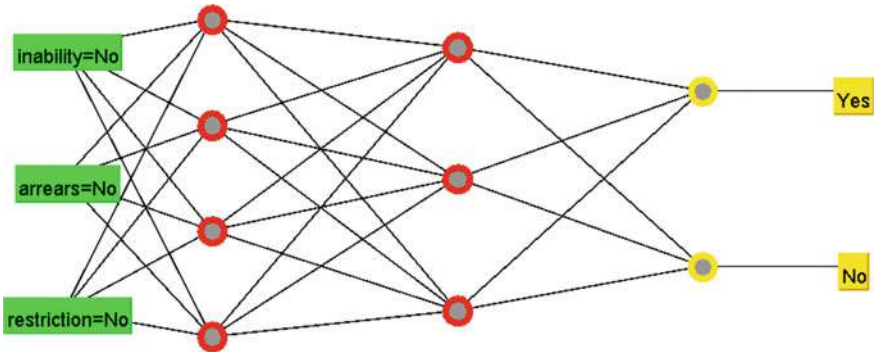


Fig. 2 Neural network of the model with the best performance, predicting indicator CEN (IW-AB-RN: input variables—two hidden layers)

authors' knowledge. A machine learning tool was selected, as it has been shown that conventional mathematical/statistical tools fail to take into account the complexity of different human responses to the problem. On the contrary, neural networks, after proper "training", can reliably simulate the way the human brain works, giving valuable information about the nature of the phenomenon and the relationship between the different indicators that constitute the problem.

The analysis included the prediction of objective energy poverty indicators based on subjective indicators, through training a neural network, i.e., the multilayer perceptron. Three objective indicators were selected as output variables, i.e., 10%_actual, 10%_required, and CEN, while five different subjective indicators were selected as input variables.

The analysis showed that the use of certain subjective indicators—or their possible combinations—is marginally informative for accurately predicting the indicators 10%_actual and CEN, while is not informative at all for predicting the indicator 10%_required. It appears that the relationship between objective and subjective indicators is not really strong; yet, a prediction of energy poverty can be achieved to a certain degree, i.e., in the order of 56–58%. More precisely, from the variety of human behaviors and responses, the human response of "restricting other essential needs in order to meet heating needs" proves to be the key parameter of predicting energy poverty based on the widespread indicator of 10%_actual, while the human condition of "living in an inadequately heated home" proves to be the key parameter reflecting energy poverty based on the CEN indicator. Even further, the combination IW-AB-DL-RN (as input variables) can accurately predict the indicator of 10%_actual at the percentage of 56.51%, while the combination IW-AB-RN (as input variables) can accurately predict the indicator of CEN at the percentage of 57.48%, both of them providing valuable models.

These findings, even if not strong relationships come up on the examined cases, are valuable outcomes for the energy poverty analysis, as quantitatively supporting

previous theoretical assessments of relationships between energy poverty indicators, with the use of an innovative approach, i.e., neural networks. The best possible assessment of indicators, both objective and subjective, as well as the analysis of their possible relationships, can lead to a deeper understanding of the phenomenon of energy poverty and, hence, to more targeted policy measures. Future research may focus on an even deeper analysis of the problem through neural networks, by predicting energy poverty indicators based on socioeconomic factors and by expanding the research to further case studies. Artificial intelligence is expected to be an important tool in understanding energy poverty and, therefore, in adopting effective strategies for tackling energy poverty issues.

Acknowledgments This research is co-financed by Greece and the European union (European Social Fund-ESF) through the operational program «Human Resources Development, Education and Lifelong Learning» in the context of the project "Reinforcement of Postdoctoral Researchers—2nd Cycle" (MIS-5033021), implemented by the State Scholarships Foundation (ΙΚΥ).

References

1. WEO (World Energy Outlook), SDG7: data and projections. https://www.iea.org/reports/sdg7-data-and-projections. Last accessed 2020/07/04
2. DECC (Dept. for Energy and Climate Change) (2015) Annual fuel poverty statistics report, 2015. DECC, London
3. Roberts D, Vera-Toscano E, Phimister E (2015) Fuel poverty in the UK: is there a difference between rural and urban areas? Energ Policy 87:216–223
4. EC (European Commission) (2010) An energy policy for customers. Commission Staff Working Paper. European Commission, Brussels
5. Papada L, Kaliampakos D (2016) Measuring energy poverty in Greece. Energ Policy 94:157–165
6. Papada L, Kaliampakos D (2017) Energy poverty in Greek mountainous areas: a comparative study. J Mountain Sci 14(6):1229–1240
7. EPOV (European Energy Poverty Observatory), Indicators and data. https://www.energypoverty.eu/indicators-data. Last accessed 2021/09/09
8. Papada L, Kaliampakos D (2020) Being forced to skimp on energy needs: a new look at energy poverty in Greece. Energ Res Soc Sci 64:101450
9. Eurostat, EU statistics on income and living conditions (EU-SILC) methodology. Economic strain. https://ec.europa.eu/eurostat/web/income-and-living-conditions/data/database. Last accessed 2021/09/08
10. Eurostat, EU statistics on income and living conditions (EU-SILC) methodology. Housing deprivation. https://ec.europa.eu/eurostat/web/income-and-living-conditions/data/database. Last accessed 2021/09/08

11. Price CW, Brazier K, Wang W (2012) Objective and subjective measures of fuel poverty. Energ Policy 49:33–39
12. DECC (Dept. for Energy and Climate Change) (2009) Annual report on fuel poverty statistics. DECC, London
13. Price CW, Brazier K, Pham K, Mathieu L, Wang W (2007) Identifying fuel poverty using objective and subjective measures. Centre for Competition. Policy Working Paper 07–11. University of East Anglia, Norwich
14. Benardos A (2008) Artificial intelligence in underground development: a study of TBM performance. Underground spaces. WIT Trans Built Environ 102:21–32
15. Fausett L (1994) Fundamentals of neural networks. In: Architectures, algorithms and applications. Prentice Hall International Editions, Hoboken, New Jersey
16. Sietsma J, Dow JF (1991) Creating artificial neural networks that generalize. Neural Netw 4:67–79
17. Rajić MN, Milovanović MB, Antić DS, Maksimović RM, Milosavljević PM, Pavlović DL (2020) Analyzing energy poverty using intelligent approach. Energ Environ 0958305X2090708
18. Longa FD, Sweerts B, van der Zwaan B (2021) Exploring the complex origins of energy poverty in the Netherlands with machine learning. Energ Policy 156:112373
19. Papada L, Kaliampakos D (2018) A stochastic model for energy poverty analysis. Energ Policy 116:153–164
20. Frank E, Hall MA, Witten IH (2016) The WEKA workbench. In: Data mining. Practical machine learning tools and techniques, 4th ed. Morgan Kaufmann, Waikato, New Zealand

Chapter 18
Implementation of 32-bit ISA Five-Stage Pipeline RISC-V Processor Core

Manjunath Kalmath, Akshay Kulkarni, Saroja V. Siddamal, and Jayashree Mallidu

1 Introduction

Choice of instruction set architecture tells the type of instruction the processor can handle. Instruction set architecture influences the implementation technology and application. ISAs such as MIPS, ARM, and ×86 have contributed in development of various computing devices ranging from mobile devices to super computers. But these ISAs hold proprietary, and one has to pay for those companies for using ISAs. New ISA, RISC-V from UC Berkeley, has now attracted most of the companies toward it. RISC-V [1] is an open-source instruction set architecture (ISA) and anyone can use it for free. RISC-V allows the architects or hardware engineers to use the ISA as per their application need. For example, in developing the basic embedded systems one may require base integer instruction set which is enough to support operating system. In developing the systems to perform cutting-edge applications, one may require to add the different extensions along with base integer instructions. RISC-V with only 32-bit base integer instruction set is called as RV32I. RV32G constitutes the base integer instruction set—I, integer multiplication and division extension—M, atomic instructions extension—A, and single precision floating—point extension—F.

M. Kalmath · A. Kulkarni · S. V. Siddamal (✉) · J. Mallidu
KLE Technological University, Hubli, India
e-mail: sarojavs@kletech.ac.in

J. Mallidu
e-mail: jayashree.mallidu@kletech.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_19

2 Literature Survey

Single cycle RISC-V CPU core [2] developed has datapath and control path units. Datapath consists of ALU, register file, decode logic, sign extension, shift, and shuffle logic. Control path consists of memory control logic, fetch, decode control logic, and different control lines for ALU. Control unit accepts opcode of 7-bit, funct3 of 3 bits, and 30th bit of the instruction. Based on this, different control lines are activated. This is a tiny processor can used only for simple embedded applications. This processor is not pipelined so it cannot meet the demands of cutting-edge applications, where-speed is a parameter. In paper [3], pipelined RISC-V core supporting base integer instruction set is discussed. This core can also be added with integer multiplication and division extension, instruction cache, data cache, and branch prediction unit. VexRiscv is implemented using SpinalHDL, and from this, RTL is generated which is a Verilog—HDL file. It is difficult to understand the implementation code and difficult to debug if the generated RTL raises some error. In paper [4], optimized RISC-V soft processor RVCorePb with five-stage pipeline is implemented. It uses RISC-V base integer instruction set, RV32I. It consists of various optimization units to enhance the operating frequency such as ALU optimization unit, instruction fetch unit optimization, and sign-extension optimization unit and has a branch prediction unit. Core is implemented using Verilog—HDL. This core shows better performance when compared to VexRiscv. In [5], a three-stage pipelined RISC-V core called RSIC-V-mini is discussed. It supports RISC-V base integer instruction set. RISC-V-mini is implemented using chisel, hardware design language. It also consists of instruction cache and data cache. In [6], a six-stage pipelined processor supporting base integer instruction set of RISC-V is discussed. It is implemented in system Verilog. Instruction fetch unit is of two stages tohold on-chip memory with a fixed one cycle latency. Jumps are calculated in execute stage, thus giving three-cycle penalty for any taken branch. In [7], five different RV32IM cores are designed for educational use. Sodor core can be used as component within rocket chip SOC generator. These cores cannot be integrated with external memory RV32I. It consists of various optimization units to enhance the operating frequency such as ALU optimization unit, instruction fetch unit optimization, and sign-extension optimization unit and has a branch prediction unit. Core is implemented using Verilog—HDL. This core shows better performance when compared to VexRiscv.

The rest of the paper is organized as follows. Section 3 discusses microarchitectural of five stages. In Sect. 4, pipeline hazards and its remedies are explained. Section 5 discusses the results and finally the conclusion.

3 Micro-architecture of Five-Stage Pipeline

A single core, 32-bit, five-stage pipeline without-of-order integer computation processor is proposed. The RISC-V ISA sets are selected for implementation,

comprising base integer RV32I. The details of these ISAs are taken for UC Berkeley green sheet. R-format, I-format, S-format, SB-format, U-format, and UJ-format are six instruction formats of RISC-V used. The design of RISC-V processor has two parts: datapath which consists of the hardware elements required for the execution of the operations. It performs the functionality that is decided by the control path. Control path which is the main part of the processor which decides what each piece of datapath should essentially do. It makes the decision based on the input instruction.

Datapath Design

Since there are six different formats in the base RISC-V 32I, the datapath should be designed in such a way that there is a harmony among all these different formats, and all use up the essential hardware correctly as per the requirement. These have been classified into different formats based on their behavior. The slight modifications need to be done in the entire datapath to support each of the formats. Initially, to begin with the implementation of datapath, the datapath for each type of instruction format is built, so that we can build up the entire datapath and also ensure that the new instruction format added does notaffect the operation of the already existing instruction formats. R-type instruction format holds the basic building blocks of the datapath, hence to begin with the implementation of the datapath, R-type instruction format was implemented. To simplify the building of datapath for the R-type instruction format, consider the following example, sub $\times 0$, $\times 2$, $\times 3$. To have this working, the processor needs to execute the instruction in a highly organized manner. This entire procedure can be split up into following five steps.

- Get the instruction and check for its functionality. In this case, it is the subtraction operation, sub $\times 0$, $\times 2$, $\times 3$.
- Later the instruction is parsed into different fields (rd, rs1, rs2, and operation) rd $= \times 0$, rs1 $= \times 2$, rs2 $= \times 3$.
- Read the data based on parsed operands R[$\times 2$] R[$\times 3$].
- Later the fields such as funct3, funct7, and opcode are used to determine the operation to be performed and operation is executed. R[$\times 2$] $-$ R[$\times 3$]
- The result obtained after subtracting is written back to the destination register, $\times 0$. R[$\times 0$] $=$ R[$\times 2$] $-$ R[$\times 3$].

Now in order to perform all these steps meticulously, the hardware resources required was realized and integrated as shown in Fig. 1.

With the slight modifications in the above datapath, the processor will be able to implement both R-type and arithmetic I-type instruction formats. In order to implement the load, I-type instruction formats, there is a necessity of the hardware memory component. Hence, an additional piece of data memory blocks, namely DMEM along with a 2:1 multiplexer, is added for this purpose. It uses a control signal, namely MemRW, to decide whether to read or write data to DMEM block. Hence, it takes an address and outputs the value to the MUX. This MUX is essentially used to determine whether the processor should execute arithmetic I-type instruction or load I-type instruction. The select line for this MUX is write back select. It is used to determine whether the data from ALU is to be written back to the register file or

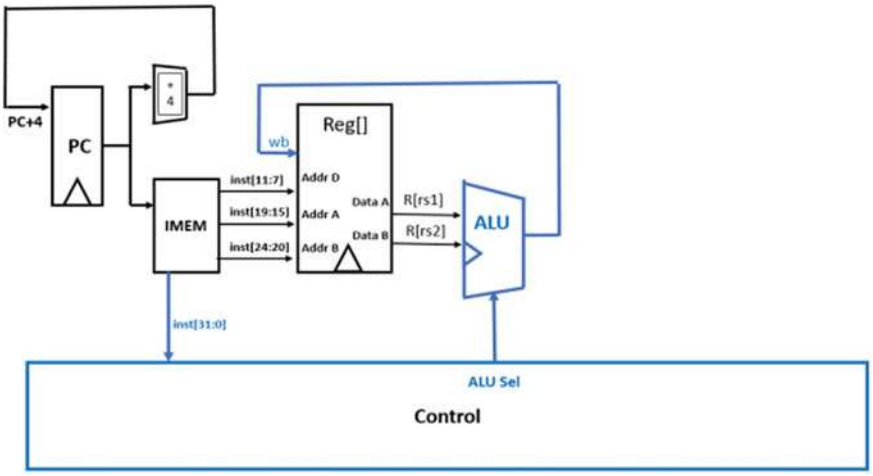


Fig. 1 Hardware resources to execute R-type instruction

the data obtained from the DMEM is to be written back. DMEM is used to parse the instruction fields which help in determining the type of load operation to be performed. Funct3 encodes the size of the data to be loaded. It also helps in determining the type of load operation. If the Most Significant Bit (MSB) of the funct3 is set to 0, it indicates the processor to perform the signed load operation, else if it is set to 1, it indicates the processor to perform the unsigned load operation.

Working on the other data formats the final micro-architecture is designed as shown in Fig. 2. The figure gives details on implementation of datapath and control path of the RISC-V core.

Datapath consists of instruction fetch unit along with instruction memory, immediate generator, register file, branch comparator, ALU, and data memory. Control unit generates the respective control signals based on opcode, funct3, and 30th bit of instruction. Signals, namely PCSel, responsible for generating the correct PC value indicating whether it is of branch instruction or other type of instruction. RegWEn is enabled for register file to write back the value to the register. BSel responsible for choosing the R- type or I-type. WBSel is responsible for writing the computed value from data memory or ALU to the register. MemRW is enabled, which is responsible for reading or writing from the data memory. Instruction memory and data memory are of 4 kB in size. Initially, the machine equivalent of the program is stored in the instruction memory. Instruction fetch will happen in the first stage. For the instructions other than conditional and unconditional branches, program counter gets incremented by 4. For the conditional and unconditional branches, the program counter gets updated to target address. In the instruction decode stage, destination register, source register, and operand fields will be known from the instruction fetched. In case of immediate type of instruction, ImmSel control signal will be activated. Further the instruction execute stage specified operation takes place based on the opcode,

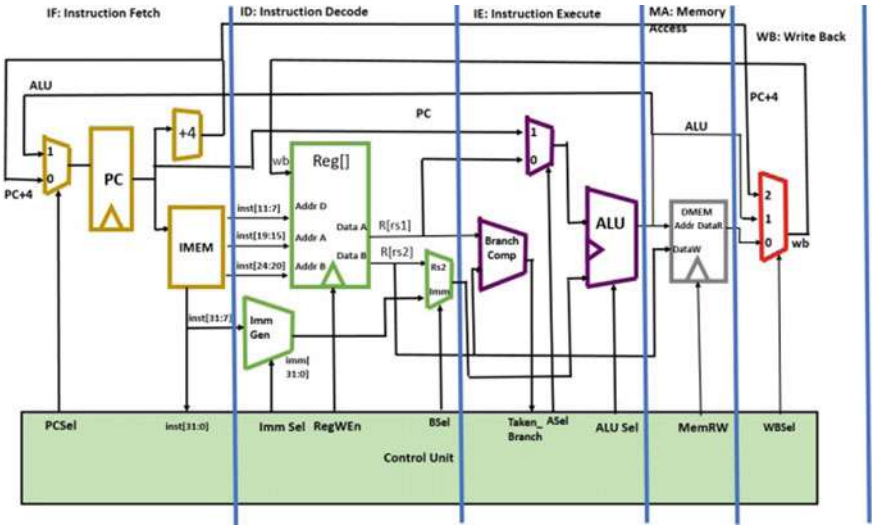


Fig. 2 Five-stage pipelined RV32I CPU core

funct3, and 30th bit of instruction fetched. ALUSel control signal is also generated. In the memory stage, the computed value is written to the data memory. Value can also be read from the data memory. In the write back stage based on the WBSel, the computed value from the data memory or result from ALU will be written to the destination register. There are 32 registers in the register file each 32-bit in size. Branch comparator generates taken branch signal in case of branch instructions occurrence. ALU performs all arithmetic and logical operations.

4 Pipeline Hazards and Its Removal

The major contribution in this paper is the hazard removal. Hazard is an undesirable situation that prevents the execution of the next instruction in next clock cycle. It can also be termed as the condition that causes the instructions to wait for their execution. These hazards are categorized into three types.

- Structural hazards.
- Data hazards.
- Control hazards.

Structural hazards are caused when the required resource is busy. Hence, it occurs when two instructions try to access the resource simultaneously. The resources at which this conflict may arise are ALU, memory, and register file. To avoid this conflict of resources, the possible solution is stalling. In order to overcome the structural hazard, the execution of instruction is delayed and is made to wait to execute. In

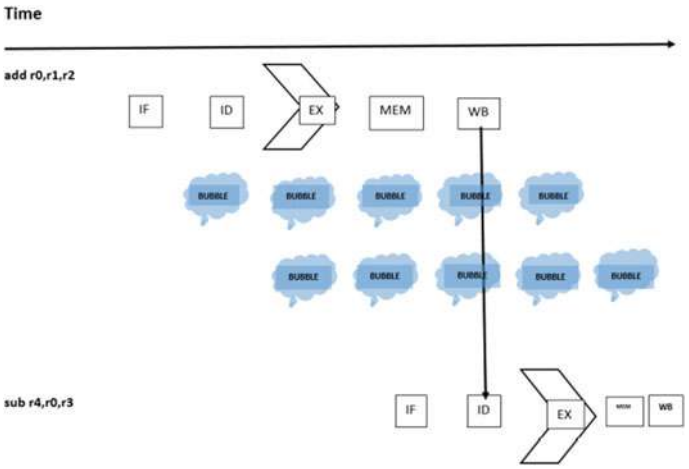


Fig. 3 Stalling to resolve data hazard

designing pipelined RISC-V core, we mainly faced with read after write hazard and control hazard. In case of read after write hazard, the next instruction is trying to access the previous instruction computed result, but the result is not yet computed, so next instruction will access the invalid result. In order to prevent this, we added the stalls in the pipeline. Stall in case of RISC-V is condition of No operation. ADDI X0, X0, 0 is the instruction which indicates No operation and basically it adding zero to zero register. Figure 3 shows the example of data hazard occurrence. R3 register holds the result of addition of R2 and R1. Load instruction following the add instruction uses the R3 register value which is not computed yet so we have added the stalls for two clock cycles and continued after R3 result is made available. Control hazards deal with the control instructions specifically with the branch instructions. The result of branch instructions will be got after execute stage but at this time already the other two instructions waiting in the queue for execution will have entered into the fetch and decode state also. And suppose if the branch wants to jump to fourth instruction, then whether the second and third instructions that are loaded into the pipeline should be determined whether to be executed or not. In this case, the control hazard occurs.

Hence, the fetching of the next instruction depends on the outcome of the branch, but this outcome would be ready only after the execute stage of the first instruction. The solution to overcome the control hazards is stalling. The instruction fetch stage of the second instruction is stalled for two clock cycles as the outcome of the branch would be available in the execute stage at the end of the third clock cycle (Fig. 4).

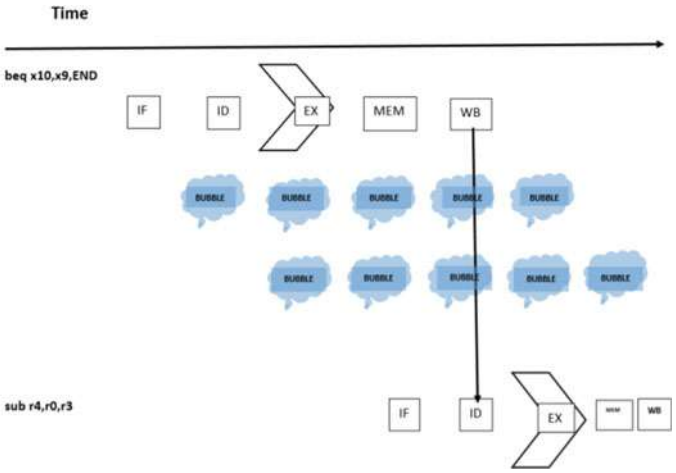


Fig. 4 Stalling to resolve control hazard

5 Results and Discussion

Verification of Designed RISC-V Core

Designed RISC-V core is verified by storing the machine equivalent code in the instruction memory. Results are observed in the Verilog simulations about the instruction type, whether branch is taken or not, values stored in the data memory, register contents, ALU output, status of pipeline hazards, and the status of different control signals. In order to check the implementation of each instruction, simple program is written to find the sum of first ten numbers as shown in Fig. 5. This program uses R-type, I-type, S-type, L-type, and B-type instructions.

FPGA Implementation of Designed RISC-V Core

The designed RISC-V core is implemented of Artix-7 100 T FPGA board. The results of first 10 numbers starting are stored in the memory location. The respective memory location value is displayed using LEDs (both User LEDs and RGB LEDs). Since final result is 45,only 6-bits are enough to represent this value so RGB LED0, RGB LED1, User LED 0, User LED 1, User LED 2, and User LED 3 are used. Btn0 is used as Debounce switch. Table 1 gives the core utilization report which has a greater impact on power utilization. Operating frequency—138 MHz total power consumed is 124 mW.

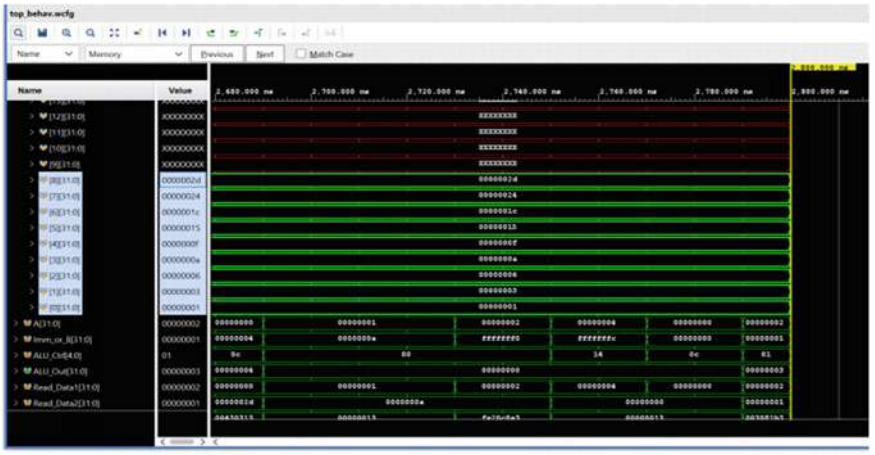


Fig. 5 Simulation result of example

Table 1 Utilization of core

Resources	Utilization	Available	% Utilization
LUT	769	63,400	1.21
LUTRAM	49	19,000	0.26
FF	839	126,800	0.66
IO	14	210	6.67
BUFG	2	32	6.25

6 Conclusion

Designed RISC-V core (RV32I) supports all the instructions mentioned in the RISC-V RV32I base instruction set. Data hazard and control hazard of RISC-V pipeline are handled using stall method. For overall testing of the RISC V core, program written calculates the sum of first 10 numbers starting from 0 to 9. The program uses R-type, I-type, S-type, and B-type instructions. Verification of RISC-V core is done using Verilog simulation. The core is implemented on Artix- - - 7 100T target board.

References

1. Waterman A, Lee Y, Patterson DA, Asanovic K (2014) The RISC V instruction set manual, volume I: user-level ISA, version 2.1
2. Dennis DK, Priyam A, Virk SS, Agrawal S, Sharma T, Mondal A, Ray KC (2017) Single cycle RISC-V micro architecture processor and its FPGA prototype. In: Seventh international symposium on embedded computing and system design (ISED), NIT Durgapur, West Bengal, India, 18–20 December 2017

3. SpinalHDL VexRiscv: a FPGA friendly 32 bit RISC-V CPU implementation. [Online]. Available https://github.com/SpinalHDL/VexRiscv
4. Miyazakim H, Kanamori T, Islam MA, Kise K RV-CoreP: an optimized RISC-V soft processor of five-stage pipelining. arXiv:2002.03568v1 [cs.AR], February 2020
5. University of California, Berkeley riscv-mini: simple RISC-V 3-stage pipeline in Chisel. [Online]. Available https://github.com/ucbbar/riscv-mini
6. University of Cambridge Clarvi: simple RISC-V processor for teaching. [Online]. Available https://github.com/ucam-comparch/clarvi
7. University of California, Berkeley The Sodor processor: educational microarchitectures for RISC-V ISA. [Online]. Available https://github.com/ucbbar/riscv-sodor

Chapter 19
Adaptive Access Control Model Using Risk-Based Technique in Dynamic Environment

Smita Athanere and Ramesh Thakur

1 Introduction

Security is main concern in current communication system. But, achieving security is most tedious task. It just likes a bad dream. One of the solutions for solving this issue is building a convenient effective and efficient access control system. This system will permit access authorized users for system resource access and allow only registered users for authorization. Need of present communication system is as follows: It should be dynamic as well as heterogeneous in nature, so managing with the privacy and security issues is difficult in these systems.

There are basically two types of access control systems [1] are in use: static and dynamic. Static method computes its decision on the basis of rigid and predetermined factors. Static method cannot provide accurate and reliable solution. In dynamic method access, decision is taken on the basis of some contextual and real-time information for granting access to system resources.

The decision of access control is taken on the basis of security risk value. Estimation of the security risk value is done by this access control system involves an analysis on each request of access. This estimated value of risk is compared with access policies to take an access decision. This model can provide solution to such unpredicted issues where any access policy violation takes place. These systems are capable to handle any exceptional request to access resources like when a granting of request takes place which is initiated by an unauthorized user to perform some activity also known as breaking the glass situation. The important issue resolved by this risk-based access control system is flexible resource access. This paper proposes a dynamic risk-based access control model for IoT communication system. This access

S. Athanere (✉)
Computer Engineering, IET, DAVV, Indore, MP, India
e-mail: smita.athanere@gmail.com

R. Thakur
International Institute of Professional Studies, DAVV, Indore, MP, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_20

system is managing the access request of different users to resources by performing aggregation and quantification of risk metrics made by some risk policies by resource owner based on contextual and real-time information of that access request.

1.1 Motivation

As we know, having CIA—confidentiality, integrity, and availability is the important issue in any type of communication system. In current scenarios, if technique used for access control is not flexible, then they results in same decision in variant situation corresponding to an access request. So, in response to this not adaptive, it is highly needed to develop such access control method which considers different environmental factors so that system can take valid decision of permission in normal as well as in unanticipated situation too. We tried to propose a novel architecture for access control system in which it is tried to collect all related environmental context.

1.2 Objective

In this research, we have shown a novel access control scheme for IoT communication system which is heterogeneous in nature. The suggested framework consists of algorithm which collects related environment factors. The following are the work's most significant contributions:

(i) Collection of all dynamic traits associated with an access request need to be done to avoid static nature of any access control model.
(ii) Estimation of risk need to be done by applying proper risk estimation method.
(iii) Comparison of calculated risk against set of access policy needs to be done.
(iv) Validation of final value of risk against threshold to take decision of grant and deny.
(v) Designing and implementation of proper architecture of proposed method.

1.3 Contribution

We proposed a novel architecture for access control system in which it is tried to collect all related context associated with an access request. We proposed to collect like criticality of resource, history of user, severity of risk, magnitude of impact, sensitivity of data, and context of user. After estimating the risk value, it is checked against access policies. If final value is less than a decided threshold value, then only, decision of grant and deny is taken.

1.4 Organization

The related work is explained in Sect. 2. In Sect. 3, it is about IoT architecture. Section 4 describes security issues in IoT. Section 5 is about risk-based access control in IoT. Section 6 describes proposed model and its architecture; Sect. 7 discusses the methods for risk assessment. In Sect. 8, conclusion is given.

2 Related Work

Atlam et al. [1] presented a systematic literature review and analysis of the risk-based access control model to provide an understanding for the risk-based access control. Out of 1044 articles, 44 articles were selected for a minor examination in terms of recent risk-based models on the basis of some selected search strategy, risk factors, and methods for risk estimation. The result gives a basic view of various in a summarized form showing perspective of different researchers. In different articles, there are several risk factor are suggested in various context like 18 articles adopted risk history, and there results show that this dynamic model basically used context or domain. In addition to this, it provided and also demonstrated an accurate and efficient risk estimation technique like decision tree which can be implemented in different domain. Lack of dataset related to a particular context scenario is the major issue.

Santos et al. [2] presented risk-based access control architecture for cloud computing which is dynamic, with cloud federation's application. This model is based on XACML (Extensible Access Control Markup Language) which is providing flexibility for sharing resource and information in a dynamic environment like cloud in which scalability and distribution feature holds. In this, some policies are considered based on some risk metrics considered on the basis of user and different providers.

Atlam et al. [3] proposed a paper in which they stated about access control model for IoT environment. They build some criteria to permit or deny access request for resources. They had shown overview of some common access policies and also discussed access models and their access policies. On the basis of reviews, they proposed XACML, most compatible policy for IoT platform which is scalable, flexible, and distributed in nature. In addition to this, they also proposed adaptive risk-based access control with XACML(Extensible Access Control Markup Language).

Metoui et al. [4] proposed a novel access control approach which enhances the privacy and also offering fine grained adaptive access control for sensitive datasets. It also enhanced the more accessing of data and also reduces the internally generated privacy threats. In this paper, author presented the impact of authorized actor threats and its relevance. They implemented a privacy threat identification methodology evaluating privacy violation risk in cybersecurity systems. This also worked on

privacy-aware risk-based access control framework, and it is integrated with RBAC—role-based access control to achieve futures like scalability, flexibility, and usability. They checked it against many use case to meet industry requirements.

Rahmati et al. [5] proposed a paper a smart homes model for risk-based permission in which they described about smart homes like they seek permission as smart phone permission. Operations are grouped and created permission model over these groups. Because of risk symmetry and over privilege of different apps, it dramatically enhances the exploitation of system since usable apps are malicious in nature. These issues motivated them to introduce risk-based permission for group of devices. They designed Tyche by app rewriting method to get risk-based permission. They performed a study on domain expert and user-perceived risks. On the basis of this study, all operations are regrouped into three classes like high risk, medium risk, and low risk. They deigned it for 146 operations and 61 groups. Also, they had done evaluation on three smart apps available and proved reduction of access of high-risk operation by 60%.

Choi et al. [6] proposed a paper in which they presented a framework for risk-based access control in medical information systems using context-sensitive information for which their motivation was to create a risk assessment framework to enhance security in healthcare information internal and external threats like patients' sensitive data disclosure, any modification, or unauthorized access happens in existing system. In first part, they described about WBAC access control model. In second part, they told different levels of users and their related security levels. For calculation of user trust level, they suggested for rewards and penalty points. The main objective of their work is to design risk assessment framework to evaluate access request risks and related risk value and access decision.

Clark et al. [7] presented a paper named risk-based access control with uncertain and time-dependent sensitivity in which they have shown how to process uncertain and time variant security specification integrating with fuzzy-MLs model. It also addressed difficulties that old system was experiencing in dynamic and unpredictable environment. They used probability distribution and also extended this concept to contextual information.

Chen et al. [8] presented a paper; in this paper, author described about inherent uncertainty in access control system which leads to unpredictable risks that should be addressed and quantified in separate way. The quantification of resource makes that resource countable. So, allocate the resource to achieve goal of optimal utilization of risk. They also worked on Bell–LaPadula multilevel security. Their resultant access control system is just like fuzzy logic control system.

Walters et al. [9] proposed and implemented an access control model taking issue adaptive risk using fuzzy logic with expert judgment. They discussed the access control model for IoT since IoT has capability to connect billions of devices to communicate on wireless or wired media for performing daily life activities. There are several advantages of IoT but also faces many security issues. They describe the rigid and static model is not suitable for required security level for IoT, so they suggested a dynamic and adaptive one. Proposed model collects attributes at the time of access request, sensitivity of data, depth of severity of action to be performed, and

history of estimated risk value for each and every access request. Activities of users are monitored throughout session. They used fuzzy logic system along with expert judgment for risk estimation. They also tried to collect ranges for fuzzy set and fuzzy rules for updating risk estimation process.

3 IoT-Internet of Things

Internet of Things (IoT) communication is important from a social, technical, and economic point of view. Items which are durable, vehicles like cars bus, consumable goods, component usable in industries, sensors, and other objects used in daily life can be included in Internet connectivity. These devices can be considered for powerful data analytic capabilities. Hence, IoT is changing our lives. In Fig. 1 showing popularity graph of IoT communication [10]. Key factors of IoT include:

Transformation Abilities: As far as IoT connectivity is concern, we can say it is turning the whole world into a "hyper-connected world" as shown in Fig. 2. In reality, impacts of trends and projections of shifting thinking for IoT objects usage actively or passively.

IoT: The term IoT defines the scene where the capability of computers and connectivity of network applied to objects, sensors, and everyday usable items like remotes, cars, microwave, and fans. This technology permits these devices to interact with each other and consume.

Models for Connectivity: IoT can be implemented by any of models in a flexible way. Like some active models of communications are node to node, node to gateway, node to cloud, and data sharing.

Support for Technologies: When we combining sensors, computers, and networks to control and monitor devices, this initiates IoT. In combination with other grown

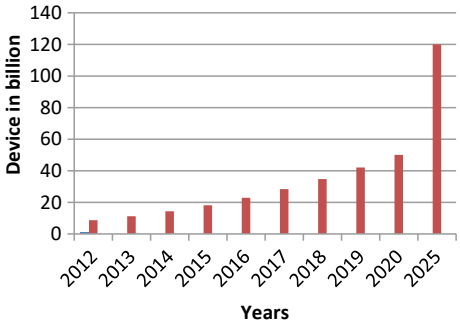


Fig. 1 IoT growth

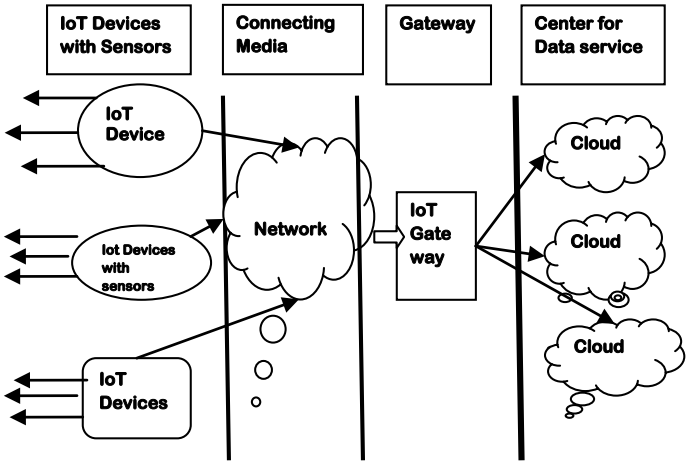


Fig. 2 IoT architecture

technology, Internet of Things has become a widespread and popular reality like IP-based networking, data analytic advances, cloud computing enhances, and ubiquitous connectivity.

Achieving security means approaching confidentiality, integrity, and authorization. In this method, we are considering attacks inner as well as outer attacks from the network. Figure 2 is showing the architecture of IoT [10]. Dynamic nature of IoT communication needs an access control system which is flexible and scalable.

4 Security Factors in IoT

There are many security issues in IoT [11] which are needed to be considered are mentioned below.

Weak password: Password which can be easily guessable can be break by brute-force attack easily grant access in unauthorized manner.

Insecure update mechanisms: The mechanism which is available for updating the devices is insecure techniques for secure delivery, anti-rollback, and lack of notification of any changes.

Insecure interfaces: The insecure Web, surrounding mobile interfaces, backend APIs, and cloud are responsible for weak encryption, lack of authorization and authentication, weak input data filtering, and output data filtering.

Insecure network: For communication among the different IoT devices, Internet services are required which in turn compromises the CIA, i.e., confidentiality, integrity, and authentication of an information and access remotely in unauthorized

way. Now a days, people are using digital locks for their cars, bikes, home, but this is again a combination of software and hardware, so attackers can make unauthorized access [12].

Insecure components: Usage of third-party components compromises the whole system because these components customize the operating system.

Inadequate privacy protection: Attacker can get all the credentials and necessary information needful for authentication and authorization if there are insecure devices or components. They raise vulnerabilities for attackers for injecting malicious code to grant insecure access [13].

Insecure data transfer and storage: Since all the devices involved are communicating with other devices, so data transfer of data between them and components involved can be insecure, or storage can be insecure. Attackers can use much malware to access IoT devices in an unauthorized ways like Trojan, worms, virus, and any spyware which can monitor users all activities [12].

Poor device management: Poor security support for devices deployed in production, including management of assets, updates, systems monitoring, and response capabilities, and lack of administration.

Insecure default settings: Lack of mechanism to make more secure by restricting or modifying the configuration. The default setting can make insecure the system can result in spoofing attack which in turn responsible like spam mails, denial-of-service attack, phishing attack, fraud clicks like on advertisements.

Lack of physical hardening: Measures of physical hardening are lacking and permit attackers to access crucial and sensitive information. This can result in sniffing types of attacks. In sniffing interception of communication path among IoT devices over the Internet (Wi-Fi) is done to get data like passwords credit card or some important mails [14, 15].

5 Access Control in IoT

Main aim of any access control model is used for providing the restriction on users who authenticated successfully. This system ensures confidentiality and integrity of system resources by granting the proper permission to different authorized users [16, 17]. Risk can be due to threat or any vulnerable cause resulting into a damage or loss. Security risk we can say any leakage of information which can breach the system, occurred because illegal access. In risk access control model, amount of risk is the main parameter to provide the access to the system. It depends on formulation of access request for permitting or denying, some contextual parameters collected and analyzed to calculate risk, i.e., amount of damage. In exceptional situation like medical and military, it can provide efficient solution to problem since insufficient information.

For granting access to resource, user sends an access request. This access request is processed and analyzed. The module is responsible for estimation of risk; calculate the risk involved that request by comparing with the request policies. If risk is greater than a predefined value, then by accepting request permission is granted otherwise not [18]. Risk can be described as a mixture of the probability of harm occurrence and its severity. Risk estimation methods are as follows:

a. ***Qualitative—least preferred one***
b. ***Semi Quantitative***
c. ***Quantitative—highly recommendable.***

Qualitative: Scenario where we can apply qualitative methods is where unavailability of quantifiable data or credibility of available data is more. In these cases risk estimation is advised to use an n*m matrix to categorize the risks into different levels like low, medium, and high. Dependability of this method is on how detailed description of probability is given, and related severity is mentioned supplied by analyst based on repetition and rating of consistency at different time instances.

Semi-quantitative: This method is similar to qualitative method with the difference that quantifiable data are available. This method is most useful where lot of data is collected. This method includes scales for probability of occurrences of harm that are different for each and every product like harm per hour, harm per use, or harm per device. Estimate risk on the available quantifiable data.

Quantitative: This is most preferable method and based on concept of quantification of risk indicates that risk $= p_1 * p_2$ where p_1 is probability of occurrence of a hazardous situation and p_2 is occurrence of harm. Need to consider in this method most hazardous situation which can distort the risk management. It calculates risk in five different categories considering harm severity like catastrophic, critical, serious, minor, and negligible. It consider whole spectrum of severities of harm and identifies the highest risk. Table 1 is mentioning comparison.

Table 1 Comparison between qualitative and quantitative

Qualitative	Quantitative
Risk level	Project level
Easy and quick	Takes more time
Subjective evaluation in terms of probability and impact	Probabilistic estimates in terms of time and cost
No requirements of special software or tools required	There can be requirement of specialized tools

6 Proposed Model

Our proposed model is depending on risk estimation-based method. We are proposing a model to secure very important and sensitive data by an efficient and secure access control based on risk. IoT is a kind of heterogeneous device interconnection, so access policies which are based on static criterion are not so adoptive and flexible, so in proposed system, dynamic features are considered to make an access decision which is more real times. So, involved risk in a access request is most dynamic criterion if we are able to estimate the risk involved in a user request, then it becomes very easy to make a decision whether grant permission or deny.

In this model, we are going to consider quantitative method of risk calculation where we are going to consider traits like criticality of resource, history of user, severity of risk, magnitude of impact, sensitivity of data, and context of user. Figure 3 is depicting this.

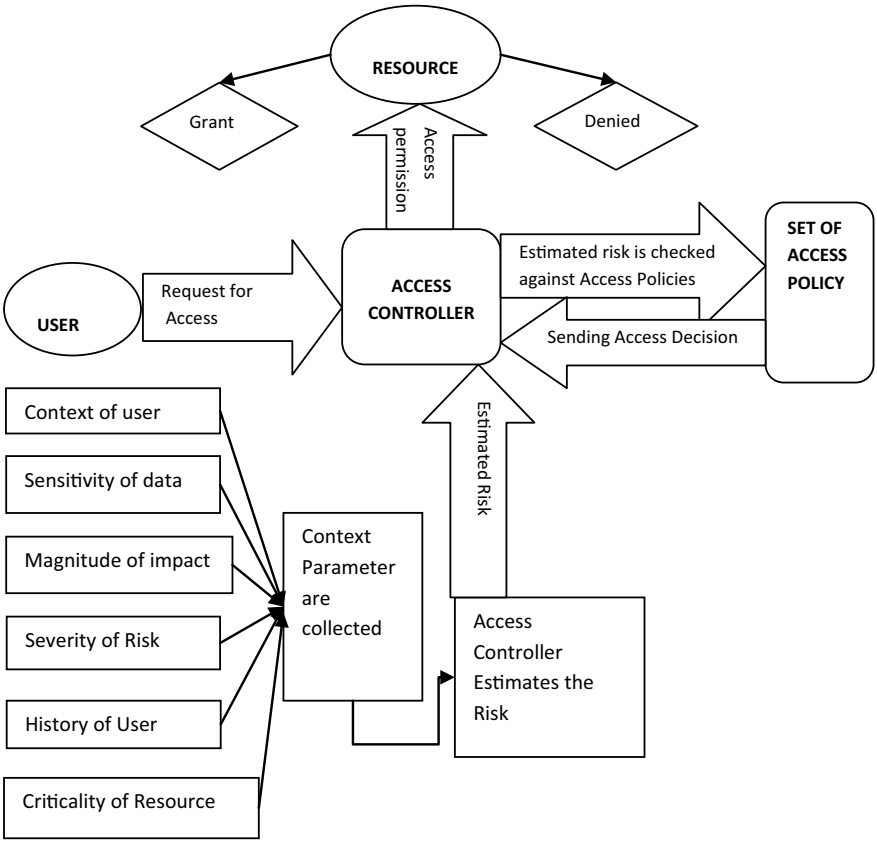


Fig. 3 Proposed risk-based model

6.1 Validation of Proposed Work

In this model, we used significant features of all methods mentioned in related work. After collection of all factors like criticality of resource, history of user, severity of risk, magnitude of impact, sensitivity of data and context of user, and a quantitative risk estimation technique, i.e., based on decision tree applied to collected information and calculate a value of risk. This estimated risk is cross-validated against set of access policies to produce final value of risk which should be less than threshold, so proposed technique would be satisfactory performed current state-of-the-art methods.

7 Methods for Risk Assessment

There are many real-time challenges in risk estimation process faces. The objective behind estimation process of risk is to estimate or predict the later possibility of leakage of information that can come from the current access of the system. To determine such cases is not an easy task. Following are the risk estimation techniques.

a. ***Fuzzy Logic***
b. ***Risk Assessment***
c. ***Game Theory***
d. ***Decision Tree***
e. ***Monte Carlo Simulation***
f. ***Expert Judgment***

Fuzzy Logic: Fuzzy logic is a computation approach for the knowledge representation which is based on membership degrees and also imitates how people think. It is also represent degrees of truth. It also takes care of features like human experiences, common sense, and intuition. Fuzzy logic and fuzzy set operations allow extracting fuzzy set of likelihood and their related severity consequences, the mathematical rule to get it club with expert knowledge [19, 22].

Fuzzy logics are used to create the MLS, i.e., access control based on multilevel security which enables human to get information from IBM machines. This MLS system estimates risk of any access request by seeing difference between security levels of subject and object. If difference value is higher, then it shows highest risk, and if difference value is lower, then it shows the lowest risk. This fuzzy model depicts results by an interval [0, 1] where zero represents absolute permit and 1 represents absolute deny. Further, it is divided into subintervals for getting risk bands.

The methods of fuzzy logic estimation is consist of three steps

a. ***Fuzzification***
b. ***Reasoning using fuzzy rules***
c. ***Defuzzification of outputs.***

By fuzzification, we can get a linguistic term for given input variable. Membership function can be described as by expert or can be taken from data, analytically. Experts

design fuzzy rules, and we can get a crisp number from output fuzzy variables by defuzzification.

Risk Assessment: This is the main method of risk management. This method is used to study potential threat or risk involved in any IT-based scenario. Main goal of assessment is to find out risk context and acceptable risk value in each scenario by comparing similar risks. This method can be applicable to scenario where data availability is more and severity of potential damage. By this, well described assessment of risks can avail a proper base for loss prevention and also threat prevention. There is three different approaches suggested by [20, 21] which can assess risk based on object sensitivity, subject trustworthiness. This approach uses context of application or on priority of application. In this method, risk assessment calculates the risk value formed on outcomes of actions in term of CIA (confidentiality, integrity, and availability) and environment, user's context, and resource. Threshold value plays a role in this method.

Game Theory: It is a part of applied mathematics that can be applicable in many areas like political science, artificial intelligence, economic, evolutionary biology, and information security. Game theory is having four parts like players, payoffs, their procedures, and the data they are having. In this, strategic decision makers are players in relation with the game. Players make strategies and use them against the opponents move. So becomes very important to find out appropriate strategy. The reaction of an assigned player is affected by the action taken by him and other player [24, 25]. Analysis of risk is computed on the basis of priority or values of benefit provided by users instead of probability. This risk analysis technique is based on values or preferences of advantages which the subjects can supply instead of subjective probability provided by game theory.

Decision Tree: Decision trees are a supervised learning method based on non-parametric concepts used in classification and regression. The main objective of this method is to design a model that estimate or predict the values for target variable with the help of simple decision rules extracted from features of data. A tree can be considered as piecewise constant approximation. For developing a model for decision tree, whole data are partitioned into different training and validation sets. Identification of suitable rules is done by training data to get best partition for attributes by using recursive partitioning; on the other hand, validation of decision tree is done by validation data [26]. Decision tree method is very simple to understand and interpret since tree can be visualized easily. It requires less data preparation. The data using are in logarithm in the number of data points means for prediction of data. This method can handle both numerical as well as categorical data. It can handle multi-output issues [25]. It is based on white box model and use Boolean logic. Validation can be done by statically testing. Decision tree also having some limitations like learner can create over complex tree can become unstable because of some data variations; sometimes, prediction cannot be smooth not continuous, or biased tree can be possible.

Monte Carlo Simulation: This is very dynamic and robust method. In this model, prediction of random variable is very tough known as probability of outcomes and capable to calculate the risk of more than thousands of scenario. This technique can be used to understand the uncertainty in estimation or forecasting the impact of risk. This can be applicable to supply chain, finance, engineering, and science. This is also known as multi-probability simulation. Monte Carlo simulation works on assignment of multiple values to a variable which is uncertain to compute multiple results and then finding average of all results to obtain a particular value known as an estimate [27]. Monte Carlo named after gambling destination which in Monaco because chance and random outcome are key points of this technique happens in games like slot, roulette, and dice.

Expert Judgment: This is the very important method for risk estimation. This method performs risk estimation of the processes which rely on quantitative risk assessment having numerical data like conditional probabilities and event frequencies in risk model [28]. By using this method, one can specify correct value for a particular scenario where traditional method cannot. A weight is assigned by expert to potential damage for a specific security threat involved with any risk factor. Selection of expert is done from different domains for getting different points of view. To get more proper results, more experts are needed. Table 2 is mentioning pros and cons of all techniques. Differences are given in Table 3.

Table 2 Pros and cons of different techniques available for risk estimation

Features	Fuzzy logic	Risk assessment	Game theory	Decision tree	Monte Carlo simulation	Expert judgment
Usable	Yes		Yes			Yes
Fast		Yes		Yes	Yes	Yes
Scalable		Yes			Yes	
Dynamic	Yes		Yes	Yes	Yes	
Include expert experience	Yes			Yes		Yes
Enormous resources needed			Yes	Yes	Yes	
Time overhead	Yes	Yes		Yes		
Subjective	Yes	Yes	Yes	Yes		Yes

Table 3 Comparative chart of different risk estimation techniques

Technique of risk estimation	Application domain	Risk factors	Limitations
Fuzzy logic [19, 22]	Online health monitoring system	Difference between security level of object an subject Data sensitivity, severity of action, and history of user risk	Time overhead of fuzzy inference is high and lacked adaptive features
Risk assessment [20, 21]	Numeric problem A case study of a hospital	Sensitivity of object, subject trust, history points of reward and penalty, outcomes of actions	User risk history has not been used and lack of adaptive. Limited risk factors, no risk prediction technique is used and lacked Adaptive features
Game theory [23, 24]	Scenarios of online bookstore	Access benefits of the subject	Limited risk factors and lacked adaptive features
Decision tree [25, 26]	Classification and regression	Piecewise constant approximation	Over complex tree unstable
Monte Carlo simulation [27]	Gambling like slot, roulette, and dice	Probability of outcomes of actions	Chance and random variables
Expert judgment [28]	Numeric problem	Conditional probabilities and event frequencies	Selection of expert

8 Conclusion

As we know, IoT has become very ubiquitous in nature because it is providing unlimited capabilities which can make our routine life smoother. This can have capabilities to connect billions of devices. These devices also have properties to sense there environment context. So, in this highly connected and dynamic network, privacy becomes prime concern. IoT provides us unlimited benefits but simultaneously raise the issues of security. Available access control models are static and rigid. They are based on predefined rules and results into more or less same results not able to provide a level of security. For IoT environment, we need flexible and dynamic access control model which is based on current context of access request. In this paper, a system model is presented for risk-based access control; in this model, IoT devices sense the traits of user access request like criticality of resource, history of user, severity of risk, magnitude of impact, sensitivity of data, and context of user. Values of these collected traits are provided to the risk estimation module of access controller. Since we are having data, we can use any of quantitative method mentioned in Sect. 6 which can be—fuzzy logic, Monte Carlo simulation, expert judgment, decision tree, game theory, and risk assessment. After getting risk value associated with access request, it is verified against access policies and a threshold value. If policy test is cleared and

risk value is less than threshold, then only, access request is granted otherwise denied; we can extend these proposed approach to hierarchical infrastructure network which can also be achieved by risk-based decision-making.

References

1. Atlam HF, Azad MA, Alassafi MO, Alshdadi AA, Alenezi A (2020) Risk-based access control model: a systematic literature review. Future Internet 1–13. www.mdpi.com/journal/futureinternet
2. Dos Santos DR, Westphall CM, Westphall CB (2014) A dynamic risk-based access control architecture for cloud computing. In: IEEE/IFIP NOMS 2014—IEEE/IFIP network operations and management symposium management a software defined world, pp 1–9
3. Atlam HF, Alassafi MO, Alenezi A, Walters RJ, Wills (2018) XACML for building access control policies in Internet of Things. In: Proceedings of the 3rd international conference on internet of things, big data and security (IoTBDS 2018), Madeira, Portugal, May 2018
4. Metoui N (2018) Privacy-aware risk-based access control systems. Ph.D. thesis, University of Trento, Trento, Italy, May 2018
5. Rahmati A, Fernandes E, Eykholt K, Prakash A (2018) Tyche: a risk-based permission model for smart homes. In: Proceedings of the 2018 IEEE cyber security development conference, SecDev 2018, Cambridge, MA, USA, 30 September–2 October 2018, pp 29–36
6. Choi D, Kim D, Park S (2015) A framework for context sensitive risk-based access control in medical information systems. Hindawi Publ Corp Comput Math Methods Med 1–10, April 2015
7. Clark JA, Tapiador JE, McDermid J, Cheng P-C, Agrawal D, Ivanic N, Slogget D (2010) Risk based access control with uncertain and time-dependent sensitivity. In: Proceedings of the 2010 international conference on security and cryptography (SECRYPT), Athens, Greece, 26–28 July 2010
8. Chen P, Pankaj C, Karger PA, Wagner GM, Schuett A (2017) Fuzzy multilevel security: an experiment on quantified risk—adaptive access control. In: IEEE symposium on security and privacy (SP'07), pp 222–227
9. Walters RJ, Atlam HF, Wills GB, Daniel J (2019) Fuzzy logic with expert judgment to implement an adaptive risk-based access control model for IoT. Mob Netw Appl 1–13
10. Shafique K, Khwaja BA, Sabir F, Quazi S, Mustakim M (2020) Internet of Things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. Int J Adv Comput Sci Appl 208–219, January 2020. https://ieeexplore.ieee.org/document/8972389/
11. Kavre M, Gadekar A, Gadhade Y (2019) Internet of Things (IoT): a survey, December 2019. https://ieeexplore.ieee.org/xpl/conhome/9102215/proceeding
12. Tabassun K, Ibrahim A, Rahman SAI (2019) Security issues and challenges in IoT, May 2019. https://ieeexplore.ieee.org/document/8716460
13. Gochhayat SP, Lal C, Sharma L (2020) Reliable and secure data transfer in IoT networks, November 2020. https://www.researchgate.net/publication/333788784_Reliable_and_secure_data_transfer_in_IoT_networks
14. Zeebaree SRM (2019). DES encryption and decryption algorithm implementation based on FPGA 1–27, March 2019. https://www.researchgate.net/publication/338544501_DES_encryption_and_decryption_algorithm_implementation_based_on_FPGA
15. Sawant AG, Nitnaware VN, Dengale P, Gandewar SGA (2019) Twofish algorithm for encryption and decryptionJETIR 6(1)
16. Elky S (2006) An introduction to information system risk management. SANS Institute, pp 1–18

17. Watanabe H, Fujimura S, Nakadaira A, Miyazaki Y, Akutsu A, Kishigami J (2016) Blockchain contract: securing a blockchain applied to smart contracts. In: 2016 IEEE international conference on consumer electronics, pp 467–468
18. Chen P, Pankaj C, Karger PA, Wagner GM, Schuett A (2007) Fuzzy multi-level security: an experiment on quantified risk-adaptive access control. In: 2007 IEEE symposium on security and privacy (SP'07), pp 222–230
19. Li J, Bai Y, Zaman N (2013) A fuzzy modeling approach for risk based access control in eHealth cloud. In: 2th IEEE international conference on trust, security and privacy in computing and communications, TrustCom, pp 17–23
20. Khambhammettu H, Boulares S, Adi K, Logrippo L (2013) A framework for risk assessment in access control systems. Comput Secur 39:86–103
21. Diep NN et al (2007) Enforcing access control using risk assessment. In: The fourth European conference on universal multiservice networks, pp 419–424
22. Ni Q, Bertino E, Lobo J (2010) Risk-based access control systems built on fuzzy inferences. In: Proceedings of the 5th ACM symposium on information, computer and communications security, ASIACCS 10. ACM, New York, NY, USA, pp 250–260
23. Rajbhandari L, Snekkenes EA (2011) Using game theory to analyze risk to privacy: an initial insight. In: IFIP PrimeLife international summer school on privacy and identity management for life, Springer Berlin Heidelberg, pp 41–51
24. Khambhammettu H, Boulares S, Adi K, Logrippo L (2013) A framework for risk assessment in access control systems. Comput Secur 39:86–103
25. Wang S et al (2016) A vertical handoff method via self-selection decision tree for internet of vehicles. IEEE Syst J 10(3):1183–1192
26. Shang K, Hossen Z (2013) Applying fuzzy logic to risk assessment and decision-making. Casualty Actuarial Society, Canadian Institute of Actuaries, Society of Actuaries, pp 1–59
27. Goerdin SAV, Smit JJ, Mehairjan RPY (2015a) Monte Carlo simulation applied to support risk-based decision making in electricity distribution networks. In: 2015 IEEE Eindhoven PowerTech, PowerTech 2015
28. Kahneman D, Slovic P, Tversky A (1974) Judgment under uncertainty: heuristics and biases. Science 185(4157):1124–1131

Chapter 20
Machine Learning-Based Platform for Classification of Retinal Disorders Using Optical Coherence Tomography Images

Ahmed M. Salaheldin, Manal Abdel Wahed, and Neven Saleh

1 Introduction

In the last few years, there have been one billion individuals all over the world suffering from vision impairment, with its different degrees of severity starting from myopia to complete vision loss. Vison loss affects the daily lives of these people, including their personal activities and communicating with their relevant in the work and school. Hence, the time of the diagnosis of this problem plays an important role in preventing complete vision loss [1].

Computer-aided diagnosis (CAD) systems emphasize a huge contribution to the detection and diagnosis of different diseases. Because of the advances in engineering technology and the use of the CAD system, human errors will be reduced [2]. The use of artificial intelligence (AI) concepts helps the medical field in different ways, such as diagnosis of diseases, telemedicine, medical surgery, and drug design [3].

The literature is rich with studies that focus on the development of CAD systems that classify different retinal disorders such as CNV, DME, Drusen, and normal cases by representing both machine learning and deep learning model for the same task. Saleh and Salaheldin [4] have proposed a study to select the optimal computer-aided system for the classification of different retinal disorders. In the work presented by Hussien et al. [5], a machine learning-based platform is presented to classify the retinal disorders into age-related macular degradation (AMD), diabetic macular edema, or normal class. Random Forest was used as the classifier of the model and used a 15-fold cross-validation approach. The model achieved more than 95%

A. M. Salaheldin (✉) · N. Saleh
Systems and Biomedical Engineering Department, Higher Institute of Engineering in El-Shorouk City, Shorouk Academy, Cairo, Egypt
e-mail: a.sadek@sha.edu.eg

M. Abdel Wahed
Biomedical and Systems Engineering Department, Faculty of Engineering, Cairo University, Giza, Egypt

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_21

classification accuracy and 0.99 area under the curve (AUC), but the model reached more than 96% accuracy in classifying the image into normal or eye with pathology.

The work has been introduced by Alsaih et al. [6] and proposed a model to distinguish between the different pathologies of the retina into DME and normal. They used three different classifiers in the implementation of the model: linear support vector machine (SVM), RF, and kernel SVM. The optimal performance of the tried models was linear SVM with the support of principal component analysis (PCA) with a sensitivity of 87.5% and specificity of 87.5%. Abdulrahman and Khatib [7] presented a classification model to classify OCT images into CNV, DME, Drusen, and normal classes by using SVM as a classifier and the genetic programming technique to extract the optimal combination of the features that will be classified. The model achieved a classification accuracy of 90.65% based on local and global feature extraction.

A new classification model has been introduced by Srinivasan et al. [8]. The classification process was done using the one-vs-one SVM fashion as a classifier. The model was validated by using the cross-validation technique. The algorithm correctly recognized 100% of AMD cases, 100% of DME cases, and 86.67% of normal cases. A model for the classification of the DME and OCT images was developed by Dash and Sigappi [9]. They presented two techniques for the classification process: local binary pattern with Random Forest and scale-invariant feature transform with Random Forest where the two methods achieved 100% and 88% for the classification accuracy, respectively.

A computerized macular pathology diagnosis model has been proposed by Liu et al. [10]. The model was designed to classify the spectral domain OCT images into the normal macula and three macular pathologies (macular edema, macular hole, and age-related macular degradation). A nonlinear support vector machine was used as a classifier in the model. For diagnosing normal macula, macular hole, macular edema, and AMD, the cross-validation AUC on the development dataset was 0.976, 0.931, 0939, and 0.938, while the AUC on the holdout testing set was 0.978, 0.969, 0.941, and 0.975, respectively.

A computer-aided diagnosis system has been proposed by Saleh et al. [11] to classify the OCT images into four classes: CNV, DME, Drusen, and normal. The model has been implemented based on the use of deep learning concepts, especially the modified SqueezeNet model and the Inception v3 model. The system attained a classification accuracy of 98% and 98.4% for modified SqueezeNet and Inception v3, respectively.

The objective of the study is to implement a hybrid system combining the concepts of machine learning and deep learning aspects to classify the optical coherence tomography images into CNV, DME, Drusen, and normal.

The following is an outline of the article's structure. Section 1 discusses related work on retinal classification diseases using various methods. Section 2 contains a description of the materials and methods used in this investigation. Section 3 depicts the study's findings, which explain the classification of retinal diseases based on the technique used. Section 4 includes a summary and discussion of the study's findings,

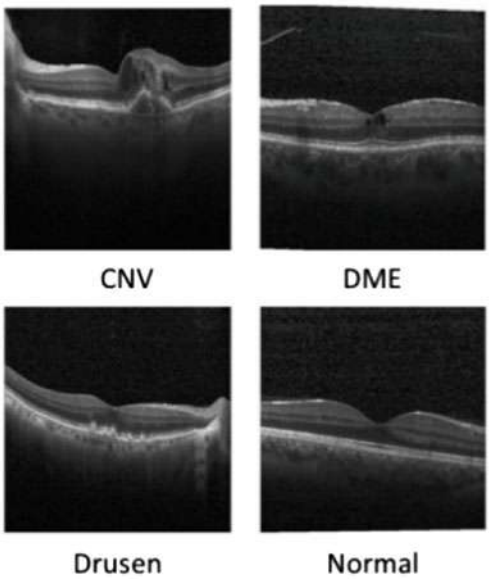


Fig. 1 An example of the OCT image illustrating the four retinal classes

as well as comparisons to another research. Finally, Section 6 concludes the study and presents the future work.

2 Material and Methods

2.1 Dataset

Optical Coherence Tomography images have been chosen from a dataset that belongs to adults that suffer from different retinal disorders as described in a study by Kermany et al. [12]. The data was collected over a period of more than four years to achieve the collection of 83,484 images that were classified into four classes: CNV, DME, Drusen, and normal. The collection of the data has been done at the medical Center of ophthalmology at the California Retinal Research Foundation. Figure 1 shows the difference between the retinal layers with associated pathology, clarifying the four classes.

2.2 Preprocessing Approach

The quality of the input image directly emphasizes the performance of the model. Hence, the need for preprocessing techniques is high to achieve high performance in

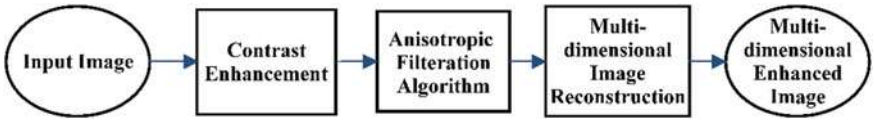


Fig. 2 A block representation of the suggested OCT image preprocessing stage

the classification of the data and the evaluation computation. In the following subsections, the different preprocessing approaches will be discussed. The summary of the preprocessing phase is illustrated in Fig. 2. It starts with the contrast enhancement procedure and progresses to the final multidimensional form of the picture, which will be the input to the suggested CNN architecture.

Image contrast enhancement

To improve the contrast between the distinct retinal layers and the background, a strategy was used in which the intensity values of each image were mapped to the new values by saturating the bottom and top 1% of all pixel values. This process improves the contrast of the produced image, which aids in the clarification of the image's many features.

Anisotropic diffusion filtration

To eliminate the blurring and localization issues associated with linear diffusion approaches, a nonlinear diffusion method is used. This technique is provided by an anisotropic diffusion filtration algorithm, namely the Perona-Malik algorithm [13]. Simply stated, it is a technique for decreasing image noise without destroying considerable quantities of image content. Edges, lines, and other characteristics are typically important for interpreting OCT pictures. Indeed, these components play a significant role in the presentation of several retinal layer diseases. Indeed, these components play an important role in the presentation of several retinal layer diseases. The number of iterations, conduction coefficient, and stability factor must all be calculated before applying the method. There is no set procedure for computing other than trial and error. As a result, these parameters were calculated with the assistance of a medical specialist. The number of iterations was set at 5. The value of the conduction coefficient (k) was chosen to be 40. The stability factor (λ) was set at 0.25. The final improved preprocessed image matrix is computed (1). North gradient (C_{n}), south gradient (C_{s}), west gradient (C_{w}), and east gradient (C_{e}) coefficients have been determined.

$$
\begin{aligned}
M_i &= M_{i-1} + \lambda * (C_{\mathrm{n}} * N + C_{\mathrm{s}} * S + C_{\mathrm{e}} * E + C_{\mathrm{w}} * W) \\
C_{\mathrm{n}} &= \mathrm{e}^{-\left(\frac{N}{k}\right)^2} \\
C_{\mathrm{s}} &= \mathrm{e}^{-\left(\frac{S}{k}\right)^2} \\
C_{\mathrm{e}} &= \mathrm{e}^{-\left(\frac{E}{k}\right)^2} \\
C_{\mathrm{w}} &= \mathrm{e}^{-\left(\frac{W}{k}\right)^2}
\end{aligned}
\tag{1}
$$

where M is the preprocessed image, N is the input image's north gradient component, S is the input image's south gradient component, E is the input image's west gradient component, and W is the input image's west gradient component.

Multidimensional image reconstruction

Because no two OCT images have the same size, the images must be resized to a certain size that corresponds to the input layer of the CNN architecture being used. Every OCT image is retrieved with m × n size. The input size of most CNN input layers in the software accepts a three-dimensional image as input. As a result, we suggested a method for generating a three-dimensional image from a greyscale image. The procedure begins by generating three separate image matrices. Each one carries the original image data in a separate dimension. The matrices are then combined into one matrix of a certain size with a three-dimensional scale and the same original data.

2.3 Classification Algorithms Based on Machine Learning Techniques

There are two scenarios that have been applied during the classification of the data using the different machine learning algorithms. Firstly, the feature extraction phase has been implemented using the SqueezeNet model. Secondly, the transfer learning-based Inception v3 model was used to extract the features from the data. Four classifiers were used in the implementation of the model: support vector machine, k-nearest neighbor, decision tree, and an ensemble model. The classification process regardless of the classifier used is shown in Fig. 3 starting from the feature extraction step, the optimization procedure, classification process, and performance computation.

Feature extraction using squeezeNet model

The proposed model of SqueezeNet model [11] has been used to extract the features from the data from 12,000 OCT images split equally over the four classes. The data partitioning is applied to be 68% for training, 17% for handout validation, and 15% for the testing process. The deep neural network builds a hierarchical representation of the images it receives. Deeper layers have higher-level features that are built on the lower-level elements of earlier layers. To obtain the training images' feature representations, the data was transformed into numerical form by calculating the activation function of the features. Features were retrieved from the added dropout layer to produce 1000 features for each image.

Feature extraction using inception v3 model

The transfer learning-based model of Inception v3 [11] has been used to extract the features which will be used to build the system. The features were extracted from 18,000 OCT images and split into 68% for training, 17% for the handout validation and hyperparameter tuning, and 15% for testing of the model and performance computation. Features were extracted from the global average pooling layer, which provides 2048 features for each image. The huge number of features provides a

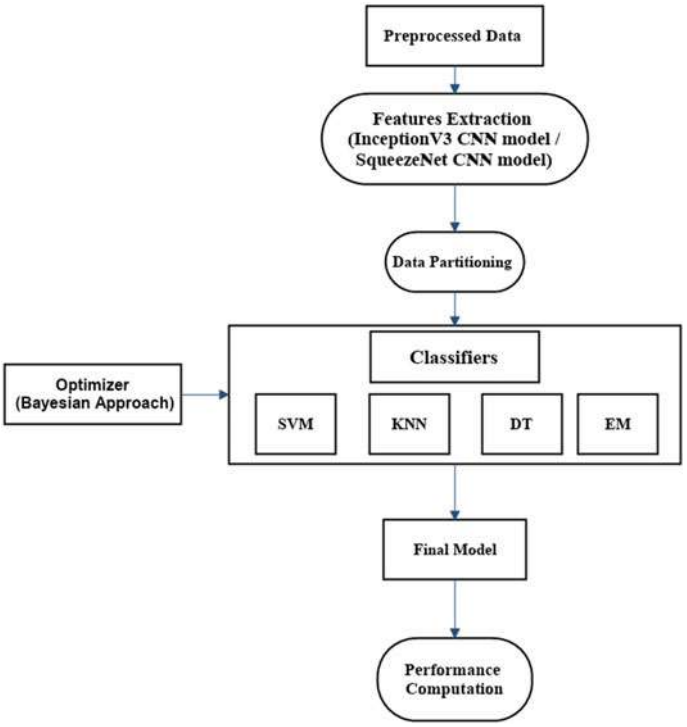


Fig. 3 Machine learning-based platform for classification of retinal disorders

full investigation of the image which directly impacts the classification process. As known, the features extracted from any CNN model are patches of images that were converted into numerical data by calculating the activation function of each patch to be classified into CNV, DME, Drusen, and normal.

Classification using support vector machine classifier

Two scenarios have been implemented; the first one uses the features extracted from the SqueezeNet model and trains the model using 8160 OCT images. The validation of the model has been applied using 2040 OCT images with the association of the Bayesian optimization approach. The second approach has been applied using the features extracted from the Inception v3 transfer learning-based approach. The model was optimized using the Bayesian approach to find the optimal hyperparameters of the model. The system has been constructed using 18,000 OCT images, which are split into 12,240 images for the training, 3060 images for the validation task, and 2700 images for the performance computation, and the data is equally split into the four classified classes. The optimal hyperparameters and the optimizer options of the system are summarized in Table 1, including the kernel function, box constraint level, multiclass method, number of iterations, and acquisition function.

Classification using K-Nearest neighbor classifier

Table 1 Validation and optimization parameters of SVM classifier

Criteria	SqueezeNet model	Inception v3 model
Kernel function	Linear	Linear
Box constraint level	0.10913	0.0010132
Multiclass method	One-vs-One	One-vs-One
Number of iterations	30	30
Acquisition function	Expected improvement per second plus	Expected improvement per second plus

Two different models are implemented; the first model is based on the features extracted from SqueezeNet and the transfer learning-based model Inception v3 model features have been used in the implementation of the second model. The Bayesian optimization approach was used during the implementation of the models to select the different hyperparameters such as the number of neighbors, distance metric, and distance weight. Table 2 summarizes the different hyperparameters and the validation parameters of the two models. The partitioning of the data into the first model is 8160 images for the training, 2040 images for the validation, and 1800 images for the performance computation of the model, while in the second model there are 12,240 images for the training, 3060 images for the validation of the model, and 2700 images for the testing of the model and measuring the evaluation criteria.

Classification using decision tree classifier

In the implementation of the decision tree algorithms, two models have been trained using the features of the SqueezeNet and Inception v3 transfer learning-based model. A dataset of 12,000 OCT images and 18,000 OCT images have been used for the implementation of the model for SqueezeNet and Inception v3, respectively. The Bayesian optimization approach has been used to reach the optimal parameters of the decision tree such as the maximum number of splits and split criterion. Both optimization and validation of different parameters are summarized in Table 3.

Table 2 Validation and optimization parameters of K-NN classifier

Criteria	SqueezeNet model	Inception v3 model
Number of neighbors	144	12
Distance metric	City block	Spearman
Distance weight	Inverse	Inverse
Number of iterations	30	30
Acquisition Function	Expected improvement per second plus	Expected improvement per second plus

Table 3 Validation and optimization parameters for decision tree classifier

Criteria	SqueezeNet model	Inception v3 model
Maximum number of splits	62	61
Split criterion	Maximum deviance reduction	Maximum deviance reduction
Number of iterations	30	30
Acquisition function	Expected improvement per second plus	Expected improvement per second plus

Table 4 Validation and optimization parameters for ensemble model classifier

Criteria	SqueezeNet model	Inception v3 model
Ensemble method	Bagging	AdaBoost
Number of learners	66	39
Maximum number of splits	2364	16
Learning rate	–	0.90426
Number of iterations	30	30
Acquisition function	Expected improvement per second plus	Expected improvement per second plus

Classification using ensemble model classifier

A dataset of 12,000 OCT images and 18,000 OCT images have been used in the building of the system using SqueezeNet and Inception v3, respectively. The dataset is classified into CNV, DME, Drusen, and normal. The Bayesian optimization approach has been applied to find the optimal parameters of the model such as the ensemble method, number of learners, and maximum number of splits. The partitioning of the data into the first model is 8160 images for the training, 2040 images for the validation, and 1800 images for the performance computation of the model, while in the second model there are 12,240 images for the training, 3060 images for the validation of the model, and 2700 images for the testing of the model and measuring the evaluation criteria. Table 4 shows various parameters of the validation and optimization process.

3 Results

The research involves the development of a computer-aided diagnostic system for the classification of retinal abnormalities using OCT images. The MATLAB R2020b machine learning and deep learning toolboxes are used to code the algorithms. Three deep learning models are used to categorize the CNV, DME, Drusen, and normal

classes, while four machine learning approaches are used to create four classifiers. A collection of evaluation matrices is used in this context to evaluate the performance of each method independently. Nine parameters are included in the matrices: accuracy, sensitivity, specificity, precision, error rate, false-positive rate, false-negative rate, negative predictive value, and F1-score. All factors are classified as true positive (TP), true negative (TN), false positive (FP), and false negative (FN) (FN). The nine factors are computed using the formulas (2)–(10) [14] as shown below

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FN} + \text{FP} + \text{TN}) \tag{2}$$

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \tag{3}$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \tag{4}$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{5}$$

$$\text{Error Rate} = (\text{FP} + \text{FN})/(\text{TP} + \text{FN} + \text{FP} + \text{TN}) \tag{6}$$

$$\text{False Positive Rate} = \text{FP}/(\text{FP} + \text{TN}) \tag{7}$$

$$\text{False Negative Rate} = \text{FN}/(\text{FN} + \text{TP}) \tag{8}$$

$$\text{Negative Predictive Value} = \text{TN}/(\text{TN} + \text{FN}) \tag{9}$$

$$\text{F1-Score} = (2 \times (\text{Sensitivity} \times \text{Precision}))/(\text{Sensitivity} + \text{Precision}) \tag{10}$$

3.1 Performance of SVM Classifier

A novel method has been proposed to classify retinal disorders into CNV, DME, Drusen, and normal cases through four classifiers. A hybrid artificial intelligence-based system has been applied on 12,000 and 18,000 OCT images for the SqueezeNet model and Inception v3 model, respectively. The overall performance of the system attains a classification accuracy of 97.39% and 99.43% for the SqueezeNet model and Inception v3 model, respectively. Table 5 shows the different performance criteria selected to evaluate the model.

Table 5 The performance of SVM-based platform

Criteria	SqueezeNet model	Inception v3 model
Classification accuracy	0.9739	0.9943
Sensitivity	0.9478	0.9885
Specificity	0.9826	0.9962
Precision	0.9461	0.9871
Error rate	0.0261	0.0057
False-positive rate	0.0174	0.0038
False-negative rate	0.0522	0.0115
Negative predictive value	0.9827	0.9962
F1-score	0.9465	0.9878

3.2 Performance of K-NN Classifier

Two scenarios have been adopted to classify retinal disorders into CNV, DME, Drusen, and normal classes. The features were extracted using two deep learning networks: SqueezeNet and Inception v3 net. Performance computation achieves a classification accuracy of 97.47% and 99.56% for the SqueezeNet model and Inception v3 model, respectively. Besides classification accuracy, eight evaluation criteria have been computed to evaluate the performance of the model and the overall performance computation is presented in Table 6.

Table 6 The performance of K-NN-based platform

Criteria	SqueezeNet model	Inception v3 model
Classification accuracy	0.9747	0.9956
Sensitivity	0.9494	0.9911
Specificity	0.9831	0.997
Precision	0.9465	0.9893
Error Rate	0.0253	0.0044
False-positive rate	0.0169	0.003
False-negative rate	0.0506	0.0089
Negative predictive value	0.9832	0.997
F1-score	0.9477	0.9902

Table 7 The performance of the Decision Tree-based platform

Criteria	SqueezeNet model	Inception v3 model
Classification accuracy	0.9689	0.9798
Sensitivity	0.9378	0.9596
Specificity	0.9793	0.9865
Precision	0.9334	0.9515
Error rate	0.0311	0.0202
False-positive rate	0.0207	0.0135
False-negative rate	0.0622	0.0404
Negative predictive value	0.9793	0.9866
F1-score	0.9353	0.9555

3.3 Performance of Decision Tree Classifier

An automatic system is used to classify the retinal disorders into CNV, DME, Drusen, and normal. The classification process has been done using a decision tree classifier. The features are extracted from SqueezeNet and Inception v3 models. The system attains a classification accuracy of 96.89% and 97.98% for the SqueezeNet model and Inception v3 model, respectively. The performance computation includes other parameters to evaluate the performance of the system, such as sensitivity, specificity, and precision. Table 7 shows the evaluation criteria for the decision tree classifier for the two models.

3.4 Performance of Ensemble Model Classifier

A machine learning-based model is proposed to classify the OCT images into the main common retinal disorders using the concept of ensemble model as a classifier, with the role of CNN model, SqueezeNet and Inception v3 models as a feature extractor. The system attains a classification accuracy of 97.25% and 99.31% for the SqueezeNet model and Inception v3 model, respectively, and the other evaluation parameters are presented in Table 8.

3.5 Performance of All Proposed Models

Eight models are presented to classify OCT images into CNV, DME, Drusen, and normal classes. The classification models are presented in numbers as given in Table 9.

Table 8 The performance of Ensemble Model-based platform

Criteria	SqueezeNet model	Inception v3 model
Classification accuracy	0.9725	0.9931
Sensitivity	0.945	0.9863
Specificity	0.9817	0.9954
Precision	0.9421	0.9845
Error rate	0.0275	0.0069
False-positive rate	0.0183	0.0046
False-negative rate	0.055	0.0137
Negative predictive value	0.9817	0.9954
F1-score	0.9435	0.9854

Table 9 Classification models as number list

SVM-SqueezeNet platform	Model (1)
SVM-Inception v3 platform	Model (2)
K-NN-SqueezeNet platform	Model (3)
K-NN-Inception v3 platform	Model (4)
Decision Tree-SqueezeNet platform	Model (5)
Decision Tree-Inception v3 platform	Model (6)
Ensemble Model-SqueezeNet platform	Model (7)
Ensemble Model-Inception v3 platform	Model (8)

4 Discussion

The study proposes a novel model for the diagnosis and classification of the most commonly ophthalmological disorders: CNV, DME, Drusen, and normal cases. The classifiers are SVM, K-NN, DT, and EM. The performance evaluation of each model is computed using nine criteria; accuracy, sensitivity, specificity, precision, error rate, false-positive rate, false-negative rate, and F1-score. The tuning of the hyperparameters of each model is carried out using a Bayesian optimization algorithm. Further, the results of each model are validated using 2040 OCT images and 3060 OCT images for the SqueezeNet model and Inception v3, respectively, while the testing phase is performed using 1800 OCT images and 2700 OCT images for the SqueezeNet model and Inception v3, respectively. The images are equally split into four classes.

The highest value of specificity is calculated for the DME class, except the SVM is provided for the CNV. For precision, the best value is provided for the CNV class. As such, the CNV class is considered the winning card among the other classes. This can be explained by the characteristic hyper-reflective appearance of CNV in OCT images compared with the other classes. By regarding the overall performance of

the four classifiers, the K-NN classifier has attained the highest values considering all the nine criteria.

Comparing the study with the related works is presented in Table 10. The comparison considers the number of classes, the classifier, tested sample size, accuracy, sensitivity, and specificity. According to the table, the proposed models achieved high accuracy when compared to the works in [5–7, 9]. Furthermore, the classifiers are invariant in the number of classes. On the contrary, the proposed SVM model has attained better classification accuracy compared with the same classifiers in the literature. Besides, it enhances the accuracy by 4% compared with the work that was presented by Srinivasan et al. [8].

Relevant studies have presented the classification with only two classifiers: the SVM and the RF. Moreover, the tested sample range is 40–251 OCT images. The proposed work presents four classifiers, three of which are for the first time in this problem. Additionally, the number of tested samples is 1800 OCT images and 2700 OCT images for the SqueezeNet model and Inception v3, respectively. Besides,

Table 10 Comparison between machine learning algorithms and relevant studies

Study	Classes	Classifier	Tested samples	Accuracy	Sensitivity	Specificity
Hussain et al. [5]	2	RF	251	95%	94.67%	100%
Alsaih et al. [6]	2	SVM	238	NA	87.50%	87.50%
Abdulrahman and Khatib [7]	4	SVM	160	90.65%	NA	NA
Srinivasan et al. [8]	3	SVM	45	95.50%	NA	NA
Dash and Sigappi [9]	2	RF	40	100% for LBP, 88% for SITF	100% for LBP, 91% for SITF	NA
Liu et al. [10]	4	SVM	131	90.50%	NA	NA
Saleh et al. [11]	4	Modified SqueezeNet	1000	98%	95.9%	98.63%
Saleh et al. [11]	4	Inception v3	2700	98.4%	96.7%	98.9%
Model (1)	4	SVM-SqueezeNet	1800	97.39%	94.78%	98.26%
Model (2)	4	SVM-Inception v3	2700	99.43%	98.85%	99.62%
Model (3)	4	K-NN-SqueezeNet	1800	97.47%	94.94%	98.31%
Model (4)	4	K-NN-Inception v3	2700	99.56%	99.11%	99.70%
Model (5)	4	DT-SqueezeNet	1800	96.98%	93.78%	97.93%
Model (6)	4	DT-Inception v3	2700	97.98%	95.96%	98.65%
Model (7)	4	EM-SqueezeNet	1800	95.25%	94.50%	98.17%
Model (8)	4	EM-Inception v3	2700	99.31%	98.63%	99.54%

regarding the number of classes, which is four, concerning the classification accuracy, we have achieved the best results. Our models have achieved accuracy of 99.43%, 99.54%, 97.98%, and 99.31% compared with 90.65% and 90.50% of the studies [7] and [10], respectively. Hence, we added more evaluation criteria, i.e., six criteria, some of which are presented for the first time for the underlined problem. Dash and Sigappi have presented 100% classification accuracy using LBP for feature extraction, but they used only 40 OCT images for classifying only two classes [7]. The proposed models achieved good overall performance over the relevant studies in the same task of the classification of retinal disorders. The limitation of the study is that only three retinal disorders have been detected according to the source of data. Furthermore, for considering huge datasets of images, a supercomputer is required to reduce the time of training.

5 Conclusion

Classifying retinal disorders based on OCT images is a state-of-the-art topic in the field of automated detection and preliminary diagnosis. The study presents a novel ensemble of four classes of retinal disorder: CNV, DME, Drusen, and normal cases. Four classifiers have been presented, namely SVM, K-NN, DT, and EM. The features have been extracted using the SqueezeNet and Inception v3 CNNs. We present a novel technique based on a machine learning approach to the underlined problem. The proposed models have attained significant results compared with the relevant studies. The K-NN model has achieved the optimal classification accuracy of 99.56%. The study introduces nine evaluation criteria for performance computation. Indeed, the study provides a novel platform for aiding ophthalmologists in the diagnosis and classification of retinal diseases, which in turn reduces effort and time.

References

1. Blindness and vision imapairment, world health organization, 21 Feb 2021. [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment
2. El-Hossiny AS, Al-Atabany W, Hassan O, Soliman AM, Sami SA (2021) Classification of thyroid carcinoma in whole slide images using cascaded CNN. IEEE Access 9:88429–88438
3. Ng A (2021)Machine Learning. Coursera, 1 March 2021. [Online]. Available: https://www.coursera.org/learn/machine-learning
4. Saleh N, Salaheldin AM (2021) A benchmarking platform for selecting optimal retinal diseases diagnosis model based on a multi-criteria decision-making approach. J Chinese Inst Eng https://doi.org/10.1080/02533839.2021.1983466
5. Hussain MA, Bhuiyan A, Luu CD, Smith RT, Guymer RH, Ishikawa H, Schuman JS, Ramamohanarao K (2018) Classification of healthy and diseased retina using SD-OCT imaging and Random Forest algorithm. PLOS ONE 13(6):e0198281

6. Alsaih K, Lemaitre G, Rastgoo M, Massich J, Sidibé D, Meriaudeau F (2017) Machine learning techniques for diabetic macular edema (DME) classification on SD-OCT images. BioMedical Eng. OnLine 16(1)
7. Abdulrahman H, Khatib M (2020) Classification of retina diseases from OCT using genetic programming. Int J Comput Appl 177(45):41–46
8. Srinivasan PP, Kim LA, Mettu PS, Cousins SW, Comer GM, Izatt JA, Farsiu S (2014) Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images. Biomed Opt Express 5(10):3568
9. Dash P, Sigappi A (2018) Detection and recognition of diabetic macular edema from oct images based on local feature descriptor. Int J Pure Appl Math 119(12):1–8
10. Liu Y-Y, Ishikawa H, Chen M, Wollstein G, Duker JS, Fujimoto JG, Schuman JS, Rehg JM (2011) Computerized macular pathology diagnosis in spectral domain optical coherence tomography scans based on multiscale texture and shape features. Invest Opthalmol Vis Sci 52(11):8316
11. Saleh N, Abdel Wahed M, Salaheldin AM (2021)Transfer learning-based platform for detecting multi-classification retinal disorders using optical coherence tomography images. Int J Imag Syst Tech https://doi.org/10.1002/ima.22673
12. Kermany DS, Goldbaum M, Cai W, Valentim CC, Liang H, Baxter SL, McKeown A, Yang G, Wu X, Yan F, Dong J, Prasadha MK, Pei J, Ting MY, Zhu J, Li C, Hewett S, Dong J, Ziyar I, Alexander A, Zhang R, Zheng L, Hou R, Shi W, Fu X, Duan Y, Huu VA, Wen C, Zhang ED, Zhang CL, Li O, Wang X, Singer MA, Sun X, Xu J, Tafreshi A, Lewis MA, Xia H, Zhang K (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172(5):1122-1131.e9
13. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 12(7):629–639
14. Smolyakov V (2021)Ensemble learning to improve machine learning results. Medium, 22 Aug 2017. [Online]. Available: https://blog.statsbot.co/ensemble-learning-d1dcd548e936. Accessed 18 May 2021

Chapter 21
CFD Studies on Hydrodynamics and Heat Transfer of Nanofluid Flow in Straight and Helical Tubes

M. F. Ahmed and M. M. Mandal

1 Introduction

Fluid containing nanometer-sized particles is known as a nanofluid. These types of fluids are synthesized by adding nanoparticles in base fluids like water or ethyl glycol. The nanoparticles which are used in preparing nanofluids are typically carbon nanotubes, metals, metal oxides, carbides, etc. Previous investigations have reported that thermophysical properties of nanofluids such as thermal conductivity, viscosity, thermal diffusivity, and convective heat transfer coefficients are enhanced as compared to conventional base fluids [1].

Enhancement in energy efficiency and improvement in the performance of devices used in the electronics, aerospace, telecommunications industries are some of the key areas for research work. It has been found that devices that are currently being used emit heat in excessive amounts per unit area since the component integration is very large, and the space they are present in is compact and limited. This, in turn, has a drastic effect on the efficiency and shelf life of the device and results in an increase in the cost of maintenance and energy consumption. Therefore, these days' industries are motivated to employ energy-saving policies. Researchers have applied several approaches of process intensification to improve heat transfer in the area of thermal engineering and science [2]. One of the approaches was to enhance the thermal conductivity of the fluid, and because of this, there has been an increase in the use of nanofluids.

The conventional heat transfer fluids that were commonly used have low cooling capacity. One of the major thrust areas of nanotechnology is further increasing the cooling capacity of conventional heat transfer fluid. They are believed to have enhanced thermal properties in comparison with conventional fluids or suspensions.

M. F. Ahmed · M. M. Mandal (✉)
Guru Gobind Singh Indraprastha University, New Delhi 110078, India
e-mail: monishamridha1@ipu.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_22

The smaller particle size and larger specific surface area of nanoparticles enhance their stability as well as their heat transfer efficiency [1].

Experimental studies on the flow of nanofluids under laminar conditions have been done by many researchers. Wen and Dieng [3] studied the enhancement of heat transfer for alumina nanofluid in a horizontal tube with Reynolds numbers up to 1950. The concentration of nanoparticles was kept below 2%, and it was observed that the enhancement was up to 50% in a few of the tests conducted. While it was found out that conventional correlations failed to predict the heat transfer coefficient correctly. Particle migration, disturbed hydrodynamics, and thermal boundary layers were presented as the reasons for the obtained results [4]. Kim et al. [5] worked on alumina and carbonic nanofluid in a laminar and turbulent horizontal tube and where a constant heat flux was applied. The volume fraction for both was taken as 3%, and it was reported that alumina-based nanofluids produced better enhancement of heat transfer coefficient as compared to others due to particle migration at the entrance of the tube. Utomo et al. [6] worked on the impacts of the presence of alumina, titania, and carbon nanotube in a horizontal tube. It was found out that the pressure drops, heat transfer coefficients, and local Nusselt number were in the range of 10% error of existing correlations for straight tubes. They reported that the nanofluid mixture was highly homogenous, and the effect of thermophoresis, Brownian motion, and non-uniform shear rate can be negligible in heat transfer. Yu et al. [7] developed a model for forced convection flows under a laminar regime which was investigated numerically using Al_2O_3 nanofluid in a compact circular tube. Uniform heat flux was assumed at the walls, and a single-phase fluid model was defined for the calculation of nanofluid physical and thermal properties. A two-phase granular mixture model was processed for 100 nm diameter of particles. The results obtained showed with an increase in volume fraction of Al_2O_3, the heat transfer rate and Nusselt number were also improved.

In the last few decades, researchers have found that coiled tubes enhance the heat transfer performance because of the action of Dean vortices arising from centrifugal forces in the curved geometry. Acharya et al. [8] analyzed the heat transfer enhancement in coiled tube heat exchangers under laminar flow regime and was able to develop correlations for constant axis coil and alternating axis coil for different ranges of Prandtl number. The Nusselt number showed a constant variation for both types of coils along with unit cells. The study established the viability of using chaotic mixing as a significant tool for developing an efficient coil tube heat exchanger. The heat transfer enhancement was around 7–20% in the case of alternating axis coil.

Manlapaz and Churchill [9] did a study on fully developed laminar convection in helical tubes and developed a numerical solution for helical flow with finite pitch. They proposed a general correlation for a varied range of Prandtl as well as Dean numbers. Mishra and Gupta [10] developed correlations by conducting numerical studies on momentum transfer of Newtonian as well as non-Newtonian fluids in curved tubes. The effects of various parameters like variation in coil diameter and coil tube diameter and pitch on pressure drop were taken into account. Kumar and Nigam [11] investigated the flow developments as well as temperature fields in an innovative device made by bending the coils. Thermal fields and flow fields were

studied using Fluent software. The flow pattern was in good agreement with previous investigators, and the configuration used resulted in a 20–30% increment in heat transfer efficiency, whereas relative pressure drop was around 5–6%.

Sangeeta Soibam et al. [1] investigated the hydrodynamics of TiO_2–water nanofluid both experimentally and numerically in straight and coiled tubes. A mixture model with the no-slip condition at the walls was used, and no diffusional flux was allowed at the outlet. The pressure drop was found to be within ±2% difference when compared to the experimental results. The values of pressure drop were recorded and found to increase with the rise in velocity, while the increase in concentration didn't affect the pressure drop much. The friction factor decreased by almost 67% when the Reynolds number was increased from 424 to 1230. It was also reported that the values of pressure drop reduced with an increment in curvature ratio of the coiled tube. Hence, it was concluded from the research that the model developed could be used to find and get a comprehensive understanding of the hydrodynamics of nanofluids to flow in several other complex tubular reactors as well as heat exchangers [1].

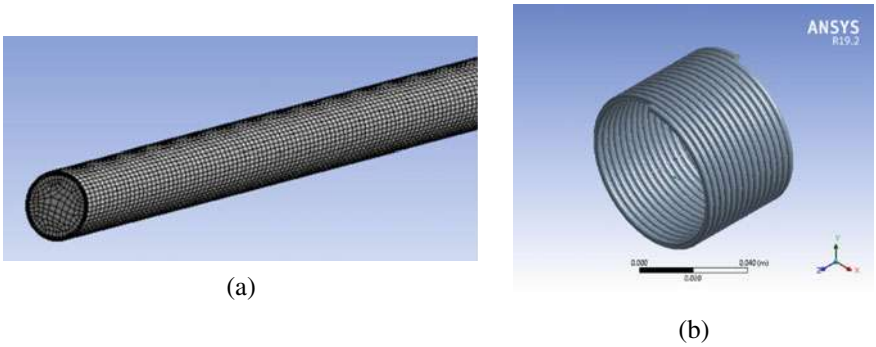
In this work, the mixture model approach in Ansys fluent was used for modeling the flow of Al_2O_3 and TiO_2 in water nanofluids. The simulations were computed for laminar flow conditions in a straight tube and helical tube of equal dimensions. The pressure drop findings for TiO_2–water nanofluid were compared with the experimental data for the geometry of identical dimensions cited in the literature [1]. The pressure drops for both nanofluids were also compared using the same simulation conditions. The effect of variation in velocity and concentration of the nanofluids on pressure drop and heat transfer was investigated. Velocity and temperature contours were obtained and analyzed.

2 Methodology

2.1 Governing Equations

A straight and coiled tube geometry of 0.003 m diameter and 1.6 m in length was created using the design modeler of Ansys Fluent 19.2 for the present study. The curvature ratio (coiling diameter to tube diameter), λ, of the coiled tube was 10 as shown in Fig. 1. An unstructured with non-uniform grid system was used to mesh the straight and coiled tube geometry.

The mixture model was used in Ansys Fluent to investigate the flow of two-phase nanofluids, taking reference from literature [1]. Two types of nanofluids, i.e., TiO_2 and Al_2O_3 nanoparticles were simulated separately for this numerical modeling. The base fluid was taken as water. Thermophysical properties of TiO_2 and Al_2O_3 nanoparticles were taken from Kahani et al. [12] and Devarajan et al. [13], respectively and are shown in Table 1. Concentrations of nanofluids were varied from 1 to 2% were used in the simulation. The computation was carried out using Ansys Fluent 19.2 where the governing equations for continuity, momentum were solved using the



(a) (b)

Fig. 1 **a** Straight tube, **b** coiled tube geometry

Table 1 Thermophysical properties of nanofluids [12, 13]

S. No.	Property	TiO_2	Al_2O_3
1	Density (kg/m^3)	4175	3890
2	Thermal conductivity (W/m K)	8.4	40
3	Specific heat capacity (J/kg K)	692	880

appropriate boundary conditions. [14] The incompressible flow was assumed to be laminar for the present study, and the energy equation was enabled.

At the inlet of the tube, the velocity of the nanoparticles as well water was specified along with the volume fraction of nanoparticles. The velocity for both nanofluids was varied from 0.12 to 0.58 m/s. The pressure was set to constant at the outlet, and diffusional fluxes were set to zero, while at the wall, no-slip conditions were assumed.

$$\frac{\partial(u_i, p, T)}{\partial n} = 0 \tag{1}$$

Here, n denotes the normal coordinate in a direction perpendicular to the outlet plane. The residuals were reduced by a factor of 10^{-6} to increase the accuracy. A SIMPLE algorithm was used for solving the pressure velocity coupling, and second-order upwind interpolation was used for momentum calculation. The simulations were run, and the results were obtained after the solution got converged. The solution was converged if the residual summed overall the computational nodes at nth iteration, fulfilled the belowmentioned criteria.

$$\frac{R_\phi^m}{R_\phi^n} \leq 10^{-5} \tag{2}$$

where R_ϕ^m designates the maximum value of residual variable ϕ after the m iterations, ϕ is applied for p, $u_{i.}$, T. For heat transfer studies, the following expression

was used to calculate the Nusselt number.

$$N_{\mathrm{Nu},\theta} = \frac{q_w d_h}{k(T_w - T_b)} \tag{3}$$

$$N_{\mathrm{Nu},m} = \frac{1}{\overline{u_s} A} \int_0^A N_{\mathrm{Nu},\theta} \mathrm{d}\theta \tag{4}$$

where q_w is total heat flux, d_h is the diameter of the tube, u_s is the velocity of the liquid, A is heat transfer area and k is thermal conductivity, and T_w and T_b are wall and bulk temperature, respectively.

3 Results and Discussions

3.1 Grid Test

A grid test was performed for checking the reliability of the mesh generated for the straight tube and the coiled tube. The pressure drop obtained at different nodes was noted for the flow of TiO_2–water nanofluid flowing in the straight tube. The simulated results were compared by the experimental results as obtained by Soibam et al. [1]. The velocity at the inlet was taken as 0.86 m/s.

Table 2 shows that the grid with 1,182,956 nodes was sufficient to obtain results comparable to the experimental value reported in the literature for the straight tube [15]. When the nodes were further increased, there wasn't any significant increase in the pressure drop. Similarly, a grid test was performed for the coiled tube of the same dimensions. The grid with 2,778,589 nodes was satisfactory for obtaining the pressure drop values in good agreement with the literature [16].

Table 2 Grid test for straight tube for velocity = 0.86 m/s

Number of nodes	Pressure difference	Experimental value [1]	Error (%)
722,304	5591.72	5747	2.7
903,060	5641.04	5747	1.84
1,033,407	5672.08	5747	1.30
1,182,956	5729.43	5747	0.30

3.2 *Velocity Contours*

Velocity contours help to understand the flow pattern of fluids flowing inside of the tubular reactors. The velocity contour at the outlet of the straight tube was obtained for the flow of nanofluids and is as shown in Fig. 2.

Maximum velocity is marked in red whereas as a minimum with blue color. It can be seen that the velocity was maximum at the center of the tube and decreased toward the tube wall. This was due to the no-slip condition assigned as a boundary condition on the wall. The velocity of the fluid was considered as zero at the tube wall. The velocity contour obtained for Al_2O_3–water nanofluid in the coiled tube is shown in Fig. 3.

It can be seen that the maximum velocities of the nanofluid gradually moved toward the outer side of the coiled tube. This happened because centrifugal force

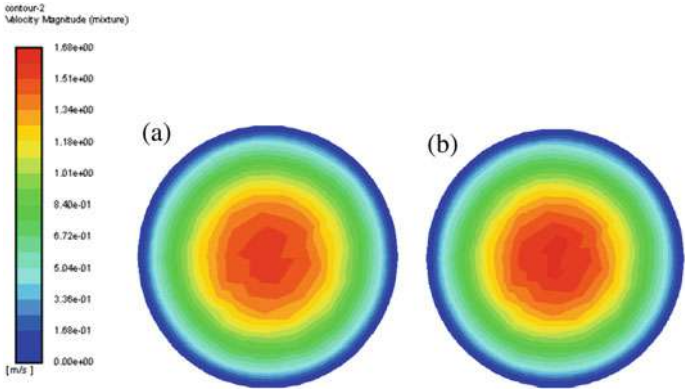


Fig. 2 Velocity contour for **a** TiO_2–water, **b** Al_2O_3 flow in a straight tube

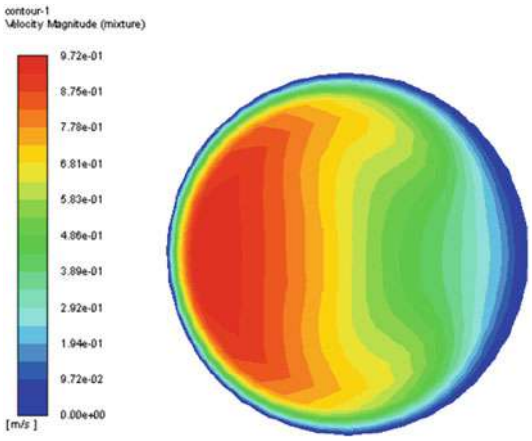


Fig. 3 Velocity contour for Al_2O_3–water nanofluid in a coiled tube

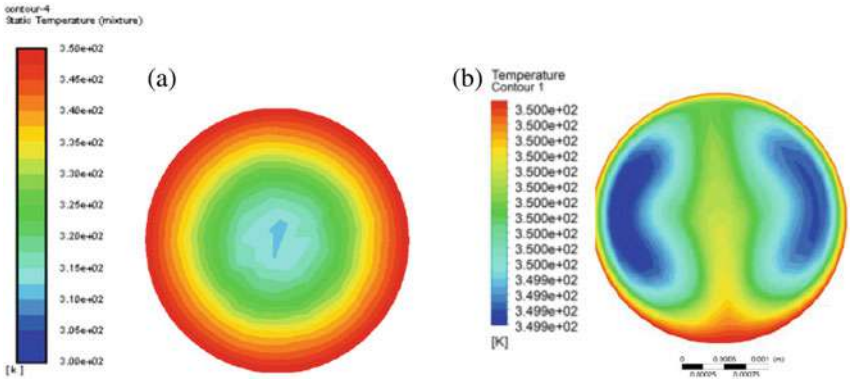


Fig. 4 Temperature contour of Al_2O_3–water nanofluid at the outlet of **a** straight tube, **b** coiled tube

was generated due to the curvature of the tube. The velocity at the boundary marked in blue color tends to be zero because of the no-slip conditions assumed at the tube wall.

3.3 Temperature Contours

A constant temperature of 350 K was considered at the tube wall, while the inlet temperature of nanofluid was taken as 300 K. The temperature contours observed at the outlet of the straight and coiled tube are shown in Fig. 4. The highest temperature is shown in red color, whereas the minimum temperature is in blue. The temperature was found to be lesser in the center of the straight tube. It increased toward the tube wall because uniform wall temperature was provided with higher temperature as a boundary condition. The figure clearly shows the two kidney-shaped vortices created due to Dean vortices in the coiled tube as cited in the literature [8–11, 15]. The ranges of temperature in the contours show that the temperature is more uniform in the coiled tube as compared to the straight tube. This shows that heat transfer was more significant in the case of a coiled tube because of intense mixing due to Dean vortices.

3.4 Pressure Drop

Pressure drop of nanofluids was observed at different velocities and concentrations to comprehend the energy losses by nanofluids flowing through tubes. The velocity of nanofluid was varied from 0.12 to 0.58 m/s. It was observed that there was an increase in the pressure drop between the inlet to the outlet of the straight tube with

an increase in the velocity of nanofluids and is shown in Fig. 5. The results obtained in the present study were validated with existing literature [1] and were in good agreement. There was no noteworthy change in values of pressure drop when the concentrations were increased in a straight tube [1].

Figure 6 shows the variation of velocity from 0.12 to 0.58 m/s at the inlet of the straight tube for Al_2O_3–water fluid for different concentrations ranging from 1 to 2%. The results displayed in Fig. 6 suggested that there wasn't any notable change in pressure drop when the concentration was varied, but the pressure drop values augmented with an increase in nanofluid velocity.

It was also observed that the pressure drop for the different velocities for Al_2O_3 was lesser than the pressure drop for TiO_2. Figure 7 shows that pressure drop was lesser by 1.6% for 0.12 m/s velocity, and it was lesser by 19.2% at a maximum velocity of 0.58 m/s at the inlet for 2% concentration. The change in pressure drop values increased between both the nanofluids with an increase in velocity. This showed that Al_2O_3 nanofluid consumed lesser power to flow through the tubes.

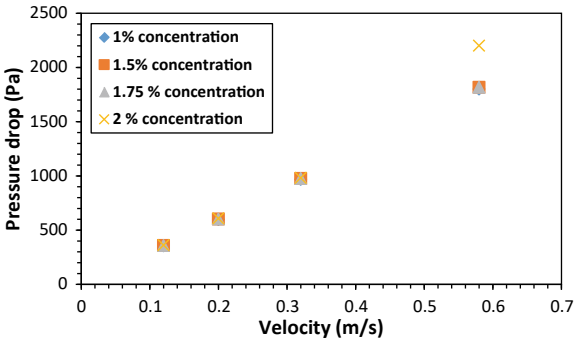


Fig. 5 Effect of variation of velocity and concentration on pressure drop (TiO_2)

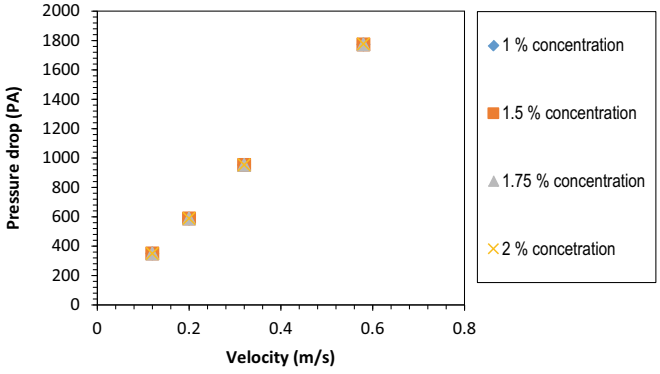


Fig. 6 Pressure drop for different Al_2O_3 nanofluid concentrations with varying velocities in a straight tube

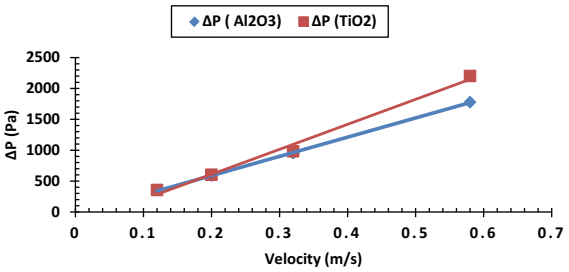


Fig. 7 Comparison of pressure drop for Al_2O_3 and TiO_2 at different velocities

To check the effect of variation of velocity on pressure drop, the velocity was increased from 0.12 m/s to 0.58 m/s in the coiled tube. The effect of concentration increase was also noted by varying the concentration from 1 to 2% as shown in Figs. 8 and 9 for TiO_2 and Al_2O_3 nanofluids, respectively. It was observed that the values of pressure drop increased with a rise in the velocity of nanofluid, but no significant

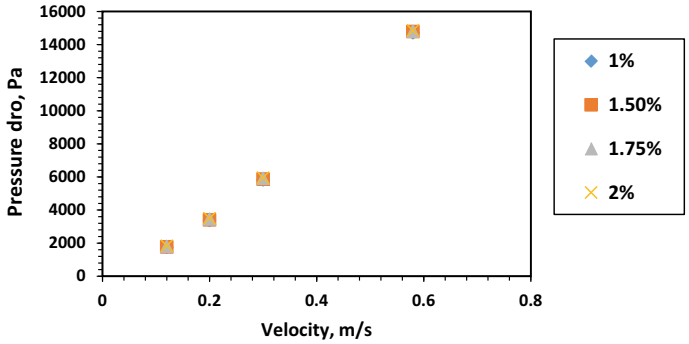


Fig. 8 Effect of variation of velocity on pressure drop for different concentrations of TiO_2 nanofluid in coiled tube

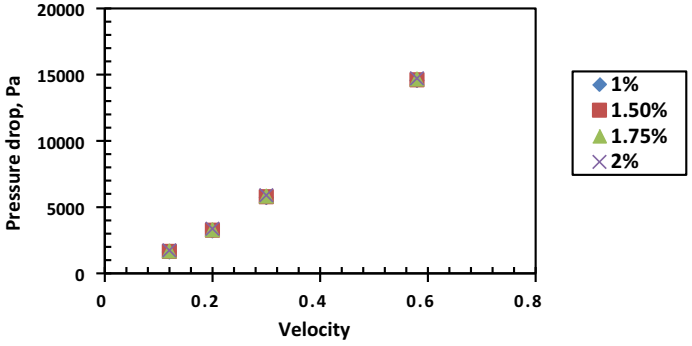


Fig. 9 Pressure drop variation with increasing velocity in Al_2O_3 nanofluid in a coiled tube

change in pressure drop values was found for an increase in nanofluid concentration. This was similar to what was observed in the case of straight tube geometry, but the pressure drop was greater than that of straight tube. It was noted the pressure drop was lesser in Al_2O_3 as compared to TiO_2 nanofluid in the coiled tube.

3.5 Heat Transfer

Heat transfer studies were carried out by calculating the Nusselt numbers for the nanofluids flowing at different velocities and concentrations in the coiled tube. The results obtained from the present computations were validated using the correlation given by Acharya et al. [8] for plain water. The computed results are reported in Fig. 10. The data obtained were in good agreement with the values calculated from the correlation cited in the literature. After validation with water, further computations were carried out for nanofluids. Figure 11 shows the comparison of Nusselt number with Reynolds number in the coiled tube for Al_2O_3, TiO_2 nanofluid with plain water. It was found that the Nusselt number in nanofluid was 13% higher than water in the straight tube for Reynolds number varying from 700 to 1300. This showed that nanofluid may influence the heat transfer capabilities if introduced in plain water. It was further observed that the Nusselt number was higher for the same Reynolds number for Al_2O_3 nanofluids as compared to TiO_2 which can be accounted to better thermal conductivity of Al_2O_3 nanofluid and better specific heat capacity. Figure 12 shows the comparison of Al_2O_3–water nanofluid in coiled and straight tubes. The Nusselt number was found to be greater than a factor of 15% at a minimum value of Re of 700 and by 33.4% at the maximum Reynolds number of 1342.

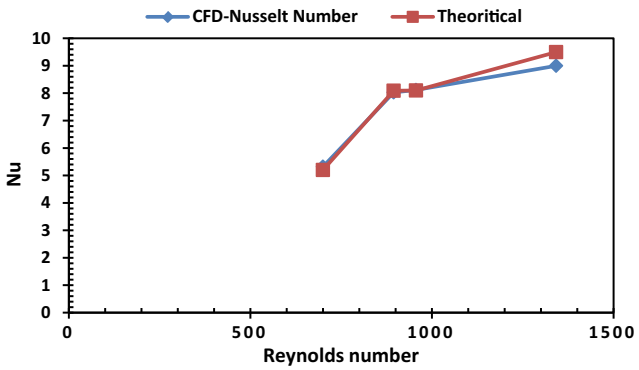


Fig. 10 Comparison of Nusselt number with Reynolds number in the coiled tube for present work with Acharya et al. [8] correlation for water

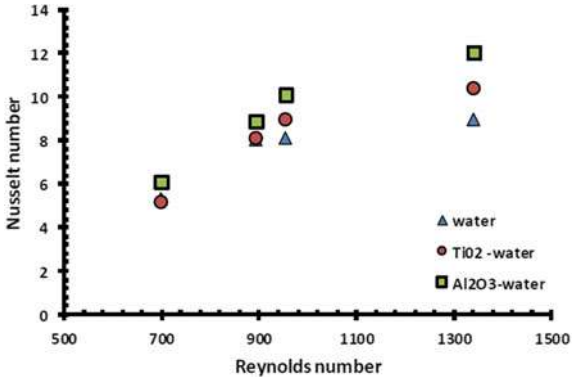


Fig. 11 Comparison of Nusselt number in coiled tubes with Reynolds number for water, Al_2O_3, and TiO_2 nanofluid in coiled tube

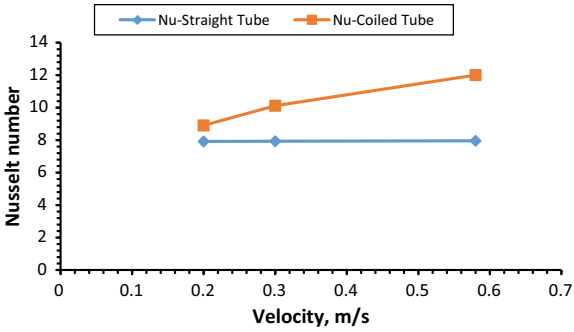


Fig. 12 Comparison of Nusselt number of Al_2O_3–water nanofluid at different velocities flowing in the straight tube and coiled tube of same dimensions

4 Conclusions

In the present study, a CFD model was developed to investigate hydrodynamics and heat transfer performance of nanofluids such as TiO_2 and Al_2O_3 with water as base fluid in a straight tube and coiled tube geometry with the same dimensions. The current studies show that there wasn't any significant difference in the pressure drop values with varying concentrations from 1 to 2% in both TiO_2 and Al_2O_3 nanofluids in the tubes. There was an increase in pressure drop with an increment in velocity from 0.12 to 0.58 m/s. It was found that the pressure drop was 1–19% lesser for Al_2O_3 nanofluid as compared to TiO_2 nanofluid. Heat transfer was higher in the coiled tube by 10–15% due to the improved mixing of nanofluid. Al_2O_3–water nanofluid may be preferred over TiO_2–water fluid since it produced more energy-efficient results due to better thermophysical properties and showcased higher heat transfer rates and less pressure drop as compared to TiO_2. This work shows that nanofluids produce better heat transfer as compared to base fluids such as water. The present model can be utilized to further develop concepts on hydrodynamics and heat transfer enhancements of the flow of nanoparticles in other complex devices like heat exchangers and chaotic tubular reactors.

Acknowledgements The authors would like to acknowledge Guru Gobind Singh Indraprastha University, New Delhi, for financial as well as administrative support to carry out this research work.

References

1. Soibam S, Babita, Sharma SK, Mandal MM (2020) Experimental and numerical investigations on hydrodynamics of nanofluid flowing in coiled tubes. Integr Ferroelectr 205(1):103–113
2. Wong KV, Leon OD (2010) Applications of nanofluids: current and future. Hindawi Publishing Corporation Adv Mech Eng 11
3. Wen D, Ding Y (2004) Experimental investigation into convective heat transfer of nanofluids at the entrance region under laminar flow conditions. Int J Heat Mass Transf 47:5181–5188
4. Rea U, McKrell T, Hu L, Buongiorno J (2009) Laminar convective heat transfers and viscous pressure loss of alumina–water and zirconia–water nanofluids. Int J Heat Mass Transf 52:2042–2048
5. Kim D, Kwon Y, Cho Y, Li C, Cheong S, Hwang Y, Lee J, Hong D, Moon S (2009) Curr Appl Phys 9:119–123
6. Utomo AT, Haghighi EB, Zavareh AIT, Ghanbarpourgeravi M, Poth H, Khodabandeh A, Palm B, Pacek AW (2014) The effect of nanoparticles on laminar heat transfer in a horizontal tube. Int J Heat Mass Transf 69:77–91
7. Yu K (2017) CFD analysis of nanofluid forced convection heat transport in laminar flow through a compact tube. J Phys Conf Ser 885:012021
8. Acharya N, Sen M, Chang HC (2001) Analysis of heat transfer enhancement in coiled tube heat exchangers. Int J Heat Mass Transf 44:3180–3199
9. Manlapaz RL, Churchill SW (1981) Fully developed laminar convection from a helical coil. Chem Eng Commun 9:185–200
10. Mishra P, Gupta SN (1979) Momentum Transfer in curved tubes. 1. Newtonian fluids. Ind Eng Chern Process Des Dev 18(1):130–137
11. Kumar V, Nigam KDP (2005) Numerical simulation of steady flow fields in coiled flow inverter. Int J Heat Mass Transf 48:4811–4828
12. Kahani M, Heris SZ, Mousavi SM (2014) Experimental investigation of TiO_2/water nanofluid laminar forced convective heat transfer through helical coiled tube. Heat Mass Transfer 50:1563
13. Devarajan MB, Krishnamurthy NP, Balasubramanian M, Ramani B, Wongwises S, Kabeel, Sathyamurthy R (2018) Thermophysical properties of CNT and CNT/Al_2O_3 hybrid nanofluid. Micro Nano Lett 1–5
14. Ansys Fluent Guide: https://www.ansys.com/en-in/products/fluids/ansys-fluent
15. Dean WR (1927) Note on the motion of fluid in a curved tube. Philos Mag 4:208–223
16. Mori Y, Nakayama W (1965) Study on forced convective heat transfer in curved tubes (1st Report, Laminar Region). Int J Heat Mass Transf 8:67–82. https://doi.org/10.1016/0017-9310(65)90098-0

Chapter 22
Bank Telemarketing Prediction Based on Deep Learning Approach

Kanchan Lata Kashyap, Nishq Poorav Desai, Avanish Sandilya, and Sarthak Patel

1 Introduction

Telemarketing is a direct marketing method in which a salesperson contacts a potential customer and solicits the customer to buy a product or service. Sometimes, recorded sales and pitches are played on communication devices via automatic dialing to speed up the process of marketing [1]. Nearly 70% of the marketing managers have reputed 500 companies said that telemarketing is "very efficient" for contacting potential customers, those who say it is only "efficient," the percentage is close to 90%. Short-term direct marketing activities are not feasible, and on average, it takes more than 5 months for the company to get real results. According to one survey, customers respond more to surveys containing personal touch rather than the types of marketing like digital marketing. Therefore, telemarketing has been proven to be a better option for bank telemarketing.

1.1 Application Scenarios

Telemarketing is also utilized extensively during political campaigns in order to boost funds and increase popularity [2]. Telemarketing is beneficial for the customers having small business's products or services located at remote places. However, certain miniscule businesses depend on the telephone, and telemarketing is widely utilized in the general marketing program to bind the advertising and individual sales efforts. By the means of telemarketing, companies can send an introduction of a product or a service through an e-mail followed by the telemarketing call to evaluate a person's desire of involvement and lastly send a salesperson to visit [3].

K. L. Kashyap · N. Poorav Desai (✉) · A. Sandilya · S. Patel
VIT Bhopal University, Bhopal, Madhya Pradesh, India
e-mail: nishq.p2018@vitbhopal.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_23

1.2 Problem Statement

When extending credit or loans to clients, banks and financial institutions are frequently confronted with the difficulty of determining which risk indicators to evaluate. Several client characteristics/attributes are typically considered; however, the majority of these characteristics have limited predictive value in determining whether or not a customer is creditworthy. Furthermore, many banks still have a huge difficulty in developing a strong and effective automated bank credit risk score that can help in the correct forecast of client creditworthiness. Therefore, it is important to analyze the effectiveness of telemarketing method due to its enormous usage.

1.3 Proposed Solution

This research aims to assist banks in improving the accuracy of selecting the customer profile by deep learning and identifying a group of consumers who are likely to sign up for depositing their money for the long term based on publicly available data collected from the UCI repository. The deep learning and various machine learning techniques have been used for the testing and training of the model, and their respective output is analyzed.

1.4 Novelty of Our Work

This research was successful in accomplishing the result of the highest accuracy among other works we have seen so far. We achieved the accuracy of 96% by optimizing our model by identifying the optimal number of hidden layers required for the DNN model without overfitting and underfitting our machine learning model. We also experimented with another routine machine learning models like SVM and random forest to align the validity on the empirical basis for optimal result.

The organization of this work is given as follows: The literature review for the existing work is described in Sect. 2. The methodology used in this work is described is Sect. 3. The Experimental results obtained from various models are described in Sect. 4 succeeded by conclusions in Sect. 5.

2 Related Work

Different authors have analyzed the effectiveness of telemarking in their work. Selma et al. used a multilayer perceptron (MLP) and the artificial neural network for the prediction of telemarketing success [4]. Ghatasheh et al. presented a cost-sensitive

analysis of bank campaigns by applying ANN and random forest (RF) [5]. The results show that the SVM method using AdaBoost algorithm obtained accuracy is 95.07% and the sensitivity is 91.65% higher than the ordinary SVM approach. Armin Lawi proposed this method using AdaBoost [6]. A divide-and-conquer technique and combination of expert technique and machine learning techniques for the prediction of bank telemarketing are applied by Moro et al. [7]. The highest 82% accuracy is achieved using divide-and-conquer approach. Palaniappan et al. proposed a customer profiling method with classification approach for bank telemarketing [8]. Md Rashid Farooqi evaluated the performance for competency of bank telemarketing prediction using data mining with the predicted accuracy of 91.48% by J48 decision tree [9]. The highest 90.20% accuracy is achieved using the proposed method. Mehrotra et al. proposed a comparative study of using data mining for bank telemarketing data [10]. The highest 80% accuracy was obtained by employing decision tree. Zeinulla et al. published a comparison technique for bank telemarketing prediction [11]. The highest 90.884% accuracy was observed using the RF technique. Koumétio et al. proposed an approach to optimize the prediction of telemarketing target calls by the SVM technique, and the highest 67.23% accuracy is achieved [12]. Tekouabou et al. obtained the highest 95.83% accuracy by applying ANN approach for prediction problems in bank telemarketing [13]. Ilham et al. presented various classification models to determine the likelihood of bank telemarketing and highest 91.07% accuracy is achieved using the SVM technique [14]. Cédric Stéphane Tékouabou Koumétio demonstrated various classification techniques for bank telemarketing to achieve the accuracy of 60.12% [15]. Dr. Prabhakar Nandru investigated the demand for the adoption of telemarketing practices for promoting long-term bank deposits to a potential bank customer and achieved an accuracy of 92.48% using LR model [16].

2.1 Contribution of the Present Work

The DNN-based model is used for the automatic prediction of the bank telemarketing. Data is pre-processed, and feature selection is also done by applying principal component analysis. Various other supervised learning techniques which are random forest, support vector machine, and k-NN are also employed to compare the effectiveness of the deep learning model.

3 Methodology Used

The proposed work consists of various steps which are (1) Data collection, (ii) Preprocessing of data, (iii) Feature selection, (iv) Training of the deep learning model, and (v) Testing of the deep learning model. The workflow of the research work is shown in Fig. 1.

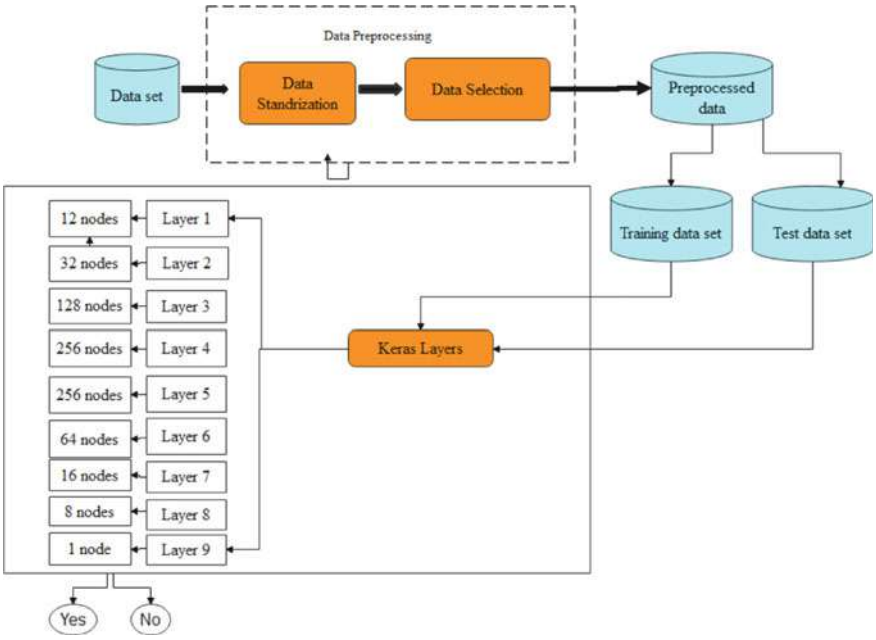


Fig. 1 Graphical abstract and block diagram

3.1 Data Collected

The validation of the proposed work is done by applying telemarketing dataset which is originally collected from the University of California, Irvine. This dataset consists of various attributes collected from 45,211 people's data [17]. The Dataset includes 16 independent variables which are described in Table 1. All the attributes having categorical content is converted into numerical values as the machine learning model do not accept strings.

3.2 Preprocessing

Data standardization is the process of adjusting and rescaling characteristics so that the resultant attribute has a mean of zero and a standard deviation of one. To guarantee uniformity, the dataset standardized using the Z-score technique is implemented.

$$z = {}^{(x-u)}/_{s} \tag{1}$$

The mean and standard deviation of various characteristics are represented by u and s, respectively.

Table 1 Sample records of dataset

Data name	Data type	Description and numeric assigned values
Age	Continuous	Customer age
Job	Categorical	Type of customer job, i.e., unknown, entrepreneur, admin, management, technician, self-employed, services, housemaid, blue-collar, student, retired, and unemployed and possible values assigned as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11, respectively
Marital status	Categorical	Marital status of a customer such as single, married, divorced, and widowed and possible values assigned as 0, 1, 2, 3, and 4, respectively
Education	Categorical	The highest level of education of customer, i.e., unknown, secondary, primary, and tertiary and possible values assigned as 0, 1, 2, 3, and 4, respectively
Default	Boolean	Default customer credit and values assigned as 0 or 1
Balance	Continuous	The average balance
Housing	Continuous	The status of housing loan (true, false)
Loan	Boolean	Client's personal loan (true, false)
Contact	Categorical	Unknown, telephone, and mobile and values assigned as 0, 1, and 2, respectively
Day	Continuous	Last contact month of year
Month	Continuous	Last contact day of year
Duration	Continuous	Last contact duration (in seconds)
Campaign	Continuous	Number of contacts done during this campaign
Pdays	Continuous	How many days before contact has been done with client
Previous	Continuous	How many times client has been contacted
Poutcome	Categorical	Outcome of the previous marketing campaign (unknown, other, failure, success) and values assigned as 0, 1, 2, and 3, respectively
Y	Binary	Target output variable with two values (yes and no) which shows that the client has taken term deposits or not

3.3 Deep Learning Model

Deep learning frameworks are trained using the fundamental sequential model, which is described as a network of Dense Layers. In this study, the Keras deep learning framework is utilized to anticipate telemarketing calls. This model has numerous layers which are sequentially maintained under the equilibrium.

3.4 Keras Layers

Keras is a minimally structured deep learning framework, which enables the creation of deep learning models based on Tensor Flow in a clean and simple manner. Keras

Layers are the basic building blocks of Neural Networks. A layer is made up of tensor-input and tensor-output computing functions, as well as a state that is recorded in Tensor Flow variables.

The model has a total of 9 layers, including 7 hidden levels. Rectified Linear Activation is used in the first eight levels of the seven hidden layers, with 12, 32, 128, 256, 64, 16, 8, and 1 node, respectively. The final layer comprises of a single node with Sigmoid Activation that predicts the output based on whether or not the potential client has deposited the term deposit.

3.5 K-Nearest Neighbors

The k-NN algorithm is a type of unsupervised machine learning technique that may be used to solve both regression and classification problems. When a sample is put into the classifier, it explores the feature space for k examples that are closest to the test samples, since k-NN implies that comparable items occur nearby. The procedure finds the class to which the majority of the k closest samples belong, as well as the class to which the test sample belongs. Depending on the data under observation, a distinct value of k must be assigned. However, the four values which are 3, 4, 5, and 6 are taken as the values of k in this work.

3.6 Support Vector Machine

The support vector machine is a supervised machine learning method that may be used to tackle both categorization and suppression problems. It finds a decision boundary in terms of hyper-plane to classify the data points in an N-dimensional space. Many possible hyper-planes can be obtained for the classification of data points, but the objective of SVM is to achieve the maximum margin of the plane. The definition of the maximum margin is given as the greatest distance between data points of both classes.

3.7 Random Forest

Random forest contains numerous independent decision trees which can be ensemble to get the class prediction. Each independent decision tree of the random forest predicts the class, and the one having the highest votes is considered as the prediction class of the model. Random forest technique uses the Gini index to take decisions on how nodes on a decision tree splits.

$$\text{Gini} = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{2}$$

Here, p_i and c denote the relative frequency of the class obtained in the dataset and the number of classes, respectively. Gini index applies the class and probability to achieve the Gini of each branch on a node.

3.8 Long Short-Term Memory RNN

A RNN-based Sequential model embedded with five distinct layers was implemented, where input layer comprised 64 nodes, 3 hidden layers, and an output layer consisting of a single node for Binary Classification. Here, the Swish function was utilized as the activation function and SGD optimizer was implied, trained under 3 epochs.

4 Experimental Results and Discussion

The Train–Test split approach is used for training and evaluating the deep learning model. The dataset has been split into two parts: a training set of 75% and a testing set of 25%. Keras Layers for Neural Networks were used to forecast whether or not a person would be interested in a term deposit. The parameters for training data example (x^i) and the label (y^i) is given by SGD optimizer which is which is mathematically defined as:

$$\theta = \theta - \eta \nabla_\theta J\left(\theta; x^i; y^i\right) \tag{3}$$

The cross-entropy loss function will be binary in nature applied for binary classification which is mathematically given as:

There are 220 epochs in the training and testing set, each with a distinct batch size and verbosity of 1. Various numbers of dense layers have been used in the experiment. Table 2 shows the results of six deep learning models with various input parameters, output form, and thick layers. The output shape and number of parameters in each layer of sequential Model is shown in Fig. 1.

Accuracy, sensitivity, specificity, precision, and F1-score of the proposed deep learning framework and standard machine learning techniques are compared. A confusion matrix is used to calculate all of the performance matrices, as illustrated in Table 3. When the goal output is positive, the projected outcome can be classified as false negative for two output classes (the customer interested to take term deposit) and the model predicted as negative (the customer not interested to take term deposit). The false positive is defined when the predicted outcome is negative (the customer not interested to take term deposit) and the model is predicted as positive (customer

Table 2 Input parameters taken for training and testing of deep learning model (Input nodes enumeration = 16 and Output node enumeration = 1)

Model number	k-cross validation	Number of dense layers	Epochs	Batch sizes	Number of splits
1	Yes	12/32/128/256/256/64/16/8	70	4	2
2	Yes	12/32/128/256/256/64/16/8	90	24	2
3	Yes	12/32/128/256/256/64/16/8	58	26	2
4	No	12/32/128/256/256/64/16/8	70	14	–
5	Yes	12/32/128/256/256/64/16/8	58	24	2
6	Yes	12/32/128/256/256/64/16/8	80	24	2
LSTM RNN	No	64/512/2048/128	3	2	–

Table 3 The results given by proposed DNN models

	Training accuracy (%)	Testing accuracy (%)	Sensitivity	Specificity	Precision	F1-score
Model 1	95.88	82.49	20	97	59	29
Model 2	95.93	82.84	28	95	58	38
Model 3	**91.04**	**90.34**	**48**	**96**	**61**	**53**
Model 4	95.91	82.42	26	95	56	35
Model 5	95.89	82.27	27	95	55	36
Model 6	95.98	82.42	29	95	55	38
SVM	92.96	73.08	13	98	47	20
RF	89.86	90.21	7	99	54	13
ANN	95.98	81.46	8	99	49	14
3-NN	88.08	88.25	19	97	46	26
4-NN	88.20	88.61	31	98	64	42
5-NN	88.18	88.45	33	92	73	73
6-NN	88.32	88.75	42	91	50	46

The Bold in the row signifies the parameters of the best performing model, which is further stated to be the proposed model

interested to take term deposit). The true negative occurs when the target outcome is negative (the customer is not interested to take a term deposit) and the model is predicted as negative output (the customer is not interested to take a term deposit). The true positive occurs when the target output is positive and the model predicts as positive. The model predicts 0 for negative output and 1 for positive output.

Principle component analysis (PCA) is also applied to select the prominent features from the given features of the dataset. As the plot of PCA is shown in Fig. 2, it is evident from the inclination of the graph presented above that all 16 values obtained must be taken into consideration.

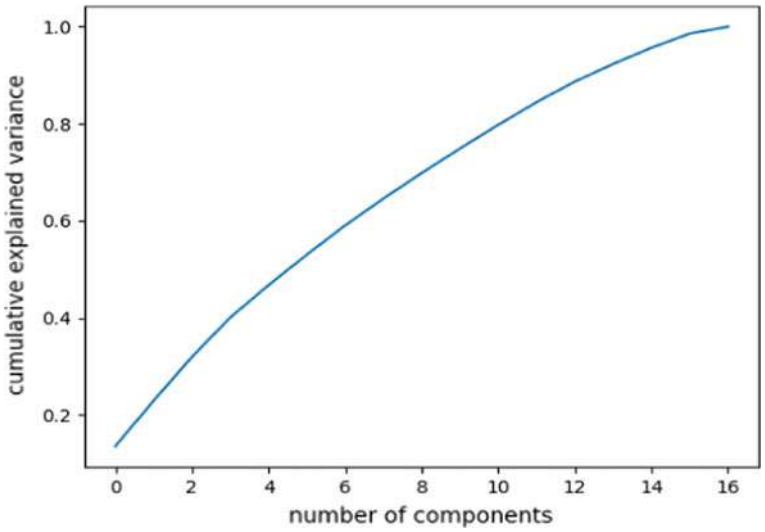


Fig. 2 Plot of PCA

All the features have been used for the experiment. The results obtained with each proposed deep learning models as well machine learning models are analyzed in Table 4. The highest value of 90.34, 91.04, 96, 48, 61, and 53% testing accuracy, training accuracy, specificity, precision, sensitivity, and F1-score is determined via the DNN model 3 (Fig. 3).

The receiver operating curve (ROC) obtained from each proposed deep learning model is shown in Fig. 3. The plot of training loss and accuracy of the deep learning model 3 is depicted in Fig. 4. The comparative analysis of the area under the curve (AUC) and training and testing accuracy is presented in Fig. 7. The comparative analysis of the results of six deep learning models and machine learning models is presented in Figs. 5, 6, and 7, respectively. The combined ROC of machine learning models is also presented in Fig. 8. It is analyzed from Fig. 8 is that the highest 92.86% AUC is achieved by the proposed deep learning model for the prediction of term deposit.

5 Conclusions

The prediction of the success of the telemarketing is presented in this work by applying various deep learning models and machine learning techniques. Research has been performed on an open-source dataset. Preprocessing of data was performed before utilizing it for training and testing of various deep learning and machine learning models. The highest value of 90.34, 91.04, 96, 48, 61, and 53% testing accuracy, training accuracy, specificity, precision, sensitivity, and F1-score is obtained,

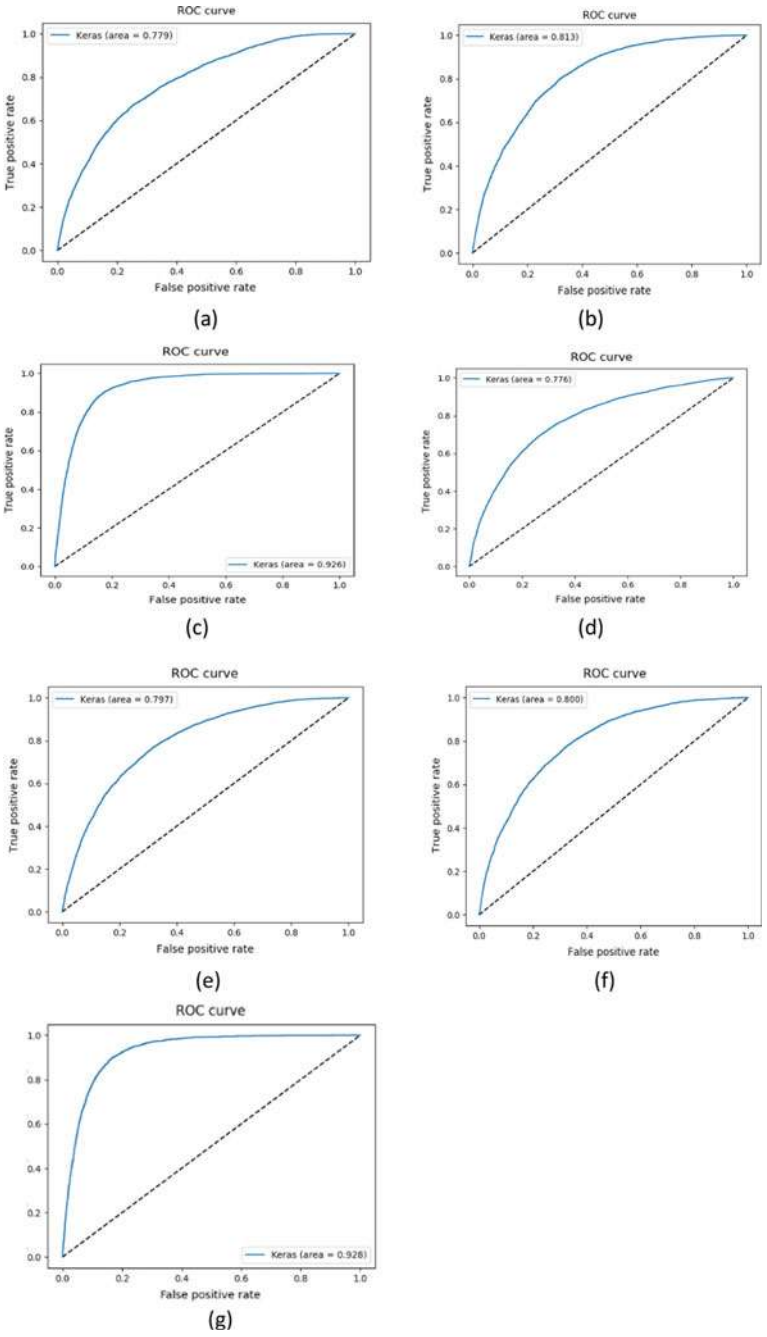


Fig. 3 ROC obtained from various deep learning model **a** Model 1 **b** Model 2 **c** Model 3 **d** Model 4 **e** Model 5, **f** Model 6, and **g** LSTM Model

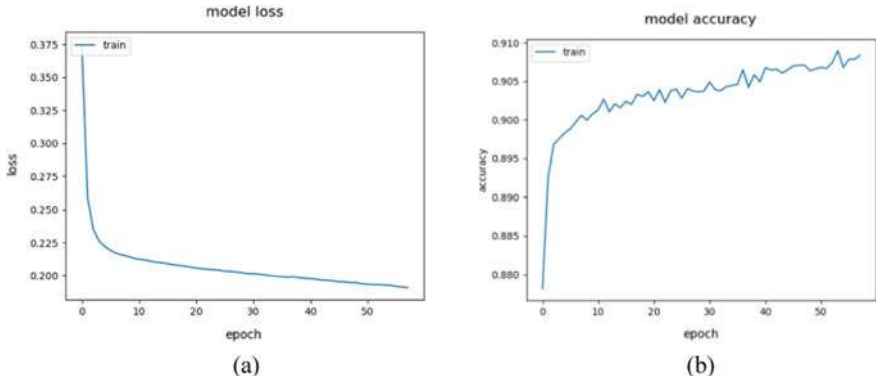


Fig. 4 **a** Training loss, and **b** the accuracy of the proposed deep learning model 3

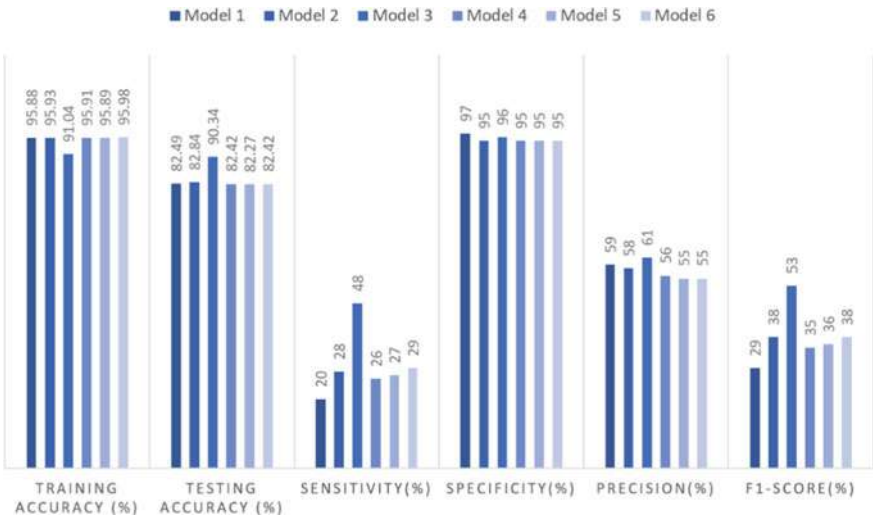


Fig. 5 Comparison of the results obtained from various proposed deep learning models

respectively, by a deep learning model. The largest area under the curve of 92% is observed by the deep learning model which is higher than the AUC obtained from the machine learning approach like SVM, RF, and k-NN. It can be concluded that the DNN model provided the best results out of all the models that have been used in the proposed work for the prediction of success of telemarketing data.

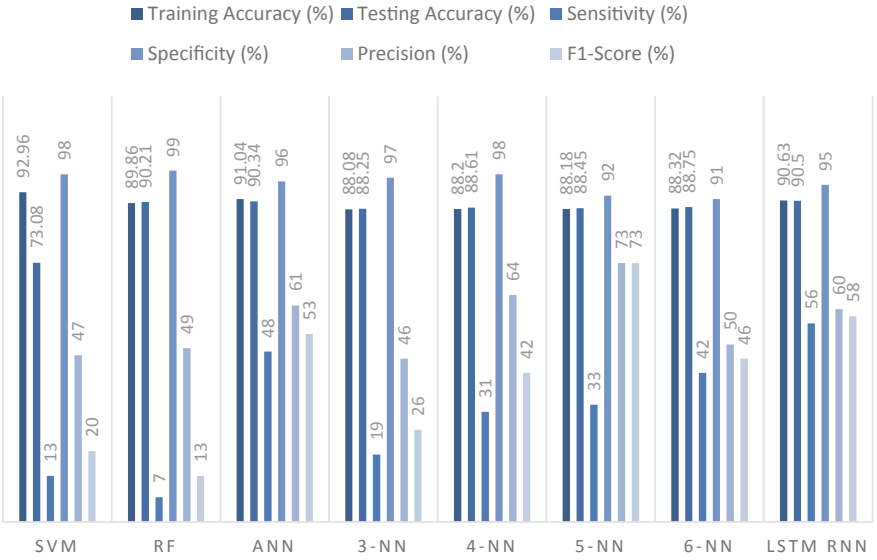


Fig. 6 Comparison of prediction results obtained from machine learning models

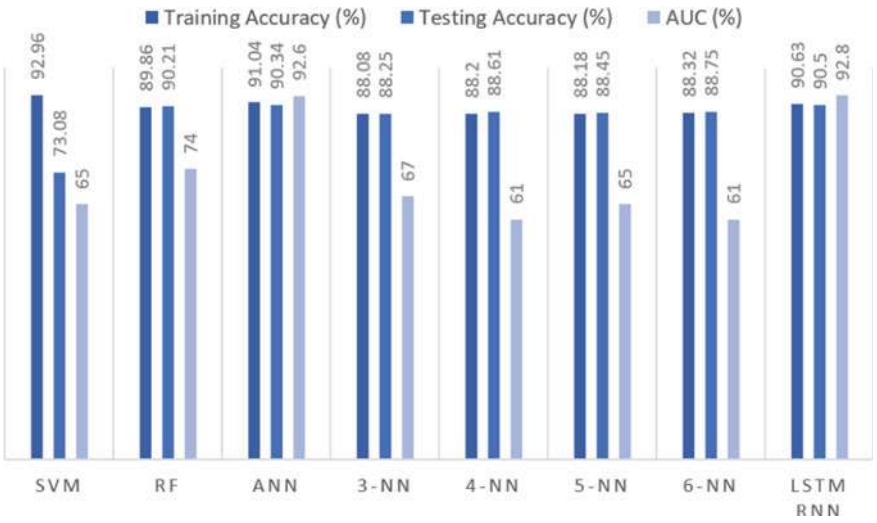


Fig. 7 Comparison of training, testing, and AUC obtained from **a** Proposed six models of deep learning, and **b** machine learning models (RF, SVM, ANN, 3-NN, 5-NN, and 6-NN)

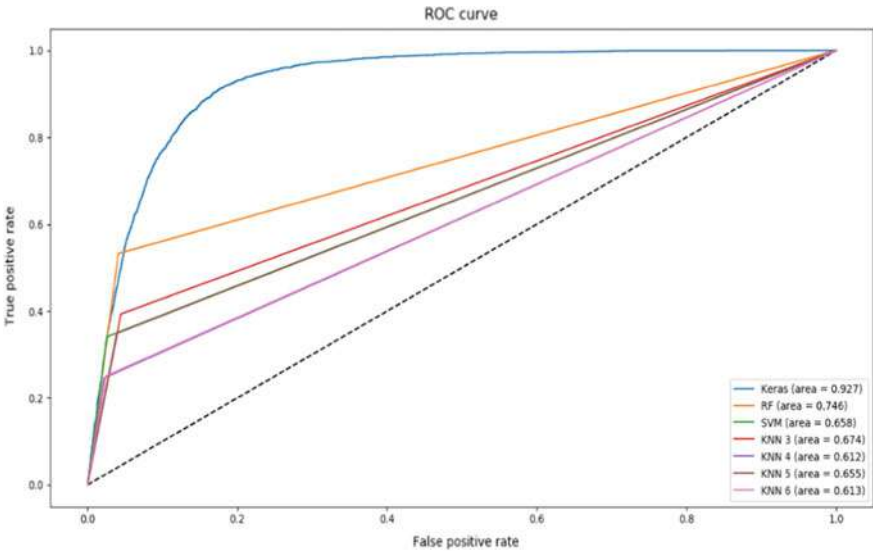


Fig. 8 The combined ROC of different machine learning and deep learning models

References

1. Su CT, Chen YH, Sha DY (2006) Linking innovative product development with customer knowledge: a data-mining approach. Technovation 26:784–795
2. Elsalamony HA, Elsayad AM (2013) Bank direct marketing based on neural network. Int J Eng Adv Technol 2:392–400
3. Vajiramedhin C, Suebsing A (2014) Feature selection with data balancing for prediction of bank telemarketing. Appl Math Sci 8:5667–5672
4. Selma M (2020) Predicting the success of bank telemarketing using artificial neural network. Int J Econom Manage Eng 14:1
5. Ghatasheh N, Faris H, AlTaharwa I, Harb Y, Harb A (2020) Business analytics in telemarketing: cost-sensitive analysis of bank campaigns using artificial neural networks. Appl Sci 10:2581. https://doi.org/10.3390/app10072581
6. Moro S, Laureano RMS, Cortez P (2011) Using data mining for bank direct marketing: an application of the CRISP-DM methodology. In: Proceedings of the European simulation and modelling conference - ESM' 2011, Guimarães
7. Moro S, Cortez P Rita P (2018) A divide-and-conquer strategy using feature relevance and expert knowledge for enhancing a data mining approach to bank telemarketing. Expert Syst. 35. https://doi.org/10.1111/exsy.12253
8. Palaniappan S, Mustapha A, MohdFoozy C, Atan R (2017) Customer profiling using classification approach for bank telemarketing. Int J Inf Visual 1:214–217. https://doi.org/10.30630/joiv.1.4-2.68
9. Farooqi R, Iqbal N (2019) Performance evaluation for competency of bank telemarketing prediction using data mining techniques. Int J Recent Tech Eng 8(2):5666–5674
10. Hassan D, Rodan A, Salem M, Mohammad M (2019) Comparative Study of using data mining techniques for bank telemarketing data. In: 2019 Sixth HCT information technology trends (ITT). IEEE, pp 177–181
11. Zeinulla E, Bekbayeva K, Yazici A (2018) Comparative study of the classification models for prediction of bank telemarketing. In: IEEE 12th international conference on application of

information and communication technologies (AICT), Almaty, Kazakhstan, 1–5. https://doi.org/10.1109/ICAICT.2018.8747086
12. Koumétio CST, Cherif W and Hassan S (2018) Optimizing the prediction of telemarketing target calls by a classification technique. In: 2018 6th international conference on wireless networks and mobile communications (WINCOM), Marrakesh, Morocco, 1–6. https://doi.org/10.1109/WINCOM.2018.8629675
13. Tekouabou CSK, Cherif W, Silkan H (2019) A data modelling approach for classification problems: application to bank telemarketing prediction. In: Proceedings of the 2nd international conference on networking, information systems and security. https://doi.org/10.1145/3320326.3320389
14. Ilham A, Khikmah L, Ulumuddin I, Iswara IBAI (2019) Long-term deposits prediction: a comparative framework of classification model for predict the success of bank telemarketing. IOP Conf Series: J Phys https://doi.org/10.1088/1742-6596/1175/1/012035
15. Koumétio CST, Cherif W, Hassan S (2018) Optimizing the prediction of telemarketing target calls by a classification technique. In: 2018 6th International conference on wireless networks and mobile communications (WINCOM), IEEE, pp 1–6
16. Borugadda P, Nandru P, Madhavaiah C (2021) Predicting the success of bank telemarketing for selling long-term deposits: an application of machine learning algorithms. St. Theresa J Human Soc Sci 7(1):91–108
17. https://archive.ics.uci.edu/ml/datasets/bank+marketing

Chapter 23
Named Entity Recognition Based on Combining Pretrained Transformer Model and Deep Learning

Bui Thanh Hung and Tran Quang Huy

1 Introduction

The named entity recognition (NER) is the process of identifying different entity types such as organization, people, location, currency, time, and percentage expression in the free text. This is an important task in information extraction before moving to higher task in natural language processing (NLP) such as text summarization, text mining, machine translation, and question answering.

There are two main approaches to solve the named entity recognition: rule based and machine learning based [1, 2]. The first approach compares the input data (usually in the form of a feature) with a rule (predefined or autogenerated). The entity is identified based on its results. However, in this approach, it is necessary to have a language understanding to build a set of rules. Therefore, the machine learning-based approach can be considered as an advanced solution to overcome the above shortcomings. The dominant methods are used such as: maximum entropy (ME), hidden Markov model (HMM), and conditional random fields (CRFs). In this approach, feature representation is not only a key problem but also an issue that needs to be addressed. Therefore, deep learning can be considered as an effective solution for automatic feature extraction.

In this paper, we propose combining the pretrained transformer models (BERT and PhoBERT) which have been previously trained with a large dataset in advanced methods that reduce the effort of training the model from scratch with deep learning approach: LSTM, BiLSTM. Finally, to improve the performance of the model, we

B. T. Hung (✉)
Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
e-mail: buithanhhung@tdtu.edu.vn

T. Q. Huy
Faculty of Information Technology, Ho Chi Minh City University of Pedagogy, Ho Chi Minh City, Vietnam

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_24

focus on hybrid deep learning approached based on long shorted-term memory (LSTM) and conditional random fields (CRFs) to achieve higher accuracy.

The rest of this paper is organized as follows. Section 2 discusses some related works. Section 3 describes the proposed model architecture. Section 4 presents the experimental results of NER task in Vietnamese documents. Finally, conclusion and future work are given in Sect. 5.

2 Related Works

The subject of named entity recognition has attracted a lot of attention from researchers around the world for a long time. Over the years, it has achieved positive results and developed gradually with many different languages. In the initial period when the problem appeared, the research focused on building manual rules systems. In recent times, research has focused on applying machine learning methods. In particular, the current outstanding technique to solve the problem entity recognition is deep learning [1, 2].

It also follows the basic three approaches in the research on the problem of named entity recognition in Vietnamese: rule based, machine learning, and deep learning method. This is mainly based on supervised and semi-supervised learning.

Nguyen Cam Tu et al. used CRFs and achieved great results [3]. Anh-Duong Nguyen et al. used sequential CRFs on top of bidirectional LSTM to jointly decode labels for the whole sentence, and they obtained results 94.88% F1-score for Vietnamese NER [4].

Hung Bui Thanh proposed combining pretrained word embedding and automatic syntax features for the bidirectional long short-term memory (BiLSTM) model [5] in Vietnamese NER and got the good results.

In transformer approaches, Nguyen Quoc Dat et al. used PhoBERT with two versions: PhoBERTbase and PhoBERTlarge which are the first public large-scale monolingual language pretrained model for Vietnamese NER with high accuracy: 93.6 for PhoBERTbase and 94.7 for PhoBERTlarge [6].

Different with the previous ones, in this research, we propose a new architecture by combining pretrained transformer models and deep learning approaches in Vietnamese named entity recognition.

3 Methodology

3.1 The Proposed Model

There are three main functions in our proposed model. The pretrained transformer model feeds all the words in the sentence into the encoder at once, which helps to

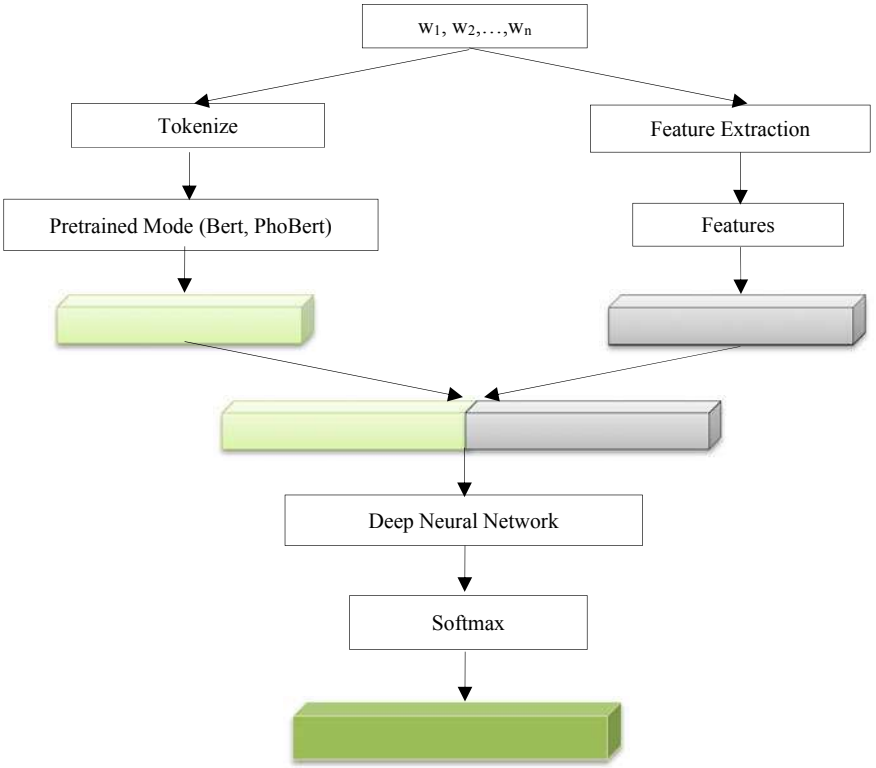


Fig. 1 Architecture of our proposed model

improve training time and preserves the relationships of the words in the encoder. Feature extraction is jointed with pretrained transformer model to produce input of named entity classification. We use deep learning approaches: LSTM, BiLSTM, BiLSTM-CRF layers that take as an input all outputs of the latest transformer encoder from the transformer model as a new model. It can make the best use of each method by combining the approaches together. Figure 1 shows architecture of our proposed model. We will describe each part in detail as follows.

3.1.1 BERT

Bidirectional encoder representations from transformers (BERTs) are a new architecture for the language representation problem announced by Google [7]. BERT is designed to train vectors representing textual languages through their two-dimensional context. As a result, the representation vector generated from the BERT model is fine-tuned with additional output layers, resulting in significantly improved architectures for natural language processing tasks.

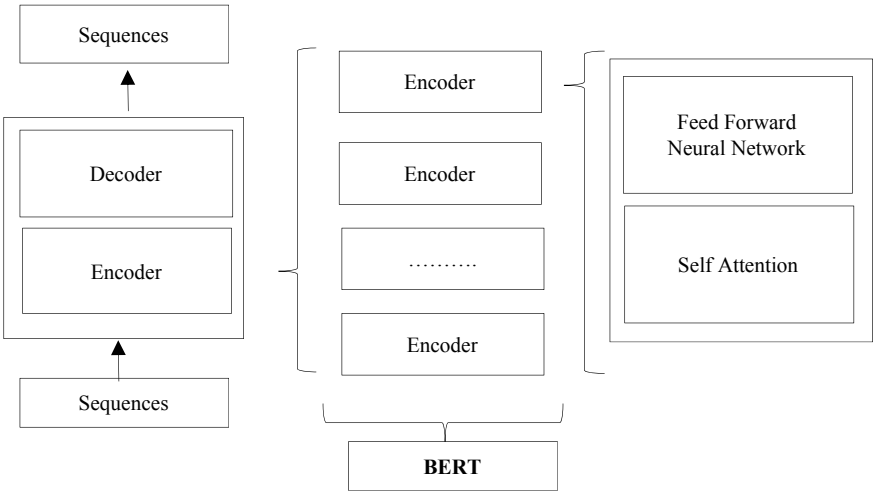


Fig. 2 Architecture of the transformer model

Figure 2 shows transformer architecture. According to Jacob Devlin et al. [7] and Vaswani et al. [8], BERT is a multi-layer bidirectional transformer encoder. BERT uses transformer which is an attention mechanism that learns the correlation between words in a text. Transformer consists of two main parts: encoder and decoder, encoder reads input data, and decoder makes predictions; BERT uses only encoder. The BERT model has two popular sizes: BERTbase and BERTlarge.

Unlike directional models read data sequentially, encoder reads the entire data once, which makes BERT be able to train the data in both directions, through which the model can learn the context of the word better by using the words around it (right, left). Figure 3 shows encoder model.

3.1.2 PhoBERT

PhoBERT is a monolingual language pretrained model which only trained for Vietnamese was built on RoBERT architecture. Figure 4 shows architecture of PhoBERT model.

PhoBERT proposed by Dat Quoc Nguyen et al. [6]. Similar to BERT, PhoBERT also has two versions: PhoBERT base with 12 transformers block and PhoBERT large with 24 transformers block. We use PhoBERT large in our experiments.

PhoBERT uses VnCoreNLP's RDRSegmenter to extract words for input data before passing through the BPE encoder.

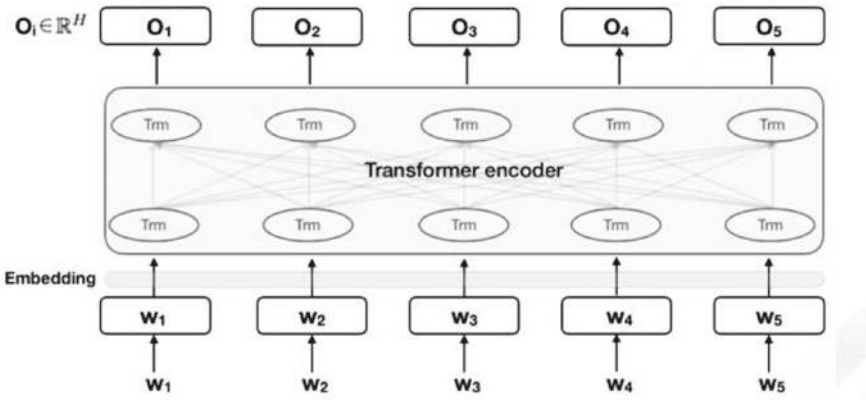


Fig. 3 Encoder model [7]

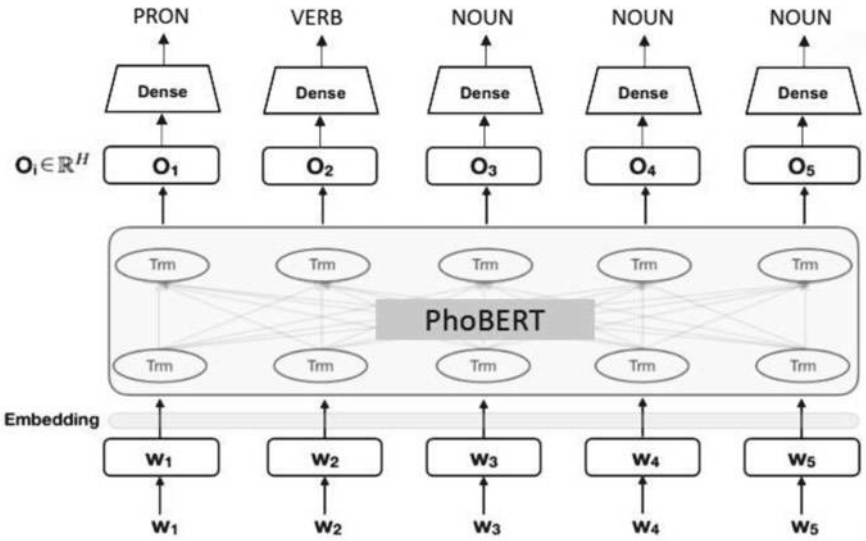


Fig. 4 PhoBERT model [6]

3.1.3 LSTM, BiLSTM

LSTM consists of multiple LSTM memory cells linked together [9–11]. Inside each cell, like the RNN regression network, there are three filter gates: forget gate f_t, input gate i_t, and output gate o_t. What makes LSTM different is the addition of cell internal state throughout the data propagation. Formally, with the input value x_t, the value h_{t-1} of the memory cell output from the previous time step *t-1,* the memory cell is updated at time t as follows:

$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right) \tag{1}$$

$$\tilde{c}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{3}$$

$$i_t = \sigma(W_I h_{t-1} + U_i x_t + b_i) \tag{4}$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{5}$$

$$h_t = o_t \tanh(c_t) \tag{6}$$

On a side note, to improve model performance on sequence classification problems, bidirectional LSTMs, which are an extension of traditional LSTMs, proves to be effective. As it relates to sort of problems where all time steps of the input sequence are available, bidirectional LSTMs train two LSTMs instead of one LSTM on the input sequence. The first on the input sequence as is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even more comprehensive learning on the problem [12–14].

3.2 Named Entity Recognition

With the text vector learned from extracted features and pretrained transformer model as an input of deep learning models to recognize named entity, we use LSTM, BiLSTM, and combination of BiLSTM-CRF to do this task. The process is described in Fig. 5.

4 Experiments

4.1 Dataset

We used the VLSP 2016 dataset for our experiments [15]. A training dataset of 16,858 tagged sentences (extracted from online news) containing 14,918 named entities. The data are saved in text file mainly includes five types of entities under the following topics: location (LOC), organizations (ORG), persons (PER), miscellaneous (MISC), and other (O) named entities. We used IOB notation in the dataset, with each word and five labels consisting of PER, LOC, ORG, MISC, and O; the labeling problem for NER can be reduced to the problem of assigning one label in nine labels (4*2 +

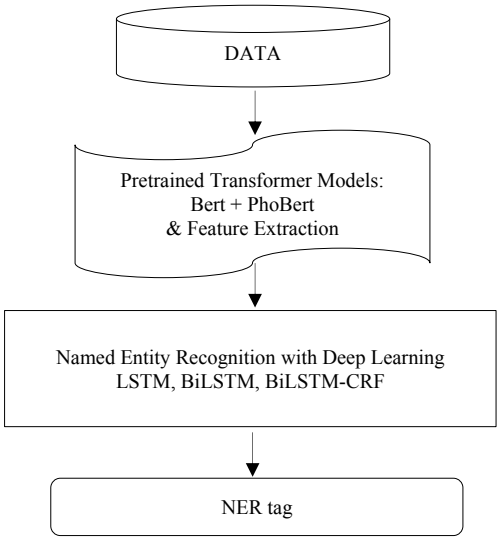


Fig. 5 Named entity recognition process

Table 1 LOC, ORG, PER, and MISC named entity in VLSP 2016 dataset

Entity	Testing	Training
LOC	1379	6245
ORG	274	1213
PER	1294	7480
MISC	49	282

Table 2 Dataset in our experiments

Dataset	Number of sentence
Training	16,861
Testing	2831
Developing	2000

1) for each word. After the preprocessing steps, we divide the data into three sets: training, developing, and testing. Table 1 shows VLSP 2016 dataset in detail, and the dataset in our experiment is shown in Table 2.

4.2 Result

We used Vietnamese Tokenize by Pyvi, chunking by NLTK toolkits (3.3), deep learning tools with Keras [16] and TensorFlow back end [17] are used in our training models. The data after being tokenized will be trained through five models: BERT,

PhoBERT, PhoBERT + LSTM, PhoBERT + BiLSTM, and PhoBert + CRF with the following parameters:

- number of training sessions (epoch): 100
- batch_size: 8
- loss algorithm: categorical cross-entropy
- activate function: Softmax
- optimization function: Adam
- hidden_size: 768
- dropout: 0.3.

To evaluate the effectiveness of the models, we use the F1-score calculated as follows.

$$\text{F1 score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{7}$$

We used the evaluation tool used for the previous systems [18] to calculate the F1-score of NER system. The results of the proposed models are shown in Tables 3 and 4 show the results of the proposed models in IOB tag; then, a comparison of the result of the proposed models with the proposed models in IOB tag is shown in Fig. 6.

From the results in Table 1, it revealed that using PhoBERT is better than BERT in all combination with deep learning approaches. PhoBERT + LSTM is better than PhoBERT + BiLSTM. However, when combining with CRF, PhoBERT + BiLSTM + CRF is the best model with 0.948 F1-score. It is the same results in Table 2 when we analyzed the results in IOB tag. Because PhoBERT is trained in large Vietnamese dataset, PhoBERT combines with deep learning approaches got the better results.

Table 3 Result of the proposed models

Model	F1-score
BERT	0.938
PhoBERT	0.942
PhoBERT + LSTM	0.946
PhoBERT + BiLSTM	0.944
PhoBERT + BiLSTM + CRF	0.952

Table 4 Result of the proposed models with IOB tag

Model	F1-score with IOB tag
BERT	0.921
PhoBERT	0.925
PhoBERT + LSTM	0.930
PhoBERT + BiLSTM	0.928
PhoBERT + BiLSTM + CRF	0.935

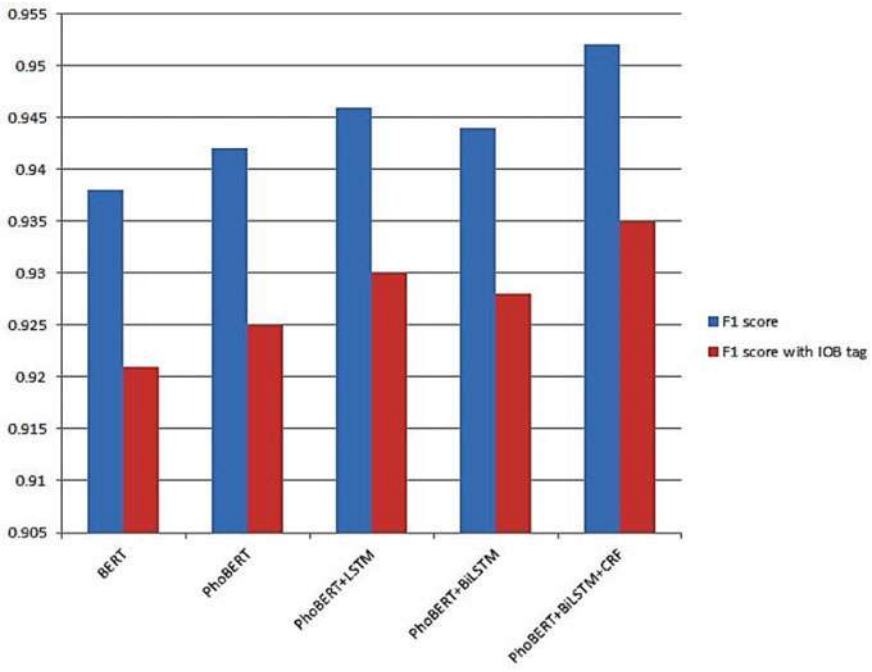


Fig. 6 Comparing the result of the proposed models with the proposed models in IOB tag

From the results in Tables 1 and 2 and Fig. 6, we got the best model which is a combination of PhoBERT + BiLSTM + CRF.

5 Conclusion

In this paper, we proposed a new architecture of combining pretrained transformer model and deep learnings for named entity recognition. We have described our proposed model in details. We also compared the result of our proposed model with the others. Our experiment on VLSP 2016 showed that the best model is combination of PhoBERT + BiLSTM + CRF. In the future, we will aim to incorporate the extracted feature into the model and further tune the hyperparameters of the BERT baseline model and deep learning to improve the results.

References

1. Sharnagat R (2014) Named entity recognition: a literature survey. Center Ind Lang Technol. https://doi.org/10.1023/A:1007558221122

2. Yadav V, Bethard S (2018) A survey on recent advances in named entity recognition from deep learning models. In: Proceedings of the 27th international conference on computational linguistics, pp 2145–2158. arXiv preprint arXiv: 1910.11470
3. Tu NC, Oanh TT, Hieu PX, Thuy HQ (2005) Named entity recognition in vietnamese free-text and web documents using conditional random fields. In: The 8th conference on some selection problems of information technology and telecommunication
4. Nguyen A-D, Nguyen K-H, Ngo V-V (2018) Neural sequence labeling for vietnamese POS tagging and NER
5. Hung BT (2021) Combining syntax features and word embeddings in bidirectional LSTM for vietnamese named entity recognition. Further Adv Internet of Things Biomed Cyber Phys Syst
6. Nguyen DQ, Nguyen AT (2020) PhoBERT: pre-trained language models for vietnamese
7. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. NAACL
8. Vaswani A, Shazeer N et al (2017) Attention is all you need. NIPS'17: proceedings of the 31st international conference on neural information processing system, pp 6000–6010
9. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
10. Hung BT (2019) Document classification by using hybrid deep learning approach. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering - LNICST, volume 298, Springer, pp 167–177
11. Hung BT, Semwal VB, Gaud N, Bijalwan V (2021) Hybrid deep learning approach for aspect detection on reviews. In: Proceedings of integrated intelligence enable networks and computing. Springer Series in Algorithms for Intelligent Systems
12. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991
13. Hung BT (2019) Domain-specific versus general-purpose word representations in sentiment analysis for deep learning models. In: Front intelligent computing: theory and applications, Springer, pp 252–264. https://doi.org/10.1007/978-981-32-9186-7_27
14. Hung BT (2019) Integrating diacritics restoration and question classification into vietnamese question answering system. Spec Issue Adv Eng Comput Sci J—ASTESJ, 4(5):207–212. ISSN: 2415-6698
15. Huyen NTM, Luong VX (2016) VLSP 2016 shared task: named entity recognition. In: Proceedings of vietnamese speech and language processing (VLSP). https://doi.org/10.15625/1813-9663/34/4/13161
16. Gulli A, Pal S (2017) Deep learning with Keras. Packt Publishing Ltd.
17. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat, S, Irving G., Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X (2016) Tensorflow: a system for large-scale machine learning. Tech Rep Google Brain arXiv preprint
18. Sundheim BM (1995) Overview of results of The Muc-6 evaluation. In: Proceedings of the 6th conference on message understanding, MUC6 '95, pp 13–31, Stroudsburg, PA, USA. Association for Computational Linguistics. https://doi.org/10.3115/1119018.1119073

Chapter 24
Electronic Dance Music Sub-genre Classification Using Machine Learning

Chetan Popli, Advait Pai, Vijayetha Thoday, and Manisha Tiwari

1 Introduction

Electronic Dance Music (EDM) is a compilation of percussive electronic music genres which generally consist of a repetitive beat and a synthesized backing track, largely produced for clubs and music festivals. Since the 1970s, EDM has evolved progressively with artists interpreting the genre according to their understanding and pushing the envelope, resulting in sounds which are heavily influenced by their cultural background. This meteoric rise in popularity of EDM across the globe led to the inception and development of several sub-genres, since it would be a seemingly impossible task to classify such an abundant collection of sounds as EDM alone. As per Ishkur's Guide to Electronic Music [1], there are 166 sub-genres of EDM including some prominent ones such as House, Dubstep, Trance and Psy-trance which are popular among EDM enthusiasts.

The abundance of sub-genres under EDM makes the task of sub-genre identification and differentiation slightly difficult, especially if the sub-genres are obscure and the differences between them are nuanced. Industry professionals, DJs and even beginners tend to segregate songs in their library sub-genre-wise to have a coherent and homogeneous playlist for harmonious transitions between two songs. In today's day and age, there is an abundance of metadata related to songs available freely on the internet from sources such as Spotify, Tunebat, Beatport and more. In the following sections, we make an attempt to classify EDM songs, using metadata from Spotify, into their respective sub-genres using machine learning.

C. Popli (✉)
Department of Electronics and Telecommunication Engineering, Mukesh Patel School of Technology Management and Engineering, Narsee Monjee Institute of Management Studies, Mumbai, India
e-mail: chetan.popli2000@gmail.com

A. Pai · V. Thoday · M. Tiwari
Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering, Narsee Monjee Institute of Management Studies, Mumbai, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_25

2 Related Work

Research work in song genre classification [2–12] is done on songs using the GTZAN Database. The GTZAN Database [2] is a song database containing 30-second audio clips from 10 genres, with 100 songs from each genre. Using these audio clips, research work has been performed to extract features which are then used to perform the classification. An important class of features called Timbral Texture Features, such as Zero Crossing Rate, Spectral Centroid, Spectral Contrast, Spectral Bandwidth, Spectral Rolloff and Mel-Frequency Cepstral Coefficients, has been majorly used [3, 4, 9–11, 13], to pass into machine learning models to perform classification. Other works have used different features such as Fast Fourier Transform Coefficients [5], Short-Time Fourier Transform Spectograms [7] and Daubechies Wavelet Coefficient Histograms (DWCH) [12]. In [14], classification was performed for MIDI files of 6 genres, using features like melodic intervals, instruments, note extension, time changes and instrument classes and drumkits.

Once the features have been extracted, researchers have used machine learning models such as support vector machines (SVM) [9, 13, 14], AdaBoost [5], Gaussian mixture models [10, 12], K-nearest neighbours [3, 4, 12, 15], naive Bayes [4, 15], decision tress [4] and random forest [4] to perform the classification. In [16], transfer learning using convoluted neural networks which is based on the concept that a neural network trained in the source task can be reused in the target task after adapting the network to a more specific dataset is used to perform classification.

The main aim of the research is to perform classification of the sub-genres of EDM. In [17], classification of EDM sub-genres such as Dubstep, Trance, Electro-House and Techno has been performed. In their work, they use a backpropagation algorithm and a probablistic algorithm to demonstrate the classifier. Our proposed methodology is to develop a classification model which uses the publicly available metadata of songs as its features. In [18], metadata in the form of lyrics, audio previews and album artwork has been used to perform song classification. In [19], a dataset containing metadata features of songs available on Spotify is used to test different feature set combinations. Using SVM-Radial Basis Function, they check which feature set yields the highest accuracy results.

3 Proposed Methodology

3.1 Data Collection

We use the Spotify Web API [20] to create a dataset of the song metadata in the .csv format. Spotipy is a library in Python used to extract the metadata of the songs. We compiled a list of 93 playlists containing songs from the above-mentioned sub-genres. The chosen playlists were curated by individuals and various record labels, containing songs from only a specific sub-genre and were homogeneous. From these

playlists, we extracted a song list of each of these sub-genres and tagged them to the respective sub-genre they belong to. A total pool of 34,500 songs was created. With the song list ready, we extracted metadata for each individual song and created a dataset containing the features for songs as well as the respective sub-genres. The metadata for each song contains song features such as danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo and other metadata like mode, type, id, uri, track_href, analysis_url, duration in milliseconds and time_signature.

3.2 Data Cleaning and Pre-processing

To clean our dataset, first we removed all null entries and duplicate entries. Then, we removed all songs which were not falling in the 100–1000 second duration range and songs where the energy parameter was less than 0.3, since EDM is a very energetic music genre. To maintain a uniform tempo range of 100–200 beats per minute (BPM) for all the songs, we multiplied all tempos below 100BPM by 2 and divided all tempos above 200BPM by 2. After this step, we further filtered the songs on the basis of their tempo by giving a specific tempo range for each individual sub-genre. The final dataset had 3,000 songs from each sub-genre, thereby having a total number of 15,000 songs.

For pre-processing, we dropped metadata which are not song features as mentioned above. After this, we scaled the data using a min-max scaler to a range of 0–1. Lastly, we used a 80:20 train-test split on the dataset.

3.3 Genre Description

The dataset we are using to perform the classification has songs from the genres: House, Drum and Bass, Techno, Hardstyle and Trap. In this section, we briefly describe each genre, along with the minimum, maximum and mean values of some key features of all selected songs from each sub-genre. Tables 1, 2, 3, 4 and 5 display the values of key song features before scaling was performed, as mentioned in Sect. 3.2.

Table 1 Feature values for house songs

	Tempo	Liveness	Acousticness	Energy	Danceability
Min	115.000	0.012	0.000	0.303	0.368
Max	134.998	0.967	0.844	0.999	0.988
Mean	125.005	0.145	0.016	0.822	0.784

Table 2 Feature values for techno songs

	Tempo	Liveness	Acousticness	Energy	Danceability
Min	120.000	0.022	0.000	0.307	0.308
Max	149.999	0.966	0.861	1.000	0.991
Mean	129.751	0.149	0.036	0.812	0.706

Table 3 Feature values for Drum and Bass songs

	Tempo	Liveness	Acousticness	Energy	Danceability
Min	160.001	0.015	0.000	0.370	0.124
Max	183.961	0.985	0.647	1.000	0.907
Mean	173.774	0.210	0.018	0.881	0.531

Table 4 Feature values for Hardstyle songs

	Tempo	Liveness	Acousticness	Energy	Danceability
Min	140.028	0.015	0.000	0.371	0.106
Max	164.996	0.982	0.905	0.999	0.939
Mean	151.154	0.277	0.040	0.900	0.480

Table 5 Feature values for Trap songs

	Tempo	Liveness	Acousticness	Energy	Danceability
Min	135.000	0.017	0.000	0.346	0.163
Max	164.942	0.975	0.834	1.000	0.965
Mean	147.478	0.264	0.029	0.891	0.594

1. *House*: House music usually uses a 'four-to-the-floor' bass drum characterized by drum rolls, off-beat hi-hat patterns and synthesizer riffs, often played in a tempo range of 120–130 BPM.
2. *Techno*: Techno has drum patterns similar to House, but uses more purely synthetic sounds. It has very little or no chord movement, few or no vocals at all and is played between a tempo range of 120–140 BPM.
3. *Drum and Bass*: Drum and Bass has very fast and complex drum patterns. It uses hip-hop breaks and re-triggering drums played at a high tempo. It is characterized by a heavy sub-bass line and usually plays at around 170 BPM.
4. *Hardstyle*: Hardstyle typically consists of deep, hard-sounding kick drums, faded or reversed bass-lines accompanied by de-tuned or distorted sounds. It is usually played between 140–160 BPM.
5. *Trap*: Trap music uses complex hi-hat patterns, synthesized drums with a long decay, atmospheric synths and lyrical verses in between the melody. Conventionally, it is played between a huge range of 100–160 BPM

3.4 Methodology

After collection, cleaning and pre-processing of the data, we use the machine learning algorithms from scikit-learn [21] to classify the test songs into one of the five sub-genres.

1. *Stochastic Gradient Descent (SGD)*: Since our use case requires us to use SGD for the purpose of multi-class classification, we change the loss function to 'log', which acts like a model equivalent to logistic regression that is fitted using SGD instead of any other solver in logistic regression.
2. *Logistic Regression*: Logisitic regression is generally used for binary classification. Since we are carrying out multi-class classification, we use the one-vs-rest or 'ovr' scheme. Additionally, we increased the maximum iterations parameter of the classifier to 10,000 along with L2 regularization.
3. *K-Nearest Neighbours Classifier (KNN)*: In our model, we change the weight function from 'uniform' to 'distance', where the points are assigned a weight by the inverse of their distance. This way, neighbours closer to the concerned data point will be given more importance. This variation of KNN is also called 'Weighted KNN'.
4. *Gaussian Naive Bayes (GNB)*: Gaussian naive Bayes can be used for both binary and multi-class classifications. For the purpose of sub-genre classification, we have not changed any parameters of the model and have used it as is.
5. *Linear Discriminant Analysis (LDA)*: We have used the LDA algorithm as it is without making any changes to the default parameters.
6. *Support Vector Classifier (SVC)*: In our model, we have changed the gamma value for training, which indicates how far the influence of a single training example reaches, to 'auto'. Along with this, we have also changed the probability parameter to 'True', which returns a vector of probabilities belonging to each class of the response variable, for each prediction.
7. *Random Forest Classifier*: In our tree model, we change the following parameters: maximum depth (max_depth) of the tree to 25, the minimum samples per leaf node (min_samples_leaf) to 1, minimum number of samples required to split an internal node (min_samples_split) to 5 and the number of trees in the forest (n_estimators) to 1200.
8. *Extremely Randomized Trees Classifier (ExtraTrees Classifier)*: ExtraTrees classifier is similar to random forest classifier and differs only in how the trees are constructed in the forest. We have not made any changes to any parameters in this algorithm since it already works on multiple subsets of the dataset.

4 Results and Discussion

After performing the classification, we get F-scores ranging from 0.832 to 0.913 and accuracy from 0.833 to 0.913 as shown in Table 6. Below, we discuss the result of

Table 6 Accuracy, precision and F-score for all algorithms

Model name	Accuracy	Precision	F-score
Stochastic gradient descent	0.833000	0.838115	0.831592
Logistic regression	0.845000	0.846077	0.844126
K-nearest neighbours	0.861667	0.863484	0.861464
Gaussian naive Bayes	0.868333	0.869960	0.868455
Linear discriminant analysis	0.874000	0.876276	0.873615
Support vector classifier	0.875333	0.876727	0.875301
Random forest classifier	0.911333	0.911829	0.911324
ExtraTrees classifier	0.912667	0.913458	0.912610

the classification models having the (i) lowest F-score, (ii) median F-score and (iii) highest F-score along with the confusion matrix.

4.1 Lowest F-score: SGD-Classifier

Stochastic gradient descent has the lowest F-score of 0.832, having F-score similar to that of logistic regression. Figure 1 is the confusion matrix of the true versus the

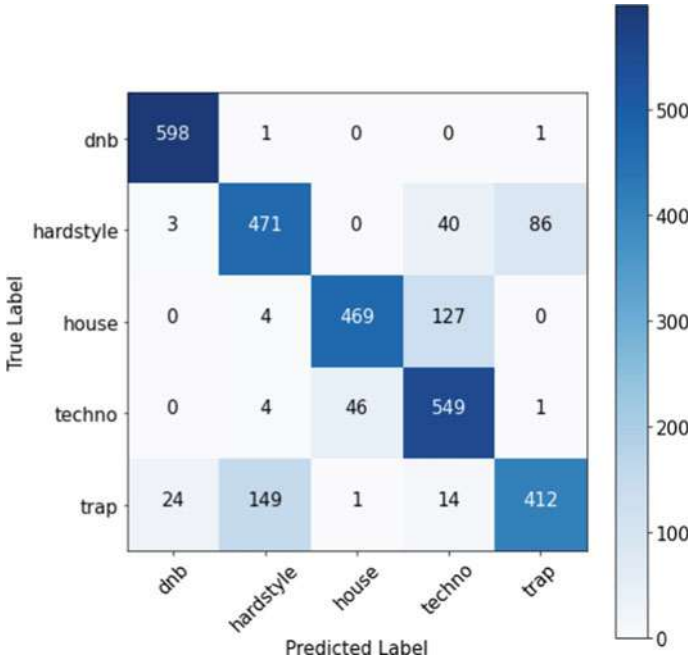


Fig. 1 Confusion matrix for stochastic gradient descent

predicted labels of the songs classified using stochastic gradient descent. About 127 songs of House have been predicted as Techno and 46 vice-versa. Similarly, Hardstyle and Trap have also been misclassified. Then, 149 songs of Trap have been predicted as Hardstyle and 86 vice-versa. Trap songs have been least correctly classified with only 412 of those being correctly predicted.

4.2 Median F-score: Linear Discriminant Analysis

In comparison to stochastic gradient descent, the linear discriminant analysis shows a higher F-score of 0.874, and there is reduction in Trap being misclassified as Hardstyle, i.e. from 149 to 133 and from 86 to 53 vice-versa as shown in Fig. 2. The number of misclassified songs between House and Techno is also high. In SGD, a total of 173 misclassifications occur between House-Techno and vice-versa. In LDA, this number is reduced to 152, but the number of Techno songs being predicted as House has increased from 46 to 68. Overall, a significant increase in Trap songs being correctly predicted is observed, i.e. from 412 to 457. K-nearest neighbours classifier, Gaussian naive Bayes and support vector machine classifier have similar F-score, accuracy and precision.

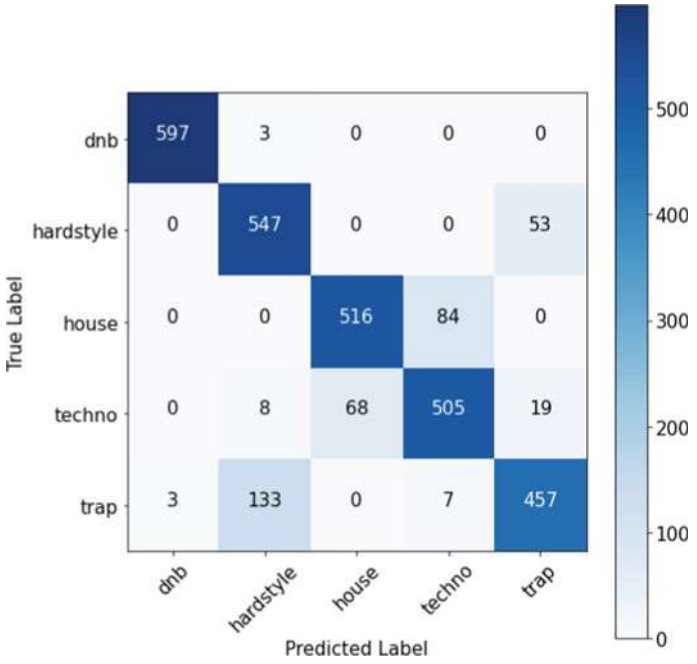


Fig. 2 Confusion matrix for linear discriminant analysis

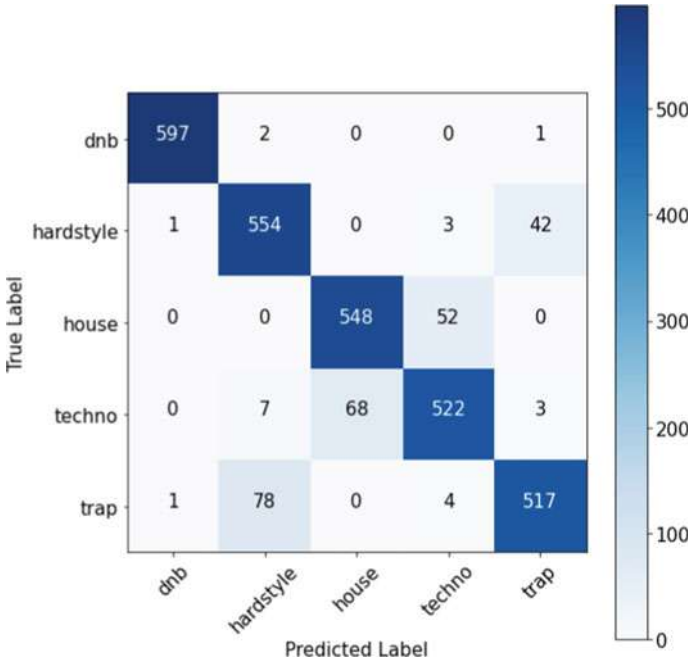


Fig. 3 Confusion matrix for ExtraTrees classifier

4.3 Highest F-score: ExtraTrees Classifier

Forest classifiers are known to employ bagging as an ensemble method to increase the prediction accuracy. The ExtraTrees classifier model yields the highest F-score of 0.913. Figure 3 is the confusion matrix of the true versus the predicted labels of the songs classified using ExtraTrees classifier. As compared to LDA, there is substantial reduction in the misclassification of Trap songs as Hardstyle, i.e. the number is reduced from 133 to 78. The misclassification of House as Techno is also reduced from 84 to 52. Surprisingly, the number of Techno songs incorrectly predicted as House remains the same between LDA and ExtraTrees classifier. This could be due to the songs having closely related feature values.

4.4 House-Techno and Hardstyle-Trap Misclassification

After comparing the results of the three models, it is observed that House-Techno and Hardstyle-Trap are the two pairs that are often misclassified. To find the underlying cause for this misclassification, we compare the mean values of every feature, for all songs of the aforementioned pairs.

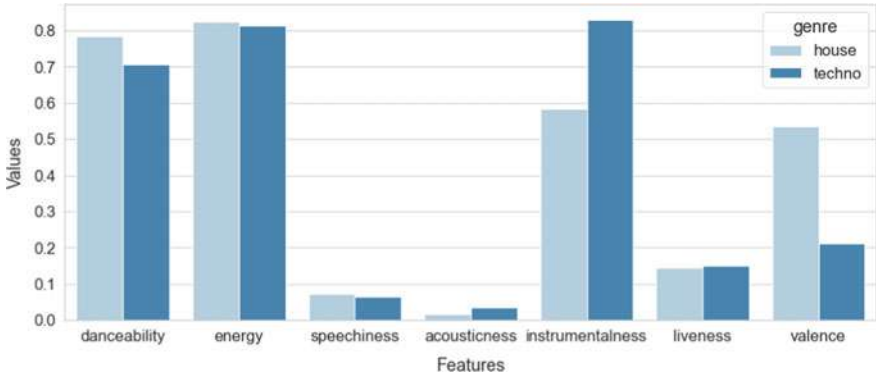


Fig. 4 Feature comparison between House and Techno

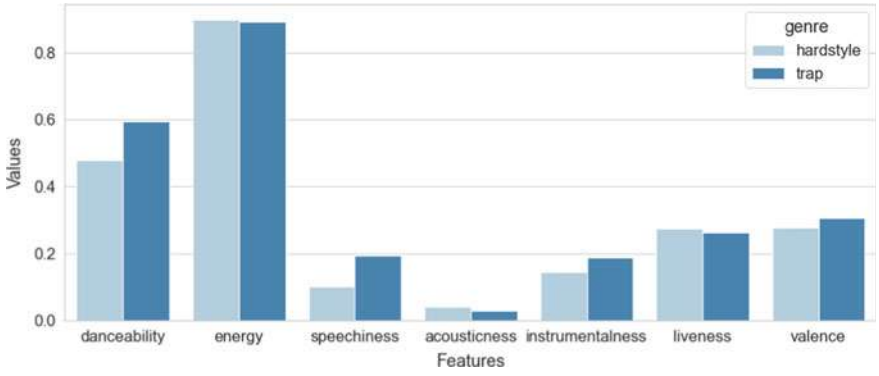


Fig. 5 Feature comparison between hardstyle and trap

In Fig. 4, it is observed that for features = ['energy', 'speechiness', 'liveness', 'acousticness'], the mean values are very close for the House-Techno pair. Additionally, 'danceability' is relatively close for the two sub-genres.

Similarly, in Fig. 5, it is observed that for features = ['energy', 'liveness', 'acousticness', 'valence'], the mean values are very close for the Hardstyle-Trap pair. Additionally, 'instrumentalness' is relatively close for the two sub-genres. This closeness of mean values of the features could have caused the misclassification of songs being observed across the three models discussed in the previous sections.

Comparing Figs. 4 and 5, it is observed that for the House-Techno pair 'valence' and 'instrumentalness' mean values significantly differ, whereas for the Hardstyle-Trap pair, only 'speechiness' significantly differs. This could possibly explain more songs of Hardstyle-Trap being misclassified than House-Techno.

5 Conclusion and Future Scope

As discussed, general research is focused towards extracting features from audio clips to perform classification. With a maximum F-score of 0.913, we are able to demonstrate that using the metadata of a song, we can perform classification on EDM sub-genres. We believe our approach will be less time-consuming and requires less computational power.

Today, most playlists are human-curated by avid listeners of EDM. Using machine learning algorithms for classification, song streaming platforms could automate the process of developing specially curated playlist for songs for sub-genres as opposed to classifying songs under a broad umbrella of EDM.

References

1. Ishkur's Guide to Electronic Music http://music.ishkur.com (2019)
2. Tzanetakis G, Cook P (2002) Musical genre classification of audio signals. IEEE Trans Speech Audio Proc 10(5):293–302
3. Karatana A, Yildiz O (2017) Music genre classification with machine learning techniques. In: 2017 25th Signal processing and communications applications conference (SIU)
4. Elbir A, Çam HB, Iyican ME, Öztürk B, Aydin N (2018) Music genre classification and recommendation by using machine learning techniques. In: 2018 Innovations in intelligent systems and applications conference (ASYU)
5. Bergstra J, Casagrande N, Erhan D, Eck D, Kégl B (2006) Aggregate features and AdaBoost for music classification
6. Gupta R, Yadav J, Kapoor C (2021) Music information retrieval and intelligent genre classification. In: Proceedings of international conference on intelligent computing, information and control systems
7. Yu Y, Luo S, Liu S, Qiao H, Liu Y, Feng L (2020) Deep attention based music genre classification. Neurocomputing
8. Nanni L, Costa YM, Lumini A, Kim MY, Baek SR (2016) Combining visual and acoustic features for music genre classification. Expert Syst Appl
9. Poria S, Gelbukh A, Hussain A, Bandyopadhyay S, Howard N (2013) Music genre classification: a semi-supervised approach. In: Mexican conference on pattern recognition
10. Bağcı U, Erzin E (2005) Boosting classifiers for music genre classification. In: International symposium on computer and information sciences
11. Barreira L, Cavaco S, da Silva JF (2011) Unsupervised music genre classification with a model-based approach. In: Portuguese conference on artificial intelligence
12. Li T, Ogihara M, Li Q (2003) A comparative study on content-based music genre classification. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval
13. Al-Tamimi AK, Salem M, Al-Alami A (2020) On the use of feature selection for music genre classification. In: 2020 Seventh international conference on information technology trends (ITT)
14. Basili R, Serafini A, Stellato A (2004) Classification of musical genre: a machine learning approach. In: ISMIR
15. Chaudhary D, Singh NP, Singh S (2018) Genre based classification of Hindi music. In: International conference on innovations in bio-inspired computing and applications
16. Liang B, Gu M (2020) Music genre classification using transfer learning. In: 2020 IEEE conference on multimedia information processing and retrieval (MIPR)

17. Baruah T, Tiwari S (2014) Intelligent classification of electronic music. In: 2014 IEEE international symposium on signal processing and information technology (ISSPIT)
18. Dammann T, Haugh K (2017) Genre classification of Spotify songs using lyrics. Audio Previews, and Album Artwork
19. Rahardwika DS, Rachmawanto EH, Sari CA, Susanto A, Mulyono IUW, Astuti EZ, Fahmi A (2020) Effect of feature selection on the accuracy of music genre classification using SVM classifier. In 2020 International seminar on application for technology of information and communication (iSemantic)
20. Spotipy Web API https://spotipy.readthedocs.io/
21. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Chapter 25
An Impact Assessment of Distributed Generation in Distribution Network

Vivek Saxena, Narendra Kumar, and Uma Nangia

1 Introduction

In the modern electrified era, the electrical energy requirement is increasing day by day to fulfill the growing load demand. Increasing load demand affects the development of a country to be forced toward reducing the load requirement. In addition, centralized power generation has its limitations for the possible extension of electrical energy generation. Distributed Generation (DG) is rising as an alternative to overcome the energy requirement of end users. Moreover, green energy generation and pollutant reduction are also possible with the utilization of renewable energy sources as a DG despite the presence of the intermittent nature of energy production [1].

Consequently, the benefits of DG can be maximized by the optimization of the size and location of DG. The benefits of optimized DG utilization can be summarized in technical benefits, cost benefits, and environmental benefits. An overview of the merits of optimized DG allocation is given in Fig. 1.

Along with numerous benefits, there is also a lot of flaws that are present in terms of power quality parameters and protection system due to the extensive use of DG.

Researchers should not neglect the penetration of DG on the protection system of the distribution network. Along with the power quality parameters, the workflow assessment of the protection system is mandatorily required for the framework of DG incorporation [2].

This paper presents the methods to improvise the voltage regulation in the distribution network despite the presence of DG. Simultaneously, the assessment of protection system modification is also reviewed.

V. Saxena (✉) · N. Kumar · U. Nangia
Delhi Technological University, New Delhi, India
e-mail: vvksaxena_1234@gmail.com

V. Saxena
A.B.E.S Engineering College, Ghaziabad, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_26

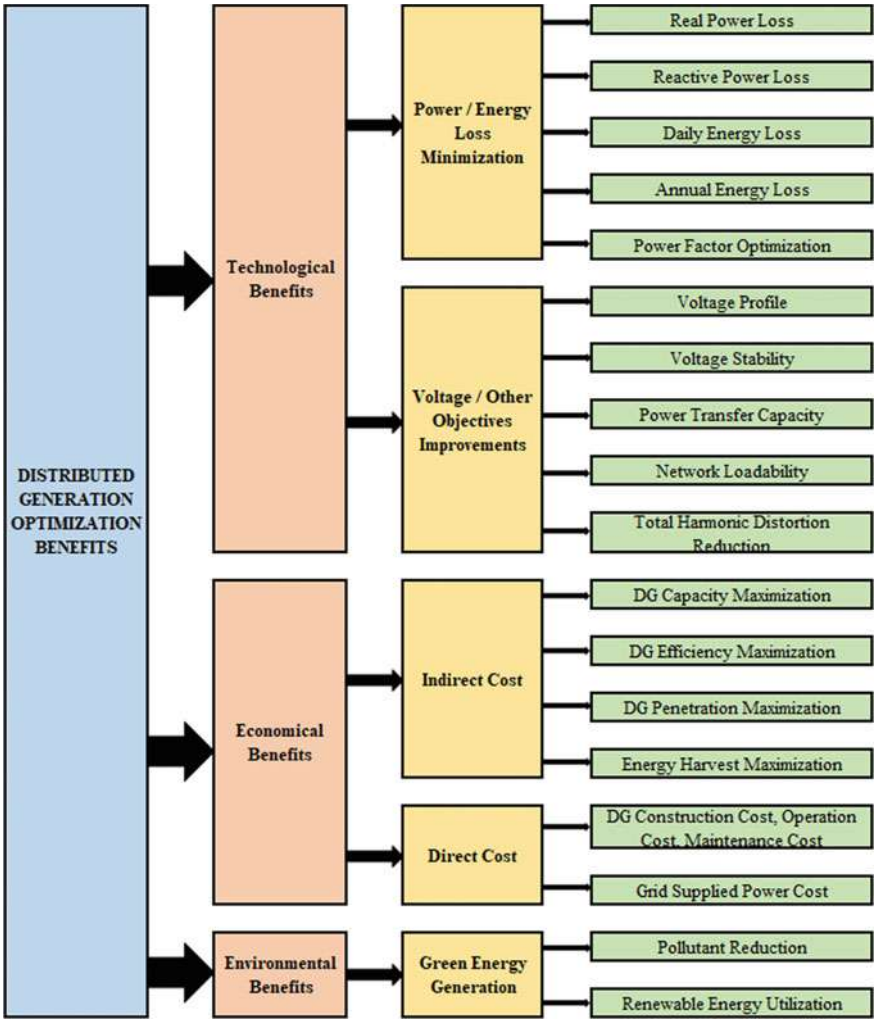


Fig. 1 Numerous benefits of optimized DG integration in distribution network

2 Types of DG

Based on active power and reactive power utilization, DG can be classified into the following categories [3]:

Type 1

Such type of DG provides the real power inoculation to the distribution system by using a suitable interface that consists of fuel cells, photovoltaic, and low-level DG units operated at unity power factor.

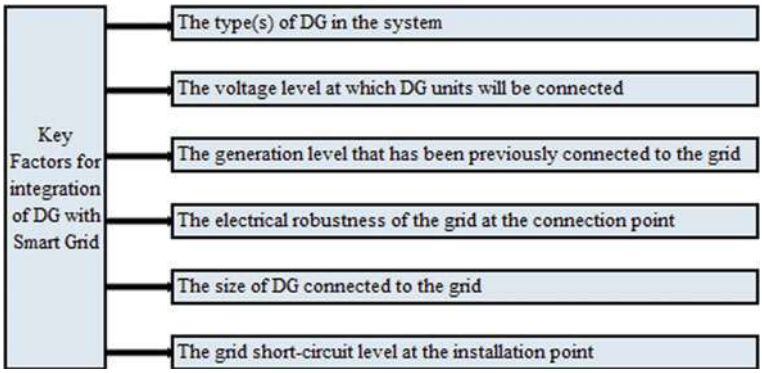


Fig. 2 Key factors for DG assimilation in smart grid

Type 2

Such type of DG injects the reactive power to the distribution systems to compensate the required reactive volt-ampere that includes synchronous compensators operated at nil power factor.

Type 3

Such type of DG behaves as a source and sink at the same time and provides the real power to the electric grid while engrossing the reactive power. Wind energy-based induction generators are mainly used.

Type 4

Such type of DG provides real and reactive power simultaneously and compensates for the complete demand. Gas turbine-based synchronous machines are utilized in this category.

The integration of DG units with the central grid is dependent on various parameters and those are summarized in Fig. 2.

3 Different Features of DG Amalgamation

Examining the effects of DG amalgamation with centralized power generation is required to be reviewed. The issues related to this association are summarized as follows:

3.1 Intermittency of DG

The intermittency of renewable energy-based energy generation is one of the most priority features of DG integration. Renewable energy sources mainly include the solar photovoltaic system and wind power generation which are unpredictable for the availability of input energy. Different types of models have been developed for climatological and atmospheric data prediction.

In rural and remote areas, solar energy is easily available at a nominal cost while the availability of nonrenewable fuel is limited. The Indian government is also encouraging the household persons to adopt this technology by awareness programs and giving easy payment schemes.

The resource availability of wind power generation is uneven, and the size of the smallest unit as compared to a solar cell is also limited. Consequently, the intermittency of wind power generation is more challenging than solar power generation [4].

The intermittency of wind power generation and possible solution for enhancing the reliability, sustainability, and generation have been discussed with the estimation of an average speed of wind [5].

Also, the authors have discussed the possible practical solutions to alleviate the effect of intermittency and it is presented in Fig. 3.

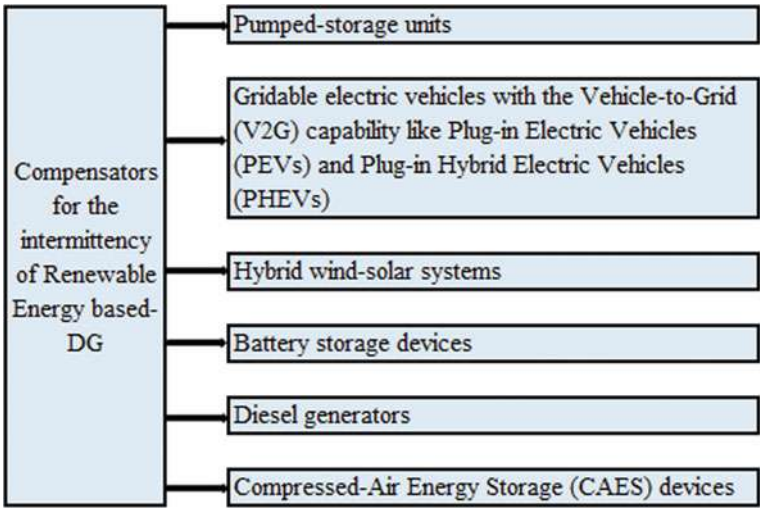


Fig. 3 Possible solutions for the intermittency of renewable energy-based DG

3.2 DG Benefits

As previously mentioned, DG benefits can be categorized in technical, financial, and environmental viewpoints.

Technical benefits: Technical benefits involve the real and reactive power loss reduction, drop in total harmonics distortion, power factor optimization, and voltage profile enhancement.

Financial benefits: Financial benefits cover the direct and indirect cost reduction. Direct cost involves the DG construction, operation, and maintenance cost while indirect cost consists of maximization of DG efficiency, capacity, and harvested energy.

Environmental benefits: Utilization of renewable energy sources, pollutant reduction, and falling dependency on limited fossil fuels.

4 Protection System

The traditional energy generation protection system is comparatively simple and uses re-closers, relays, and fuses due to the presence of a radial distribution system. Due to the high focus on DG technology, there may be a lag in the synchronization of the protection system. Regarding this, the authors suggested an approach to protect the distribution system in the presence of DG. The approach is based on the zone-wise classification of the power network along with the respective risk analysis. Moreover, offline and online data are required followed by finding the fault location after that a restoration signal is being transmitted to operate the load shedding [6].

The integration of DG in the distribution network arises new challenges in the protection system which could be overcome by the addition of emerging protection technology with the existing protection system. Merged protection has been proposed by the authors in which existing and modern protection systems work together for the overcurrent protection in the distribution network. The instantaneous signal is divided into two components, namely limit grading and wavelet energy. The outcomes are very promising for the significant reduction in relay operating time and the effect of the hazardous condition [7].

Active network management and islanded operation are effective tools for the protection scheme of the DG-integrated distribution system. Such a type of protection scheme decreases the magnitude and direction of fault current. In [8], emerging overcurrent protection has been proposed to automatically operate all circuit breakers during abnormal conditions. The comparative assessment of the traditional and proposed scheme has been concluded as the reduction in false signal and operating time of the relay.

The setting of the protection system could be aligned before the integration of DG in the electrical power system so that the proper coordination is taken place. In this

context, the authors proposed an approach for finding the optimal setting parameters of relay protection before the DG amalgamation. The optimization technique is based on the simplex technique to solve the linear programmed formulated problem. The scheme is tested on the radial distribution system of IEEE 13 and IEEE 14 bus system, and the outcomes of this approach are significantly improved as compared to previously applied techniques. The conclusion exhibits that rapid changes in the relay setting are required to avoid the impact of DG penetration on the protection system [9].

Communication plays a key role in the protection unit to neutralize the impact of DG. Additionally, multiagent-based technology has been proposed and executed on Java representative podium to ensure the adequate operation of protection unit in distribution network after assimilation of DG. An independent self-decision maker computer program which is continuously interacted with the physical parameters generates a protocol for the modification in the system. This approach is not only based on the methodology, relay parameters, and operating time rather it takes action after successful coordination with the complete relay family and each agent [10].

The authors suggested and simulated an adaptive fortification-based approach to optimize the overcurrent relay performance after assimilation of DG. Firstly, the reach of a relay is calculated followed by the DG performance under the fault condition then the level of fault current, and this process is concluded by applying the proposed methodology to get the auspicious working condition of a centralized power system having DG. The outcomes are comparatively reliable to operate the system under such circumstances [11].

The protection system must have the capability to rapidly identify the changes in the system after connecting the DG to the power grid. A Fourier transform-dependent fuzzy logic-based protection module has been presented and examined on IEEE 34 bus test system with different case studies. Precise approximation of fault current is done by projected approach, and after that, the value of setting parameters of the relay is labeled via fuzzy logic. The approach is studied on the following test cases: the presence of grid, boosting the DG power, absence of DG, and islanding level of power generation [12].

An adaptive approximation method is simulated by authors to get the optimized setting parameters of the protection system based on relevant observations which are available locally in the system. In this proposed approach, the parameters of Thevenin's equivalent circuit are calculated to guarantee the safe and reliable operation with two-layer relay protection [13].

The authors highly recommended the three-phase converters to mitigate distortion that are occurred in the shape of voltage by eliminating the harmful impact of upper-level harmonics without connecting an individual filtration device. The outcomes are evaluated for renewable energy-based DG [14].

Table 1 presents the catalog of impact assessment of DG on protection system in the distribution network.

Table 1 Catalog of impact assessment of DG on protection system

References	Complication	Remedial methodology	Encounters
[6]	An adaptive over-current protection approach during the integration of DG in the distribution system	The distribution system is divided into numerous areas persisting the characteristic to operate in independent mode	Identify the zone and category of fault when operating with DG
[7]	The protection of power system during overcurrent in the presence of DG	a real-time signal is divided into limit grading and wavelet energy components to identify the fault commencement time	Decreasing the operating time of overcurrent relay during DG operation in the distribution network
[8]	An adaptable over-current protection approach during the integration of DG in the distribution system	Automatic operations of all circuit breakers by smearing the optimized value of real parameters directly	To reduce the numbers of wrong operating actions and also to decrease the average operating time of protection relay
[9]	Arrangement and management of directional overcurrent relay in the presence of DG	The optimization approach is based on the simplex technique to solve the linear programmed formulated problem	Procurement of a group of setting parameters of relay for all types of DG scenarios
[10]	Optimization of overcurrent relay coordination in distribution network having DG	Multiagent-based technology executed on Java representative podium	Coordinated protection by communication arrangement to facilitate the necessary information data
[11]	Overcurrent relay protection in DG-integrated power system	An adaptive fortification-based approach to optimize the overcurrent relay performance after assimilation of DG	Reduced reach or line protection covered by the distance relay
[12]	Protection of DG-based distribution network	Fourier transform-dependent fuzzy logic-based protection module	Apprehending various network procedures and methodologies
[13]	An adaptive over-current protection with multi-level definite time in distribution network incorporated with DG	Calculation of optimized parameters of Thevenin's equivalent circuit based on local parameters	The line-interacted measurement of fault current for various relevant cases

5 Voltage Regulation

Integration of the DG system may lead to various turbulences in the distribution network and challenges arise to operate the system smoothly. The power flow is unidirectional during heavy load hours but it may be bidirectional due to reverse power flow to the grid during light load hours. This excess power generation from DG may cause significant voltage variation and be directed toward poor voltage regulation. In this section, a comprehensive review is carrying to enlighten the challenges and remedies due to the occurrence of voltage limits violation. Poor voltage regulation gives the negative motivation to use DG resources. Certainly, a few parameters may disturb the distribution network in the presence of DG and that can be categorized as:

- The magnitude of the voltage
- The category of distribution network
- The level of power requirement and
- The capacity of DG

5.1 Retrenchment in Grid Power

In the absence of an adequate method of voltage control, the grid operator can separate all the DG from the distribution network and then start to mitigate the poor voltage regulation to continue the reliable operation. Such a method will withdraw all the benefits of DG and renewable energy. Consequently, retrenchment in the active power generation of DGs is the alternative to control voltage variation and facilitate all the features of DG. Regarding this, the authors proposed a voltage droop methodology by controlling the pitch angle of a turbine in wind power generation during poor voltage regulation. Moreover, the approach is practically implemented in small size wind turbines of North America and the results are significantly improved [15].

Active power curtailment is also possible to charge the electric vehicle during high power generations or light load hours in roof-type photovoltaic array incorporated in low voltage distribution network. The authors presented a mutually agreed united method to control the reverse power flow results in better voltage regulation during light load hours in the presence of solar energy-based DG [16].

5.2 Reframing of Distribution Network

The reframing of the distribution network could be done by controlling the operating time of circuit breakers between feeders so that loop operation is taken place. Complete utilization of resources, power loss reduction, and voltage profile enhancement are the advantages of such a reconfiguration. Reframing of distribution network

demonstrated by authors in the presence of DG aiming toward the cost and voltage-level optimization. The methodology is implemented on IEEE 33 and IEEE 392 test system incorporation of various regulators and VAR compensators. However, the results are significantly effective for continuous and discrete voltage regulation models. Additionally, the cost of energy generation is also reduced with the reconfiguration of such a network [17].

The voltage limit violation is mitigated with the mutation of the three-phase generating power model into a mixed-integer second-order cone programming model using different methods. The result outcomes are significantly improved as compared to other methods [18].

5.3 On Load Tap Changer (OLTC)

The voltage level can also be controlled by optimizing the phase angle and leveling the voltage magnitude. OLTC is the most usable equipment for voltage regulation during the presence of DG in the distribution network. OLTC coordinates with other regulating and controlling equipment and upholds the voltage level to neutralize the impact of voltage variation on the power system.

The impact of voltage variation goes worse in single-phase power generation due to the odd distribution of power. To troubleshoot such challenges, the authors presented a coordinated OLTC integrated with solar cells to control the voltage level in the dissociated distribution system. The proposed technology is practically executed on a Danish low voltage system having the following procedural steps: OLTC framing, load framing, active power generation of photovoltaic framing, and controlling of reactive volt-ampere [19].

Few cases of voltage variations may not be handled by only OLTC but it could be improvised to limit the voltage fluctuations within the prescribed limits. In continuation, an administrative control and data acquisition-dependent multi-stage optimization approach are proposed which work with the coordination of OLTC [20].

5.4 Static Synchronous Compensator (STATCOM)

Reactive power compensation is one alternative to regulate the voltage level in the distribution network while DGs are incorporated. This procedure is done by solid state-based devices, i.e., STATCOM, and comes under flexible AC transmission categories. In continuation, reactive volt-ampere compensation is one of the remedial actions that is widely used for optimizing the voltage level.

The authors recommended distributed reactive power compensators to neutralize the voltage variations by controlling the bilateral sequences of voltage. Positive sequence and negative sequence components of voltage are enhanced via concurrent operation of positive sequence admittance and negative sequence conductance,

respectively. Both the sequences are variably aligned to optimize the voltage level with the help of a proportional resonator. The current resonator operates as a regulator to eliminate the harmonic distortion and secure the fundamental component of current [21].

5.5 *Inverter-Controlled DG*

Inverted-based DG units are a promising solution for controlling the voltage variations by delivering or absorbing the reactive power to/from the electric grid. A photovoltaic system is used as a VAR compensator during decoupled grid in hybrid mode. Test module comprises of wind energy-based generator, diesel engine, and inverted-based renewable solar unit. A fuzzy-based proportion integral controller is simulated to get the optimized setting of the unified units. The outcomes are evaluated and compared with other optimization techniques to justify the benefits of such a self-tuned distribution network incorporated with renewable DG [22].

The authors highly recommended the three-phase converters to mitigate distortion that are occurred in the shape of voltage by eliminating the harmful impact of upper-level harmonics without connecting an individual filtration device. The outcomes are evaluated for renewable energy-based DG [23].

5.6 *Energy Demand Management (EDM)*

Smart grid is a futuristic smart approach to modern power systems, and the backbone of such an advanced system is demand management of energy. EDM is an initiative taken by the distribution companies and end users that depends upon the mutual agreement for load shifting and load curtailment. The authors suggested the following key parameters that could be refined for the successful impact of EDM [24]:

- The political inclination is required to attract the participants for energy management, and the participants could be at low-level or mid-level or high-level energy users.
- Blueprint of any policy is essential to move in the right direction within the time limit, and so, the guidelines of EDM are very important to pledge the plan.
- EDM requires a listing of objectives to accumulate each target during the operations, and such a listing may be prepared based on previous experiences and feedback of each stakeholder.
- Financial stability is also a pillar to build a self-sustained system during favorable or unfavorable conditions.
- User participation is the guarantee to run the EDM because it is based on human intelligence.

EDM also deals with voltage regulation in a very frequent manner. The authors offered a methodology, in which the ice thermal storage is converted into the electrical load to control the voltage variations. To demonstrate the impact of EDM, the penetration of DGs in the smart building concept is exhibited. Intermittency of renewable energy sources is also amputated through the adaptiveness of thermal load [25].

5.7 Battery Storage

The penetration of renewable energy sources is suffered due to the intermittent nature of energy generation, and it depends on the climatological and atmospheric condition of a particular geographical area. To neutralize the effect of intermittency, battery storage is the best support by delivering/absorbing the power to/from the electrical grid. Such a real-time controlling of power provides a system with having optimized value of voltage regulation. During high energy generation or light load hours, the charging of the energy storage system is started while the storage system will deliver energy to the grid during low power generation or heavy load hours.

A coordinated consensus approach is recommended for controlling the voltage magnitude within the limits. The proposed method is executed in a low voltage distribution system, and the level of charge is maintained by local controlling devices. The bidirectional converters and voltage booster are used to connect the end users, battery storage, and photovoltaic system in dc link. The methodology is simulated during daily load variation [26].

Table 2 presents the catalog of impact assessment of DG on voltage regulation in the distribution network.

6 Conclusion

DG assimilation in electrical power systems enlightens the path of several benefits in technical, financial, and environmental aspects. Concurrently, DG catalyzes several challenges, primarily protection system, and voltage regulation that may obstruct the high dissemination of DG. Meanwhile, the future of energy generation is mainly dependent upon DG and renewable energy source utilization due to the limited fossil fuel sources. In this paper, a comprehensive review is presented on the engendered encounters in distribution networks post-amalgamation of DG. It is necessary to execute the thorough evaluation of challenges, distresses, and corrective actions to build a more reliable, stable, and capable DG-based electrical power system with high penetration of renewable energy. Variation in the peak value and direction of fault current and voltage limit violation are the main concerns that are elaborated here. Fault current challenges are encountered with the help of substituted methodologies

Table 2 Catalog of impact assessment of DG on voltage regulation

References	Problem	Solution
[15]	Regulation of voltage level in distribution network with the presence of wind energy-based DG	The pitch angle of a wind turbine is optimized according to the requirement of shortening of real power
[16]	To limit the voltage of a rooftop photovoltaic DG during crest load requirement and low level of energy generation	Shortening the real power generation by photovoltaic system and management of energy storage in electric vehicle
[17]	Reframing of unbalanced distribution network having DG to deal with voltage variations	Optimization of setting parameters of various voltage regulating device
[18]	The voltage variation in the distribution network due to intermittency in power penetration of DG during the variable level of load	Assimilation of reactive volt-ampere compensation in a reconfigured distribution network
[19]	Extenuating the voltage-level violation due to solar photovoltaic system and odd distribution of single-phase supply	The controlling of reactive power is done by the OLTC transformer to alleviate the poor voltage regulation. By which all the benefits of DG can be extracted in the distribution network
[20]	Poor voltage regulation due to high penetration of distributed energy	Well communicated and coordinated operation of various integral parts of generation and protection system
[21]	Renewable DG-based distribution networks are required to maintain the voltage magnitude	By using a static reactive power compensator which persists the current regulator
[22]	Voltage regulation in distribution network integrated with renewable and nonrenewable DG	By controlling the reactive power with the help of solid-state inverter DG
[25]	The controlling of voltage and intermittency of renewable energy-based DG in the distribution network	Ice thermal storage is converted into the electrical load to control the voltage variations
[26]	Alleviating the voltage rise challenges of low voltage solar photovoltaic system	Employing battery storage for controlling of state of charge

and proposed relay settings while poor voltage regulation is alleviated via discussed solutions.

References

1. Muthukumar K, Jayalalitha S (2016) Optimal placement and sizing of distributed generators and shunt capacitors for power loss minimization in radial distribution networks using hybrid heuristic search optimization technique. Int J Electr Power Energy Syst 78:299–319

2. Bakr HM, Shaaban MF, Osman AH (2020) Impacts of allocating distributed generation on protection system. In: 6th International conference on electric power and energy conversion system, pp 107–111
3. Hung DQ, Mithulananthan N, Bansal R (2010) Analytical expressions for DG allocation in primary distribution networks. IEEE Trans Energy Convers 25:814–820
4. Saxena V, Kumar N, Nangia U (2021) Smart grid: a sustainable smart approach. In J Phys Conf Ser 2007:012042
5. Saxena V, Kumar N, Nangia U (2021) Protagonist of renewable energy in distributed generation: a review. Technologia en marcha 34(4):3–15
6. Javadian S, Haghifam M-R, Bathaee S, Firoozabad MF (2013) Adaptive centralized protection scheme for distribution systems with DG using risk analysis for protective devices placement. Int J Electr Power Energy Syst 44:337–345
7. Costa FB, Monti A, Paiva SC (2017) Overcurrent protection in distribution systems with distributed generation based on the real-time boundary wavelet transform. IEEE Trans Power Deliv 32:462–473
8. Coffele F, Booth C, Dyśko A (2015) An adaptive overcurrent protection scheme for distribution networks. IEEE Trans Power Deliv 30:561–568
9. Huchel Ł, Zeineldin HH (2016) Planning the coordination of directional overcurrent relays for distribution systems considering DG. IEEE Trans Smart Grid 7:1642–1649
10. Wan H, Li K, Wong K (2010) An adaptive multiagent approach to protection relay co-ordination with distributed generators in industrial power distribution system. IEEE Trans Ind Appl 46:2118–2124
11. Baran M, El-Markabi I (2004) Adaptive over current protection for distribution feeders with distributed generators. In: Proceedings of power systems conference and exposition. IEEE PES, pp 715–9
12. Kumar DS, Srinivasan D, Reindl T (2016) A fast and scalable protection scheme for distribution networks with distributed generation. IEEE Trans Power Deliv 31:67–75
13. Shen S, Lin D, Wang H (2017) An adaptive protection scheme for distribution systems with DGs based on optimized Thevenin equivalent parameters estimation. IEEE Trans Power Deliv 32:411–419
14. Akmal M, Al-Naemi F, Iqbal N (2019) Impact of distributed PV generation on relay coordination and power quality. In: IEEE Milan PowerTech, pp 1–6
15. Chalise S, Atia HR, Poudel B, Tonkoski R (2016) Impact of active power curtailment of wind turbines connected to residential feeders for overvoltage prevention. IEEE Trans Sustain Energy 7:471–9
16. Zeraati M, Golshan MEH, Guerrero JM (2019) A consensus-based cooperative control of PEV battery and PV active power curtailment for voltage regulation in distribution networks. IEEE Trans Smart Grid 10(1):670–680
17. Liu Y, Li J, Wu L (2018) Coordinated optimal network reconfiguration and voltage regulator/DER control for unbalanced distribution systems. IEEE Trans Smart Grid
18. Tian Z, Wu W, Zhang B (2016) Mixed-integer second-order cone programing model for var optimisation and network reconfiguration in active distribution networks. IET Gener, Transm Distrib 10:1938–1946
19. Hu J, Marinelli M, Coppo M (2016) Coordinated voltage control of a decoupled three-phase on-load tap changer transformer and photovoltaic inverters for managing unbalanced networks. Electr Power Syst Res 131:264–274
20. Mehmood KK, Khan SU, Lee S-J (2018) A real-time optimal coordination scheme for the voltage regulation of a distribution network including an OLTC, capacitor banks, and multiple distributed energy resources. Int J Electr Power Energy Syst 94:1–14
21. Lee TL, Hu SH, Chan YH (2013) D-STATCOM with positive-sequence admittance and negative-sequence conductance to mitigate voltage fluctuations in high-level penetration of distributed-generation systems. IEEE Trans Ind Electron 60:1417–1428
22. Anantwar H, Lakshmikantha DBR, Sundar S (2017) Fuzzy self-tuning PI controller based inverter control for voltage regulation in off-grid hybrid power system. Energy Procedia 117:409–16

23. Shalukho V, Lipuzhin IA, Voroshilov AA (2019) Power quality in microgrids with distributed generation. In: International Ural conference on electrical power engineering, pp 54–58
24. Saxena V, Kumar N, Nangia U (2021) Analysis of smart electricity grid framework unified with renewably distributed generation. In: Advances in smart communication and imaging systems, vol 721. LNEE, pp 735–751
25. Luo X, Lee CK, Ng WM (2017) Use of adaptive thermal storage system as smart load for voltage control and demand response. IEEE Trans Smart Grid 8:1231–1241
26. Wang Y, Tan KT, Peng XY (2016) Coordinated control of distributed energy-storage systems for voltage regulation in distribution networks. IEEE Trans Power Deliv 31:1132–1141

Chapter 26
An Iterated Local Search Algorithm for the Degree-Constrained Minimum Spanning Tree Problem

Sudishna Ghoshal and Shyam Sundar

1 Introduction

Given a connected, undirected and edge-weighted graph $G(V, E, w)$, where V is the set of vertices of G; E is the set of edges of G; and for each edge $(i, j) \in E$, there is a positive weight $w(i, j)$ denoting the cost to connect i and j vertices, the degree-constrained minimum spanning tree (DCMST) problem seeks to find a minimum spanning tree (T) subject to each vertex in T has at most d degree, where d is a specified integer.

The DCMST problem is $\mathcal{NP}$-hard for $d > 2$ [1] and has numerous practical applications in the context of backplane wiring among pins in which any pin on the wiring panel cannot wrap by more than a fixed number of wire-ends [2]; in the design of road system that can help in serving the suburbs subject to any crossing cannot be met by more than four roads [3]; in VLSI design, fixing on the number of transistors that can be handled by the output current of a transistor corresponds to the degree constraint for VLSI routing trees [2], in electrical circuits design [4] and in communication networks where the maximum degree in a spanning tree could be measured as vulnerable to single-point failures [5].

The DCMST problem is a well-studied combinatorial optimization problem. Numerous methods based on exact approaches and heuristic approaches have been developed for this problem. Narula and Ho [4] presented a primal and a dual heuristic procedure and a branch-and-bound algorithm for this problem. Savelsbergh and Volgenant [3] proposed two heuristic approaches and a branch-and-bound approach. Later, Boldon et al. [2] presented four problem-specific heuristic approaches based on

S. Ghoshal (✉) · S. Sundar
Department of Computer Applications, National Institute of Technology Raipur, Raipur, Chhattisgarh 492010, India
e-mail: sghoshal.phd2017.mca@nitrr.ac.in

S. Sundar
e-mail: ssundar.mca@nitrr.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_27

Prim's algorithm [6]. Recently, Singh and Sundar [7, 8] also proposed two problem-specific heuristic approaches for this problem.

In addition, several metaheuristic techniques have been developed for this problem. Zhou and Gen [9] proposed a genetic algorithm (GA) that uses prufer encoding. Later, Knowles and Corne [10] proposed a GA that uses a $|V| \times (d-1)$ array encoding. Raidl and Julstrom [11] proposed a GA that uses weight-coding. Later, Raidl and Julstrom [12] further proposed two variants (ES-EA and HES-EA) of evolutionary approach based on edge-set encoding and edge-set encoding as well as incorporation of edge-cost heuristic in the generation of initial solutions, crossover operator, and mutation operator. Bau et al. [13], Bui and Zrncic [14], and Doan [15] worked on ant colony optimization approaches for this problem. Binh and Nguyen [16] and Ernst [17] worked on particle swarm optimization approaches for this problem. Bui et al. [18] developed ant-based algorithm (ABA) for this problem. Bui et al. [18] used two local search methods based on 2-Edge replacement and 1-Edge replacement in order to further improve the fitness quality of the current solution. Recently, Singh and Shyam [19] proposed a hybrid approach (HSSGA) in which a steady-state genetic algorithm is combined with local search strategies. Further, a step based on perturbation mechanism at a fixed interval of generations is added in the replacement strategy of HSSGA in order to have population diversity throughout the search process.

In this paper, we present an iterated local search algorithm (ILS) for the DCMST problem. The performance of the proposed ILS is evaluated on the available 107 benchmark instances, and the obtained results show the superiority of ILS in comparison with state-of-the-art approaches.

The structure of the remaining of the paper is as follows: Sect. 2 presents a brief introduction of an iterated local search algorithm (ILS); Sect. 3 discusses the proposed ILS for the DCMST problem; Sect. 4 discusses the computational results on the available instances; and Sect. 5 presents some concluding remarks.

2 Iterated Local Search Algorithm

This section discusses a brief introduction of iterated local search algorithm (ILS).

Iterated local search algorithm (ILS) [20, 21] is a metaheuristic framework that has been applied successfully for the solution of several $\mathcal{NP}$-hard problems. ILS is relatively simple to implement and uses very few parameters. ILS framework is composed of four components (i) generation of an initial solution (ii) perturbation mechanism (iii) local search and (iv) acceptance criterion. ILS begins with generating a single random initial solution. Hereafter, ILS, at each iteration (generation), consists of performing perturbation on the current solution, leading to an intermediate solution and then, local search is applied on this intermediate solution for improvement in its solution quality. It is assumed that the perturbation mechanism generates a modified solution (an intermediate solution) situated in the basin of attraction of a local minimum that may not be far away from the current solution, but may be better than the current solution. Finally, an additional acceptance of criterion takes the decision which of two solutions to keep in the next iteration (generation).

3 ILS for the DCMST Problem

This section describes an iterated local search algorithm (ILS) for the DCMST problem. Algorithm 1 presents the pseudo-code of ILS for the DCMST problem.

Algorithm 1: Pseudo-code of ILS for the DCMST problem

Input : A connected, undirected and edge-weighted $G = (V, E, w)$;
Output: a degree-constrained minimum spanning tree;

Generate an initial solutions S;
while Termination criterion is not satisfied **do**
  $S'' \leftarrow Perturbation(S)$;
  $S'' \leftarrow LocalSearch(S')$;
  $S \leftarrow AcceptanceCriterion(S, S'')$;

The following subsections discuss the details of various components of ILS for the DCMST problem.

3.1 Generation of an Initial Solution

Since each solution is a spanning tree that consists of $|V|$-1 edges. To encode each solution, we apply an edge-set encoding [12]. In this encoding, a solution is represented as a set of $|V|$-1 edges. Like [19], we also generate an initial solution (say S) randomly based on Prim's algorithm [6]. After that, the generated solution is further improved by applying two local search strategies, i.e., 1-edge replacement $\rightarrow$ 2-edge replacement (see Sect. 3.3 in detail).

3.2 Perturbation

In perturbation, the current solution S is perturbed through perturbation mechanism, leading to a new intermediate solution (S') which is used by the local search for searching a high-quality solution in the search space. Our perturbation mechanism is based on deletion and addition of edges. Deletion and addition of edges procedure start with creating a copy (say S') of the current solution (S). After this, a randomly picked edge (say e_{ij}) is deleted from the tree (T) of S', which in turn, partitions T of S' into two components. An edge (say e_{uv}) different from e_{ij} is searched in G that can join these two components of T of S' subject to degree constraint of T. Once the edge is searched, it is immediately added to T, resulting in a new feasible solution S'. In our perturbation mechanism, with probability P_e, deletion and addition of edges are performed once; otherwise, deletion and addition of edges are performed twice with

probability (1-P_e). Note that we consider a random non-leaf edge as an candidate edge for deletion in case of deletion and addition of edges are performed once.

3.3 Local Search

The local search is used to intensify the search around the perturbed solution (S') in order to find a better solution. We apply two local search strategies (1-edge replacement (say One_ER) $\rightarrow$ 2-edge replacement (say Two_ER)) in a cyclic way on S'. Note that if there is an improvement in the resultant solution (say S'') of S' after applying 2-edge replacement local search, then again 1-edge replacement $\rightarrow$ 2-edge replacement local search strategies are followed.

The idea of One_ER local search is taken from [19]. One_ER consists of two phases which are followed sequentially. In the first phase, All edges of T of S' whose at least one end-point has degree d are kept in a set (say L). Hereafter, at each iteration, an edge (say e_{ij}) is picked randomly from L and an edge (say e_{ab}) $\in E \setminus T$ of minimum cost ($w(a, b) < w(i, j)$) is searched for the replacement of e_{ij} in T subject to the degree constraint of T is maintained. If such e_{ab} exists, then e_{ab} replaces e_{ij} in T. This iteration is continued for $|L|$ times so that all edges in L are picked. Second phase of One_ER is applied for those candidate edges in T except those edges in L. For each candidate edge (say e_{ij}) one-by-one in T, an edge (say e_{ab}) of minimum cost ($w(a, b) < w(i, j)$) $\in E \setminus T$ is searched for the replacement of e_{ij} in T subject to the degree constraint of T is maintained. If such e_{ab} exists, then e_{ab} replaces e_{ij} in T. The rationale of applying first phase of One_ER is to reduce the degree of the vertex which has d degree in T so that the second phase gets larger scope for applying, resulting in further improvement of solution quality of S'.

The idea of Two_ER local search is taken from [18]. Two_ER, at each iteration, an edge (say e_{ij}) is picked randomly and another edge (say e_{ab}) among all non-adjacent candidate edges is picked iff the resultant T gets maximum improvement after replacing e_{ij} and e_{ab} with $e_{ia} \in E \setminus T$ and $e_{jb} \in E \setminus T$, respectively. Readers are suggested to read the detail description of Two_ER local search in [18].

3.4 Acceptance Criterion

The perturbation mechanism along with local search strategies provides the possible transition between the current solution S to S'' for the next iteration. The best one (in terms of fitness) between S and S'' is accepted for the next iteration, and the selected one will be referred to as S. In addition, if there is no improvement in the fitness of the current solution for a specified number of iterations (say $iter$), then the current solution S is replaced by a new solution generated through applying deletion and addition of edges P_{times} times (see Sect. 3.2) on the current solution S followed by local search strategies (see Sect. 3.3). The resultant new solution again becomes

S and is accepted for the next iteration. The role of acceptance criterion is vital as it plays in controlling the balance between intensification and diversification in the course of search process.

4 Computational Results

C language is used to implement ILS. ILS is tested on available 107 benchmark instances for the evaluation of its performance. The tests are executed on Linux-based PC having Intel Core i5 3.2 GHz × 4 processor with 4 GB RAM. Computational results obtained by ILS are evaluated with state-of-the-art approaches for the DCMST problem. In our all experiments with ILS, we have used $P_e = 0.75$, $P_{\text{times}} = 5$ and $iter = 5000$. A large number of trials are used to set empirically the values of all these parameters. Although these parameter values used in ILS give good values on most of the instances, they may not be optimal for all instances.

4.1 Instances

Like [18, 19], our proposed ILS is also tested on 107 benchmark instances for the DCMST problem. These instances are classified into two different groups—Euclidean and non-Euclidean instances. Euclidean instances are composed of three different sets—(i) CRD (ii) SYM and (iii) STR. Non-Euclidean instances are composed of three different sets—(i) SHRD (ii) random hard (R) and (iii) misleading hard (M). The link https://turing.cs.hbg.psu.edu/benchmarks/file_instances/spanning_tree/ can be used to download 107 benchmark instances used for the DCMST problem.

4.2 Comparison of ILS with State-of-the-Art Approaches

In this subsection, we discuss the results of ILS in comparison with the results of state-of-the-art approaches such as HES-EA [12], ABA [18] and HSSGA [19] on 107 instances. Like [12, 18, 19], ILS is also executed for 50 runs on each instance, and is also set for the same termination criterion (in terms of computational time) on the basis of instance size (1 s for $|V| < 100$, 10 s for $|V| = 100$, 60 s for $|V| = 200$, 200 s for $|V| = 300$, 600 s for $|V| = 400$, and 1000 s for $|V| = 500$).

Tables 1, 2, 3, and 4 summarize the results of different data sets discussed in Sect. 4.1. In each table, the first three columns, respectively, present the name of instance ($Instance$), the number of vertices ($|V|$) related to its instance, and the degree constraint (d) related to its instance. The next four columns ($Best$, Avg, SD, and $ATET$), respectively, present the best value, the average solution quality,

standard deviation, and the average total execution time obtained by HES-EA, ABA, HSSGA, and ILS over 50 runs. Results of HES-EA, ABA, HSSGA are taken from [19].

Table 1 summarizes the results of CMD and SYM instances (i.e., 27 instances). In terms of *Best*, ILS is better on 20, and same on 7 instances in comparison with HES-EA; ILS is better on 24, and same on 3 instances in comparison with ABA; ILS is better on 4, same on 21 and worse on 2 instances in comparison with HSSGA. In terms of *Avg*, ILS is better on 22, and same on 5 instances in comparison with HES-EA; ILS is better on 27 instances in comparison with ABA; ILS is better on 24, and worse on 3 instances in comparison with HSSGA.

Table 2 summarizes the results of CMD, SYM, and STR instances (26 instances). In terms of *Best*, ILS is better on 3, and same on 23 instances in comparison with HES-EA; ILS is better on 9, and same on 17 instances in comparison with ABA; ILS is better on 1, and same on 25 instances in comparison with HSSGA. In terms of *Avg*, ILS is better on 10, same on 10, and worse on 6 instances in comparison with HES-EA; ILS is better on 12, same on 10, and worse on 4 instances in comparison with ABA; ILS is better on 5, same on 13 and worse on 8 instances in comparison with HSSGA.

Table 3 summarizes the results on STR and SHRD instances (26 instances). In terms of *Best*, ILS is better on 8, and same on 18 instances in comparison with HES-EA; ILS is better on 10, and same on 16 instances in comparison with ABA; ILS is better on 1, and same on 25 instances in comparison with HSSGA. In terms of *Avg*, ILS is better on 17, same on 8, and worse on 1 instances in comparison with HES-EA; ILS is better on 18, same on 7 and worse on 1 instances in comparison with ABA; ILS is better on 9, same on 16 and worse on 1 instances in comparison with HSSGA.

Table 4 summarizes the results on random hard (R) and misleading hard (M) instances (28 instances). In terms of *Best*, ILS is better on 5, same on 22, and worse on 1 instances in comparison with HES-EA; ILS is better on 6, same on 21, and worse on 1 instances in comparison with ABA; ILS is better on 1, same on 26 and worse on 1 instances in comparison with HSSGA. In terms of *Avg*, ILS is better on 9, same on 9, and worse on 10 instances in comparison with HES-EA; ILS is better on 13, same on 7, and worse on 8 instances in comparison with ABA; ILS is better on 6, same on 10 and worse on 12 instances in comparison with HSSGA.

4.3 Overall Picture

This subsection discusses an overall picture of results of ILS against the results of HES-EA [12], ABA [18], and HSSGA [19] approaches on 107 benchmark instances. Table 5 summarizes this overall picture in terms of the best value (*Best*) and the average solution quality (*Avg*). It is clear that in terms of *Best* values, the performance of ILS is (i) better on 36, 49, and 7 (ii) same on 70, 57, and 97 (iii) worse on 1, 1, and 3, respectively, in comparison with HES-EA, ABA, and HSSGA. In terms of

Table 1 Comparison of ILS with state-of-the-art approaches

Instance	$\|V\|$	d	HES-EA				ABA				HSSGA				ILS			
			Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET
CRD502	50	2	**5480**	5489.94	10.36	1.00	5663	5777.88	40.01	1.00	**5480**	5483.66	3.25	1.00	**5480.00**	5480.00	0.00	1.00
CRD503	50	2	5079	5079.00	0.00	1.00	5079	5079.00	0.00	1.00	5079	5079.00	0.00	1.00	**4543.00**	4543.00	0.00	1.00
CRD702	70	2	6777	6864.42	44.17	1.00	7186	7466.98	128.42	1.00	**6763**	6798.50	33.32	1.00	**6763.00**	6763.14	0.35	1.00
CRD704	70	2	6380	6443.38	22.97	1.00	6620	6620.00	0.00	1.00	**6370**	6430.10	35.19	1.00	**6370.00**	6382.86	15.48	1.00
CRD706	70	2	**6671**	6673.60	5.31	1.00	6994	7222.00	70.26	1.00	**6671**	6707.20	32.37	1.00	**6671.00**	6672.74	6.89	1.00
CRD707	70	2	6470	6512.48	33.79	1.00	6789	7296.76	159.68	1.00	**6467**	6492.10	30.77	1.00	**6467.00**	6469.32	5.59	1.00
CRD100	100	2	7096	7142.84	28.33	10.00	7080	7183.98	172.73	10.00	**7044**	7075.16	33.69	10.00	**7044.00**	7054.12	16.37	10.00
CRD101	100	2	7723	7800.38	50.70	10.00	7737	7905.34	203.85	10.00	**7697**	7714.40	24.93	10.00	**7697.00**	7702.90	9.25	10.00
CRD102	100	2	7623	7674.50	24.30	10.00	7589	7671.18	116.45	10.00	**7535**	7588.90	22.14	10.00	**7535.00**	7568.94	13.78	10.00
CRD103	100	2	7703	7754.30	25.22	10.00	**7660**	7775.82	104.29	10.00	**7660**	7677.38	24.19	10.00	**7660.00**	7666.20	10.18	10.00
CRD104	100	2	7571	7645.06	36.03	10.00	**7541**	7661.42	190.69	10.00	**7541**	7552.54	23.80	10.00	**7541.00**	7541.00	0.00	10.00
CRD105	100	2	7123	7169.10	30.34	10.00	7077	7160.16	192.03	10.00	**7075**	7095.00	19.54	10.00	**7075.00**	7,075.96	1.44	10.00
CRD107	100	2	7721	7734.42	8.94	10.00	7728	7833.00	184.36	10.00	**7720**	7729.78	17.18	10.00	**7720.00**	7721.44	5.70	10.00
CRD108	100	2	7304	7534.84	129.73	10.00	7154	7245.68	220.79	10.00	**7137**	7155.58	16.77	10.00	**7137.00**	7138.60	3.20	10.00
CRD109	100	2	7771	8069.40	113.81	10.00	**7286**	7409.02	327.97	10.00	**7286**	7297.64	22.82	10.00	**7286.00**	7286.58	2.34	10.00
SYM502	50	2	**2116**	2135.60	9.08	1.00	2273	2415.42	62.14	1.00	**2116**	2131.14	11.15	1.00	**2116.00**	2133.24	16.68	1.00
SYM503	50	2	**1965**	1971.90	10.54	1.00	2162	2384.82	92.97	1.00	**1965**	1968.40	8.45	1.00	**1965.00**	1981.72	12.78	1.00
SYM700	70	2	2037	2093.92	26.42	1.00	2556	2805.08	121.09	1.00	**2012**	2069.64	22.71	1.00	**2012.00**	2054.06	23.42	1.00
SYM701	70	2	1940	1989.04	24.53	1.00	2487	2692.42	88.25	1.00	**1931**	1959.18	26.60	1.00	**1931.00**	1958.36	18.07	1.00
SYM702	70	2	**1864**	1899.60	12.16	1.00	2092	2386.00	72.10	1.00	1870	1926.34	32.59	1.00	**1864.00**	1911.38	21.28	1.00
SYM703	70	2	1503	1510.52	4.00	1.00	1674	1928.44	86.81	1.00	1503	1527.40	20.47	1.00	**1497.00**	1521.26	11.84	1.00
SYM704	70	2	2032	2085.92	19.89	1.00	2470	2728.02	96.45	1.00	**2017**	2066.22	25.24	1.00	**2017.00**	2059.92	21.72	1.00
SYM705	70	2	2157	2187.06	13.97	1.00	2507	2657.34	81.20	1.00	**2152**	2196.36	30.52	1.00	2154.00	2180.82	18.54	1.00
SYM706	70	2	1495	1509.58	10.71	1.00	1656	1897.04	90.66	1.00	**1489**	1517.72	16.96	1.00	**1489.00**	1504.38	11.24	1.00
SYM707	70	2	2384	2412.46	16.78	1.00	2666	2998.80	110.27	1.00	2388	2419.88	22.76	1.00	**2378.00**	2408.34	20.34	1.00
SYM708	70	2	1823	1840.94	8.27	1.00	2136	2319.34	93.89	1.00	**1816**	1843.26	17.14	1.00	1,823.00	1846.64	15.33	1.00
SYM709	70	2	**1749**	1777.18	19.62	1.00	2103	2382.16	104.93	1.00	**1749**	1783.74	27.37	1.00	**1749.00**	1778.46	21.37	1.00

Table 2 Comparison of ILS with state-of-the-art approaches

Instance	$\|V\|$	d	HES-EA				ABA				HSSGA				ILS			
			Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET
CRD501	50	2	**5553**	5605.58	38.36	1.00	5706	6158.14	148.32	1.00	**5553**	5553.66	4.62	1.00	**5553.00**	5,554.32	6.47	1.00
CRD501	50	3	**5126**	5126.00	0.00	1.00	**5126**	5126.00	0.00	1.00	**5126**	5126.00	0.00	1.00	**5126.00**	5,126.00	0.00	1.00
CRD700	70	2	6366	6436.60	34.86	1.00	6933	7050.88	32.50	1.00	**6308**	6365.44	38.86	1.00	**6308.00**	6320.92	24.09	1.00
CRD700	70	3	**5789**	5789.00	0.00	1.00	**5789**	5789.00	0.00	1.00	**5789**	5789.00	0.00	1.00	**5789.00**	5789.00	0.00	1.00
CRD100	100	3	**6196**	6196.00	0.00	10.00	**6196**	6196.00	0.00	10.00	**6196**	6196.00	0.00	10.00	**6196.00**	6,196.00	0.00	10.00
SYM500	50	2	**1759**	1777.26	8.36	1.00	1863	2070.54	79.78	1.00	**1759**	1769.44	7.47	1.00	**1759.00**	1775.18	9.65	1.00
SYM500	50	3	**1156**	1156.00	0.00	1.00	**1156**	1156.00	0.00	1.00	**1156**	1156.00	0.00	1.00	**1156.00**	1156.00	0.00	1.00
SYM500	50	4	**1105**	1105.00	0.00	1.00	**1105**	1105.00	0.00	1.00	**1105**	1105.00	0.00	1.00	**1105.00**	1105.00	0.00	1.00
SYM500	50	5	**1098**	1098.00	0.00	1.00	**1098**	1098.00	0.00	1.00	**1098**	1098.00	0.00	1.00	**1098.00**	1,098.72	2.20	1.00
SYM508	50	2	**1861**	1875.04	13.01	1.00	1985	2156.06	77.45	1.00	**1861**	1866.04	5.12	1.00	**1861.00**	1866.22	2.83	1.00
SYM701	70	3	**1270**	1271.12	1.80	1.00	**1270**	1295.30	12.10	1.00	**1270**	1270.00	0.00	1.00	**1270.00**	1,270.00	0.00	1.00
SYM701	70	4	**1198**	1198.00	0.00	1.00	**1198**	1199.34	2.22	1.00	**1198**	1198.00	0.00	1.00	**1198.00**	1198.16	0.54	1.00
SYM701	70	5	**1186**	1186.00	0.00	1.00	**1186**	1186.00	0.00	1.00	**1186**	1186.00	0.00	1.00	**1186.00**	1188.58	7.48	1.00
SYM708	70	2	1823	1841.42	8.37	1.00	2128	2313.70	92.30	1.00	1822	1846.20	15.92	1.00	**1816.00**	1845.78	11.13	1.00
STR500	50	2	**4420**	4422.28	2.88	1.00	4430	4472.96	12.01	1.00	**4420**	4420.00	0.00	1.00	**4420.00**	4420.00	0.00	1.00
STR500	50	3	**4118**	4118.00	0.00	1.00	**4118**	4118.00	0.00	1.00	**4118**	4118.00	0.00	1.00	**4118.00**	4118.00	0.00	1.00
STR500	50	4	**3956**	3956.00	0.00	1.00	**3956**	3956.00	0.00	1.00	**3956**	3956.00	0.00	1.00	**3956.00**	3956.00	0.00	1.00
STR500	50	5	**3807**	3807.00	0.00	1.00	**3807**	3807.00	0.00	1.00	**3807**	3807.00	0.00	1.00	**3807.00**	3,807.76	1.58	1.00
STR502	50	2	**6126**	6127.56	2.09	1.00	6159	6181.10	10.81	1.00	**6126**	6127.92	2.31	1.00	**6126.00**	6126.00	0.00	1.00
STR700	70	2	4700	4713.74	6.28	1.00	4744	4803.56	18.01	1.00	**4693**	4704.48	6.32	1.00	**4693.00**	4693.36	1.47	1.00
STR700	70	3	**4397**	4397.00	0.00	1.00	**4397**	4397.00	0.00	1.00	**4397**	4397.00	0.00	1.00	**4397.00**	4397.00	0.00	1.00
STR700	70	4	**4245**	4245.00	0.00	1.00	**4245**	4245.00	0.00	1.00	**4245**	4245.00	0.00	1.00	**4245.00**	4245.00	0.00	1.00
STR700	70	5	**4100**	4100.00	0.00	1.00	**4100**	4100.00	0.00	1.00	**4100**	4100.00	0.00	1.00	**4100.00**	4114.16	22.56	1.00
STR1000	100	2	**5000**	5021.10	8.91	10.00	5004	5030.96	31.67	10.00	**5000**	5004.50	6.65	10.00	**5000.00**	5000.00	0.00	10.00
STR1000	100	3	**4702**	4702.08	0.27	10.00	**4702**	4702.70	0.45	10.00	**4702**	4702.00	0.00	10.00	**4702.00**	4702.00	0.00	10.00
STR1000	100	4	**4546**	4546.00	0.00	10.00	**4546**	4546.00	0.00	10.00	**4546**	4546.00	0.00	10.00	**4546.00**	4546.00	0.00	10.00

Table 3 Comparison of ILS with state-of-the-art approaches

Instance	$\|V\|$	d	HES-EA				ABA				HSSGA				ILS			
			Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET
STR1000	100	5	**4403**	4403.00	0.00	10.00	**4403**	4403.00	0.00	10.00	**4403**	4403.00	0.00	10.00	**4403.00**	4403.22	0.67	10.00
STR1001	100	2	**5000**	5022.56	7.56	10.00	5006	5027.92	29.65	10.00	**5000**	5004.80	7.52	10.00	**5000.00**	5000.00	0.00	10.00
STR1002	100	2	7094	7103.70	7.56	10.00	7094	7127.42	30.6823	10.00	**7089**	7091.96	3.38	10.00	**7089.00**	7089.06	0.42	10.00
STR1003	100	2	7093	7104.24	8.55	10.00	7105	7129.18	31.74	10.00	**7089**	7091.98	2.75	10.00	**7089.00**	7089.00	0.00	10.00
STR1004	100	2	8705	8730.94	11.13	10.00	8732	8760.24	22.1617	10.00	**8699**	8703.00	2.99	10.00	**8699.00**	8699.06	0.42	10.00
STR1005	100	2	8703	8731.30	10.51	10.00	8727	8761.96	24.60	10.00	**8699**	8703.64	4.66	10.00	**8699.00**	8699.00	0.00	10.00
STR1006	100	2	10,543	10,555.96	8.59	10.00	10532	10,559.18	26.3034	10.00	**10529**	10,535.50	4.17	10.00	**10529.00**	10529.08	0.56	10.00
STR1007	100	2	10,532	10,555.20	8.49	10.00	10,536	10,558.10	26.3668	10.00	**10,529**	10,534.76	4.16	10.00	**10,529.00**	10,529.00	0.00	10.00
STR1008	100	2	12,446	12,474.90	12.24	10.00	12,485	12,534.92	27.6412	10.00	12,427	12,433.52	5.58	10.00	**12,426.00**	12,426.72	0.75	10.00
STR1009	100	2	12,412	12,431.14	9.23	10.00	12,440	12,480.56	29.15	10.00	**12,401**	12,412.42	5.98	10.00	**12,401.00**	12,402.60	2.41	10.00
SHRD159	15	2	**904**	904.00	0.00	1.00	**904**	904.90	1.0817	1.00	**904**	904.00	0.00	1.00	**904.00**	904.00	0.00	1.00
SHRD159	15	3	**597**	597.00	0.00	1.00	**597**	597.00	0.00	1.00	**597**	597.00	0.00	1.00	**597.00**	597.00	0.00	1.00
SHRD159	15	4	**430**	430.00	0.00	1.00	**430**	430.00	0.00	1.00	**430**	430.00	0.00	1.00	**430.00**	430.00	0.00	1.00
SHRD159	15	5	**332**	332.00	0.00	1.00	**332**	332.00	0.00	1.00	**332**	332.00	0.00	1.00	**332.00**	332.00	0.00	1.00
SHRD200	20	2	**1679**	1679.00	0.00	1.00	**1679**	1679.10	0.30	1.00	**1679**	1679.00	0.00	1.00	**1,679.00**	1,679.00	0.00	1.00
SHRD200	20	3	**1088**	1088.00	0.00	1.00	**1088**	1088.00	0.00	1.00	**1088**	1088.00	0.00	1.00	**1,088.00**	1,088.00	0.00	1.00
SHRD200	20	4	**802**	802.00	0.00	1.00	**802**	802.00	0.00	1.00	**802**	802.00	0.00	1.00	**802.00**	802.00	0.00	1.00
SHRD200	20	5	**627**	627.04	0.20	1.00	**627**	627.00	0.00	1.00	**627**	627.00	0.00	1.00	**627.00**	627.00	0.00	1.00
SHRD259	25	2	**2714**	2717.80	9.90	1.00	**2714**	2717.02	2.11	1.00	**2714**	2714.00	0.00	1.00	**2714.00**	2,714.00	0.00	1.00
SHRD259	25	3	**1756**	1756.06	0.42	1.00	**1756**	1756.62	1.26	1.00	**1756**	1756.00	0.00	1.00	**1,756.00**	1,756.00	0.00	1.00
SHRD259	25	4	**1292**	1292.00	0.00	1.00	**1292**	1292.06	0.23	1.00	**1292**	1292.00	0.00	1.00	**1292.00**	1,292.00	0.00	1.00
SHRD259	25	5	**1016**	1016.02	0.14	1.00	**1016**	1016.00	0.00	1.00	**1016**	1016.00	0.00	1.00	**1,016.00**	1,016.00	0.00	1.00
SHRD300	30	2	**3992**	3995.98	6.27	1.00	3996	4002.84	3.18	1.00	**3992**	3992.00	0.00	1.00	**3992.00**	3,992.00	0.00	1.00
SHRD300	30	3	**2592**	2592.22	0.41	1.00	**2592**	2594.34	1.36	1.00	**2592**	2592.00	0.00	1.00	**2592.00**	2,592.00	0.00	1.00
SHRD300	30	4	**1905**	1905.18	0.74	1.00	**1905**	1907.42	1.60	1.00	**1905**	1905.00	0.00	1.00	**1905.00**	1,905.00	0.00	1.00
SHRD300	30	5	**1504**	1504.26	0.44	1.00	**1504**	1506.20	1.46	1.00	**1504**	1504.00	0.00	1.00	**1504.00**	1,504.00	0.00	1.00

Table 4 Comparison of ILS with state-of-the-art approaches

Instance	$\|V\|$	d	HES-EA				ABA				HSSGA				ILS			
			Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET	Best	Avg	SD	ATET
R50n1	50	5	**4.04**	4.04	0.00	1.00	**4.04**	4.04	0.00	1.00	**4.04**	4.04	0.00	1.00	**4.04**	4.04	0.00	1.00
R50n2	50	4	**4.26**	4.26	0.00	1.00	**4.26**	4.26	0.00	1.00	**4.26**	4.26	0.00	1.00	**4.26**	4.26	0.00	1.00
R50n2	50	5	**3.94**	3.94	0.00	1.00	**3.94**	3.94	0.00	1.00	**3.94**	3.94	0.00	1.00	**3.94**	3.95	0.01	1.00
R50n3	50	5	**3.92**	3.93	0.00	1.00	**3.92**	3.92	0.00	1.00	**3.92**	3.92	0.00	1.00	**3.92**	3.93	0.01	1.00
R100n1	100	4	**8.06**	8.06	0.00	10.00	**8.06**	8.06	0.00	1.00	**8.06**	8.06	0.00	10.00	**8.06**	8.06	0.00	10.00
R100n1	100	5	**7.51**	7.51	0.00	10.00	**7.51**	7.51	0.00	10.00	**7.51**	7.51	0.00	10.00	**7.51**	7.51	0.00	10.00
R100n2	100	4	**8.06**	8.06	0.00	10.00	**8.06**	8.07	0.01	10.00	**8.06**	8.06	0.00	10.00	**8.06**	8.06	0.00	10.00
R100n2	100	5	**7.58**	7.58	0.00	10.00	**7.58**	7.58	0.00	10.00	**7.58**	7.58	0.00	10.00	**7.58**	7.59	0.01	10.00
R100n3	100	4	**8.02**	8.02	0.00	10.00	**8.02**	8.05	0.02	10.00	**8.02**	8.02	0.00	10.00	**8.02**	8.02	0.00	10.00
R100n3	100	5	**7.50**	7.50	0.00	10.00	**7.50**	7.50	0.00	10.00	**7.50**	7.50	0.00	10.00	**7.50**	7.50	0.00	10.00
R200n1	200	4	**16.09**	16.09	0.00	60.00	**16.09**	16.10	0.00	60.00	**16.09**	16.09	0.00	60.00	**16.09**	16.10	0.00	60.00
R200n1	200	5	**15.05**	15.06	0.00	60.00	**15.05**	15.06	0.00	60.00	**15.05**	15.05	0.00	60.00	**15.05**	15.07	0.01	60.00
R200n2	200	4	**15.69**	15.69	0.00	60.00	**15.69**	15.70	0.00	60.00	**15.69**	15.69	0.00	60.00	**15.69**	15.70	0.01	60.00
R200n2	200	5	**14.65**	14.66	0.00	60.00	**14.65**	14.66	0.00	60.00	**14.65**	14.66	0.00	60.00	**14.65**	14.68	0.01	60.00
R200n3	200	4	15.63	15.63	0.00	60.00	15.63	15.64	0.01	60.00	**15.62**	15.63	0.00	60.00	**15.62**	15.63	0.00	60.00
R200n3	200	5	**14.59**	14.59	0.00	60.00	**14.59**	14.59	0.00	60.00	**14.59**	14.59	0.00	60.00	**14.59**	14.60	0.01	60.00
m050n1	50	5	**6.60**	6.60	0.01	1.00	**6.60**	6.63	0.01	1.00	**6.60**	6.60	0.00	1.00	**6.60**	6.61	0.01	1.00
m050n2	50	5	5.78	5.85	0.17	1.00	**5.78**	5.79	0.01	1.00	**5.78**	5.78	0.00	1.00	**5.78**	5.78	0.01	1.00
m050n3	50	5	**5.50**	5.50	0.00	1.00	**5.50**	5.50	0.00	1.00	**5.50**	5.50	0.00	1.00	**5.50**	5.51	0.01	1.00
m100n1	100	5	**11.08**	11.26	0.28	10.00	**11.08**	11.23	0.31	10.00	**11.08**	11.09	0.00	10.00	**11.08**	11.09	0.00	10.00
m100n2	100	5	**11.33**	11.74	0.40	10.00	**11.33**	11.79	0.42	10.00	**11.33**	11.35	0.12	10.00	**11.33**	11.34	0.00	10.00
m100n3	100	5	**10.19**	10.43	0.37	10.00	10.20	10.52	0.36	10.00	**10.19**	10.19	0.00	10.00	**10.19**	10.20	0.01	10.00
m200n1	200	5	**18.33**	18.67	0.40	60.00	**18.33**	18.42	0.19	60.00	**18.33**	18.57	0.32	60.00	18.34	18.37	0.02	60.00
m200n2	200	5	19.18	20.87	0.96	60.00	19.19	19.45	0.33	60.00	**19.16**	19.23	0.24	60.00	**19.16**	19.21	0.02	60.00
m200n3	200	5	**16.13**	16.13	0.00	60.00	**16.13**	16.14	0.09	60.00	**16.13**	16.13	0.00	60.00	**16.13**	16.17	0.03	60.00
m300n1	300	5	44.53	46.44	0.81	200.00	40.71	40.85	0.26	200.00	**40.68**	40.83	0.26	200.00	**40.68**	40.69	0.01	200.00
m400n1	400	5	67.99	70.85	1.20	600.00	54.96	58.11	1.85	600.00	**54.64**	55.01	0.43	600.00	**54.64**	54.66	0.01	600.00
m500n1	500	5	97.47	100.03	1.36	1000.00	79.57	82.75	1.61	1000.00	79.31	80.40	0.57	1000.00	**79.30**	79.32	0.01	1,000.00

Table 5 Overall picture of results of ILS against the results of HES-EA [12], ABA [18] and HSSGA [19] approaches

Comparison	Best			Avg		
	HES-EA	ABA	HSSGA	HES-EA	ABA	HSSGA
Total instances	107	107	107	107	107	107
Better	36	49	7	58	70	44
Equal	70	57	97	32	24	39
Worse	1	1	3	17	13	24

Avg values, the performance of ILS is (i) better on 58, 70, and 44 (ii) same on 32, 24, and 39 (iii) worse on 17, 13, and 24, respectively, in comparison with HES-EA, ABA, and HSSGA. It is clear from Table 5 that ILS is superior to HES-EA, ABA, and HSSGA in terms of both $Best$ and Avg.

5 Conclusions

This study presented an iterated local search algorithm (ILS) for the degree-constrained minimum spanning tree (DCMST) problem. All components of ILS (initial solution, perturbation, local search strategies, and acceptance criterion) effectively coordinate and help in finding high-quality solutions. To evaluate the performance of ILS, computational experiments have been tested on a set of benchmark instances. The results demonstrate that the performance of ILS is overall superior in comparison with state-of-the-art approaches.

For future work, components of ILS used for DCMST problem can be adapted to various variants of $\mathcal{NP}$-hard spanning tree problems.

References

1. Garey MR, Johnson DS (1979) Computers and intractability, vol 174. Freeman San Francisco
2. Bruce B, Narsingh D, Nishit K (1996) Minimum-weight degree-constrained spanning tree problem: heuristics and implementation on an simd parallel machine. Parallel Comput 22(3):369–382
3. Martin S, Ton V (1985) Edge exchanges in the degree-constrained minimum spanning tree problem. Comput Operat Res 12(4):341–348
4. Narula SC, Ho CA (1980) Degree-constrained minimum spanning tree. Comput Operat Res 7(4):239–249
5. Ravi R, Marathe MV, Ravi SS, Rosenkrantz DJ, Hunt III HB (1993) Many birds with one stone: multi-objective approximation algorithms. In: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, pp 438–447

6. Prim RC (1957) Shortest connection networks and some generalizations. Bell Syst Tech J 36:1389–1401
7. Singh K, Sundar S (2019) A heuristic for the degree-constrained minimum spanning tree problem. In: Soft computing: theories and applications. Springer, pp 351–363
8. Singh K, Sundar S (2019) A new heuristic for degree-constrained minimum spanning tree problem. In: Computational intelligence: theories, applications and future directions-volume I. Springer, pp 149–160
9. Zhou G, Gen M (1997) A note on genetic algorithms for degree-constrained spanning tree problems. Netw An Int J 30(2):91–95
10. Joshua K, David C (2000) A new evolutionary approach to the degree-constrained minimum spanning tree problem. IEEE Trans Evol Comput 4(2):125–134
11. Raidl GR, Julstrom BA (2000) A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In: Proceedings of the 2000 ACM symposium on Applied computing, vol 1, pp 440–445
12. Raidl GR, Julstrom BA (2003) Edge sets: an effective evolutionary coding of spanning trees. IEEE Trans Evol Comput 7(3):225–239
13. Bau Y-T, Ho CK, Ewe HT (2005) An ant colony optimization approach to the degree-constrained minimum spanning tree problem. In: International conference on computational and information science. Springer, pp 657–662
14. Bui TN, Zrncic CM (2006) An ant-based algorithm for finding degree-constrained minimum spanning tree. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, pp 11–18
15. Doan MN (2007) An effective ant-based algorithm for the degree-constrained minimum spanning tree problem. In: 2007 IEEE congress on evolutionary computation. IEEE, pp. 485–491
16. Binh HTT, Nguyen TB (2008) New particle swarm optimization algorithm for solving degree constrained minimum spanning tree problem. In: Pacific rim international conference on artificial intelligence. Springer, pp 1077–1085
17. Ernst AT (2010) A hybrid lagrangian particle swarm optimization algorithm for the degree-constrained minimum spanning tree problem. In: IEEE congress on evolutionary computation, IEEE, pp 1–8
18. Bui TN, Deng X, Zrncic CM (2011) An improved ant-based algorithm for the degree-constrained minimum spanning tree problem. IEEE Trans Evol Comput 16(2):266–278
19. Kavita S, Shyam S (2020) A hybrid genetic algorithm for the degree-constrained minimum spanning tree problem. Soft Comput 24(3):2169–2186
20. Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. In: Handbook of metaheuristics. Springer, pp 320–353
21. Lourenço HR, Martin OC, Stützle T (2019) Iterated local search: framework and applications. In: Handbook of metaheuristics. Springer, pp 129–168

Chapter 27
Comparative Evaluation of Machine Learning Models for the Prediction of Diabetes at Early Stage

Cagri Ozkan and Boran Sekeroglu

1 Introduction

Chronic diseases continue for the long term and have permanent adverse effects on the patients' life. Diabetes is one of the most acute diseases worldwide [1] and causes by high levels of blood glucose. It can cause severe damage to body functions by affecting organs, blood vessels, eyes, etc. [2]. The records showed that there are more than 420 million diabetes patients globally [3].

There are three types of diabetes mellitus: Type 1, Type 2, and gestational diabetes [2]. Type 1 diabetes is the most severe type caused by the attack of the immune system on the pancreatic beta cells [3]. The ineffective use of insulin by the body causes Type 2 [4], and gestational diabetes is shown only in the pregnancy period [5].

Several kinds of research have been performed in healthcare sector and particularly to predict diabetes at the early stage, and recently, artificial intelligence (AI) and machine learning (ML) implementations are at the core of these researches [6, 7]. AI and particularly ML models have the ability to establish relations between the attributes of non-linear data, which yields higher rates for the healthcare sector and particularly for diabetes prediction or classification.

Ogedengbe and Egbunu [8] employed a feature selection technique, classifier subset evaluator (CSE), to select the most relevant risk factors of diabetes, and the J48 decision tree was used to predict diabetes. The proposed system was compared to the other ML models such as Naïve-Bayes (NB), support vector machine (SVM). F-measure, receiver operating characteristics (ROC), recall, and accuracy metrics were used in the evaluation.

C. Ozkan (✉) · B. Sekeroglu
Information Systems Engineering, Near East University, 99138, Nicosia, N. Cyprus , Mersin 10, Turkey
e-mail: cagri.ozkan@neu.edu.tr

B. Sekeroglu
e-mail: boran.sekeroglu@neu.edu.tr

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_28

Pei et al. [9] implemented a cost-sensitive J48 decision tree for the estimation of diabetes. Their study included 3454 participants in the final dataset, and randomly selected 70% of the data was used for the training. In addition to the estimation of diabetes, several key factors were identified using the constructed decision tree.

Hassan et al. [10] employed SVM, decision tree (DT), and k-nearest neighbor (kNN) models to predict diabetes mellitus. The authors concluded that the SVM model outperformed other considered models in terms of accuracy.

Sarker et al. [11] proposed an optimal-kNN for the prediction of diabetes mellitus. The authors compared the proposed model to the adaptive boosting (AdaBoost), logistic regression (LR), NB, SVM, and DT. It was concluded the optimal-kNN outperformed other considered models in precision, recall (sensitivity), F-measure, and ROC metrics.

Zou et al. [12] considered randomly selected 68,994 participants for their study. Principal component analysis (PCA) and minimum redundancy maximum relevance (mRMR) were used to reduce the dimensionality. Random forest (RF), J48 DT, and artificial neural networks (ANN) were employed. It was concluded that the random forest achieved superior accuracy and outperformed other models.

Kavitha and Singh [13] implemented DT, NB, and multilayer perceptron (MLP) models to predict diabetes at an early stage. It was concluded that MLP achieved superior results than other considered models and is capable of predicting diabetes accurately.

In addition to the prediction studies of diabetes, several comparative studies have been performed to demonstrate the abilities of the machine learning models [14].

Sadhu and Jadli [15] considered seven ML models, DT, RF, SVM, MLP, k-NN, NB, and logistic regression (LR), to compare diabetes prediction. The accuracy, ROC, and F-measure metrics were used for the evaluation, and it was concluded that the RF model outperformed other models in predicting diabetes.

Chaves and Marques [16] implemented NB, ANN, AdaBoost, kNN, RF, and SVM to evaluate the abilities of the models for diabetes prediction. The results showed that ANN outperformed other models.

Kareem et al. [17] implemented RF, MLP, and radial basis function neural networks (RBFNN) for diabetes prediction. It was concluded that the MLP and RBFNN achieved the highest specificity rates, and RBFNN produced the highest accuracy when trained using tenfold cross-validation.

This paper aims to perform a comparative evaluation of nine benchmark and common machine learning models to predict diabetes at an early stage and direct further research in selecting ML models by extending the recent studies.

The rest of the paper is organized as follows: Sect. 2 introduces the considered dataset and presents the review of ML models employed in this study. Section 3 presents the obtained results. Section 4 performs the comparisons and discusses the results, and finally, Sect. 5 concludes the remarks of this study.

2 Materials and Methods

2.1 Dataset

The early stage diabetes risk prediction dataset [18], which is the common for the recent studies, is considered in this study to perform a comparative evaluation of ML models.

The data was collected by questionnaires of the patients aged between 20 and 65 (mean= 48.02) in Bangladesh, and it includes 17 attributes and 520 instances for binary classification of positive ($n = 320$) and negative ($n = 200$) cases. Table 1 shows the 16 attributes of the dataset which the patients answered.

Figure 1 shows the correlations between the attributes as a heatmap. Darker and lighter colors represent more and fewer correlations between the attributes of the dataset.

Table 1 Attributes of the early stage diabetes prediction dataset

Attributes			
Age	Sex	Polyuria	Polydipsia
Sudden weight loss	Weakness	Polyphagia	Genital thrush
Visual blurring	Itching	Irritability	Delayed healing
Partial paresis	Muscle stiffness	Alopecia	Obesity

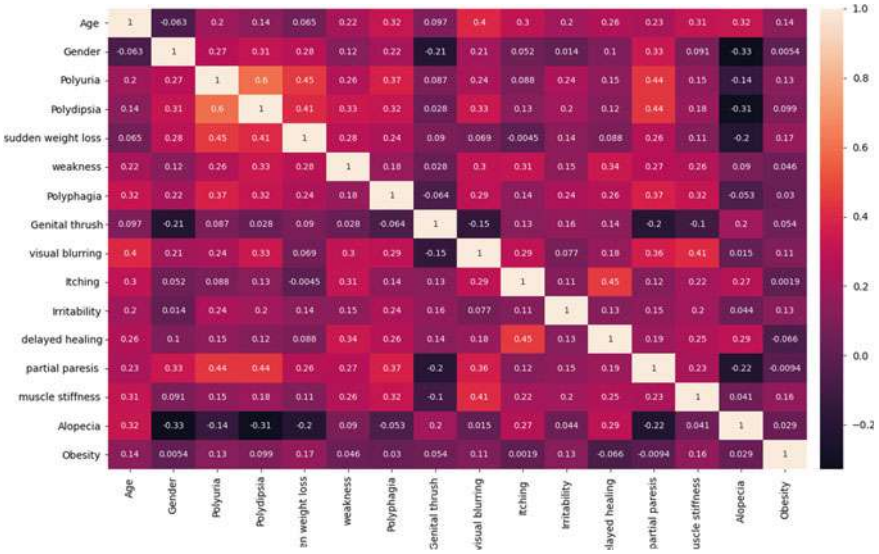


Fig. 1 Correlation heatmap of the dataset attributes

2.2 Review of Machine Learning Models

Several classification models were proposed considering different learning schemes. In this study, support vector machines with linear kernel (SVM), decision tree (DT), random forest (RF), adaptive boosting (AdaBoost), gradient boosting (GradBoost), extreme gradient boosting (XGBoost), logistic regression (LR), Gaussian naïve Bayes (GNB), and k-nearest neighbor (kNN) models are considered for the evaluation and comparison for the early stage prediction of diabetes. The following subsection overviews the considered ML models briefly.

2.2.1 Support Vector Machines

Support vector machine (SVM) maps training samples to points in a hyperplane and assigns closer features of classes as support vectors in order to perform classification [19]. Therefore, the gap between the classes becomes more significant for the distinguishment. Different kernels are used to map the samples into high-dimensional spaces and to perform non-linear classification. It has been frequently used for classification tasks [20].

2.2.2 Decision Tree

Decision trees are tree-structured algorithms with initial root node, decision nodes, and leaf nodes. They are using divide-and-conquer strategy and this brings several advantages and also disadvantages for them [21]. Its' simplicity and speed are the main advantages of decision trees; however, determination of initial root or the sequence of nodes is the main drawback of decision trees.

2.2.3 Random Forests

Random forests are a kind of tree-based ensemble learning and can be used for both classification and regression [22]. It constructs several decision trees during the training and optimizes the mean regression of the individual trees.

2.2.4 Adaptive Boosting

It can be used to improve the performance of other learning algorithms [23]. The weak learners are combined into a weighted sum that represents the final output of the boosted classifier. Therefore, the final model can be proven to converge to a strong learner. The weak learners of AdaBoost for this study is employed with decision trees.

2.2.5 Gradient Boosting Algorithm

Gradient boosting is another tree-based ensemble machine learning algorithm [24]. It aims to optimize the outputs by minimizing the loss obtained by the constructed weak learners, which are decision trees. The loss is calculated, and then a new or modified tree is added to reduce the total loss using a gradient descent algorithm. The model's output is modified after adding each tree to the model, and different stopping criteria such as no decrement in loss, adding a fixed number of trees, etc., can be applied to obtain the final output of the model.

2.2.6 Extreme Gradient Boosting

Similar to gradient boosting, extreme gradient boosting [25] is also an ensemble tree method and applies the principle of boosting weak learners using the gradient descent algorithm. However, XGBoost includes some enhancements to minimize the used resources and to improve the obtained results. Different regularization models (i.e., LASSO) are used to overcome overfitting problems during the learning. The built-in cross-validation is applied in each iteration to determine the exact number of iterations on a single run.

2.2.7 Logistic Regression

In binary classification tasks, logistic regression (LR) is a statistical model that uses a logistic (Sigmoid) function to distinguish the samples into the proper classes by calculating the probability of output [26].

2.2.8 Gaussian Naive Bayes

The Naive Bayes classifiers is another probabilistic classifier based on Bayes' theorem [27]. It needs a small amount of data for classification. In Gaussian Naive Bayes, the continuous data associated with the classes are distributed according to the Gaussian distribution.

2.2.9 K-Nearest Neighbor

In kNN, where the output is a class membership, an object is classified by using its neighbors. The data are assigned to the class considering the most common k nearest neighbors, which relies on distance for classification [28].

2.3 Evaluation and Validation of Experiments

True positive (TP), true negative (TN), false positive (FP), and false negative (FN) values produced the models are used for the calculation of evaluation metrics in classification tasks. Positive and negative values indicate the prediction of the model, and true and false values indicate whether the predicted values are correct or not.

The most common evaluation metric for the balanced data is accuracy which is the ratio of the number of correctly classified samples to the total number of samples. Equation 1 shows the formula of the accuracy.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Sensitivity is used to measure the ability of the model to predict observed positives correctly. Equation 2 shows the formulae of the sensitivity.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Contrary, the specificity is a metric to measure how the model is capable of predicting the negative samples correctly. Equation 3 shows the formulae of the specificity.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

The robustness of the accuracy is limited to the balanced data. Therefore, the receiver operating characteristic (ROC) area under curve (AUC) score (ROC AUC score) is also considered as the general evaluation of the models. It is one of the common metrics in imbalanced data of binary classification tasks.

All experiments were performed using two-stage fivefold cross-validation. In the initial stage, different parameters were assigned to each fold, and the superior parameters were defined. In the second stage, fivefold cross-validation was applied using the best parameters of the models, and the scores were obtained for each fold. The final results of the experiments were calculated by considering the mean of the metrics of all folds.

3 Results

As mentioned above, nine ML models are employed for the comparative evaluation of predicting diabetes at an early stage.

Table 2 Sensitivity, specificity, accuracy, and the ROC AUC scores obtained by the models in this study

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	ROC AUC score
DT	96.92	**97.50**	96.00	0.9675
RF	96.35	96.25	96.50	0.9638
GradBoost	97.12	**97.50**	96.50	0.9700
SVM	93.08	92.81	93.50	0.9316
Logistic regression	91.92	93.13	90.00	0.9156
XGBoost	**97.50**	97.19	**98.00**	**0.9759**
GNB	89.23	92.50	84.00	0.8825
KNN	91.92	89.69	95.50	0.9259
AdaBoost	91.73	93.44	89.00	0.9122

DTs were constructed using entropy, and GradBoost, AdaBoost, and RF were trained with 165, 150, and 500 estimators, respectively. The learning rates of XGBoost, GradBoost, and AdaBoost were defined as 0.3, 0.085, and 0.085.

The k value of the kNN model was determined as 7, and the final γ and ϵ parameters of SVM were determined as 0.0001 and 0.001.

Table 2 presents all the obtained sensitivity, specificity, accuracy, and the ROC AUC scores in this study.

Even though the Gaussian Naive Bayes achieved higher rates than the kNN in terms of sensitivity (NB = 92.50%, kNN = 89.68%), it produced the lowest specificity, accuracy, and ROC AUC scores (84.00%, 89.23%, and 0.88, respectively). AdaBoost and logistic regression achieved closer results for all metrics; however, AdaBoost achieved a higher sensitivity rate (93.43%) than LR (93.12%), while the LR produced a superior specificity rate (90.0%) than AdaBoost (89.0%). The LR and AdaBoost models achieved 91.92% and 91.73% accuracy and 0.915 and 0.912 ROC AUC scores, respectively.

Although the kNN model achieved a superior specificity rate (95.50%) than NB, AdaBoost, and LR, it produced 91.92% accuracy and 0.925 ROC AUC scores due to the lowest sensitivity rates obtained in this study.

SVM achieved 93.07% accuracy, 92.81% sensitivity, 93.50% specificity, and 0.931 ROC AUC scores and outperformed the NB, LR, AdaBoost, and kNN models in general evaluation. However, it could not achieve the rates obtained by DT and other tree-ensemble methods RF, GradBoost, and XGBoost.

GradBoost, RF, and DT obtained similar and closer specificity rates (96.50%, 96.50%, and 96.0%). However, the GradBoost and DT achieved the superior sensitivity rates of the study (97.50%), and the accuracy and ROC AUC scores of GradBoost and DT are obtained as 97.11%, 0.970, and 96.92%, 0.967, respectively. The RF model achieved 96.34% accuracy and a 0.963 ROC AUC score.

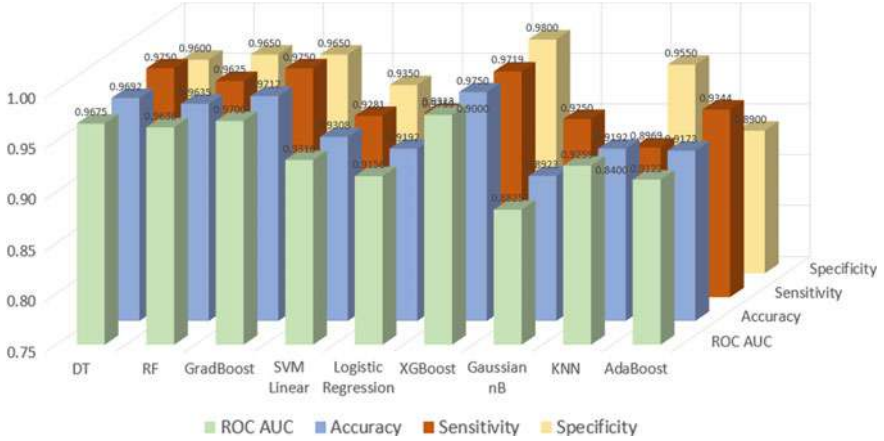


Fig. 2 Visualization of the obtained results

The XGBoost model obtained the superior specificity (98.0%), accuracy (97.50%), and ROC AUC score (0.9759) of the study.

Figure 2 visualizes the sensitivity, specificity, accuracy, and the ROC AUC scores using bar plot.

4 Comparisons and Discussion

Even though all the considered models achieved a higher than 0.88 ROC AUC score, DT and three tree-ensemble methods achieved higher results for correctly classifying the participants as healthy and diabetes.

DT and GradBoost achieved the highest true positive results (TP = 312) in detecting the positive samples. In parallel to the obtained TP and TN results, DT and GradBoost achieved the highest sensitivity rates (97.50%). These models were followed by the XGboost (TP = 312, Sensitivity = 97.19%) and RF (TP = 308, Sensitivity = 96.25%). However, XGBoost outperformed other models in classifying negative samples and produced the study's highest true negative result (TN = 196), and achieved a 98.00% specificity rate. XGBoost was followed by GradBoost, RF, DT, and kNN (TN = 193, 193, 192, 191, respectively).

Conversely, the kNN and GNB models produced the lowest TP (kNN = 287) and TN (GNB = 168) results. Accordingly, these models produced the lowest sensitivity rates of the study (89.69% and 92.50%, respectively).

Even though the LR and SVM achieved higher TP results than GNB and kNN, they were not able to achieve superior results than XGBoost, RF, GradBoost, and DT.

Table 3 TP, TN, FP, and FN results obtained by the models in this study

Model	True positive	True negative	False positive	False negative
DT	**312**	192	8	**8**
RF	308	193	7	12
GradBoost	**312**	193	7	**8**
SVM	297	187	13	23
Logistic regression	298	180	20	22
XGBoost	311	**196**	**4**	9
GNB	296	168	32	24
KNN	287	191	9	33
AdaBoost	299	178	22	21

Table 3 shows the true positive, true negative, false positive, and false negative results obtained by the models.

In the general evaluation of the models, the lowest accuracy and ROC AUC scores were obtained by GNB (89.23% and 0.8825, respectively), followed by the AdaBoost model (91.73% and 0.9122, respectively).

XGBoost achieved superior results in predicting diabetes at an early stage with 97.50% accuracy and 0.9759.

Figure 3 shows the confusion matrices of XGBoost, RF, GradBoost, and DT models in which the highest TP and TN results were produced.

The results showed that the DT, and tree-ensemble models, RF, GradBoost, and XGBoost have certain abilities to establish relations between the attributes and to produce higher results than other models. However, even though the AdaBoost is an ensemble method, it could not produce reasonable results as other ensemble models due to the lower specificity rate achieved.

The obtained results demonstrated the efficiency of tree-based ensemble methods in predicting diabetes since they construct several decision trees and minimizes error. Furthermore, the construction of several trees and minimizing the prediction error within these trees provide models to learn samples robustly. This yielded these models to obtain higher prediction rates than other single-stage classification models except AdaBoost.

Table 4 presents the ranking of the considered models for different evaluation metrics.

The main limitation of this study is related to the parameters of the models. Even though the superior parameters of the considered models were determined using cross-fold validation, further implementation is required to analyze the models' abilities with different parameters. In addition, the analysis of the models using different k values for k-fold cross-validation and the comparison of the considered models to the neural-based models might affect the obtained results.

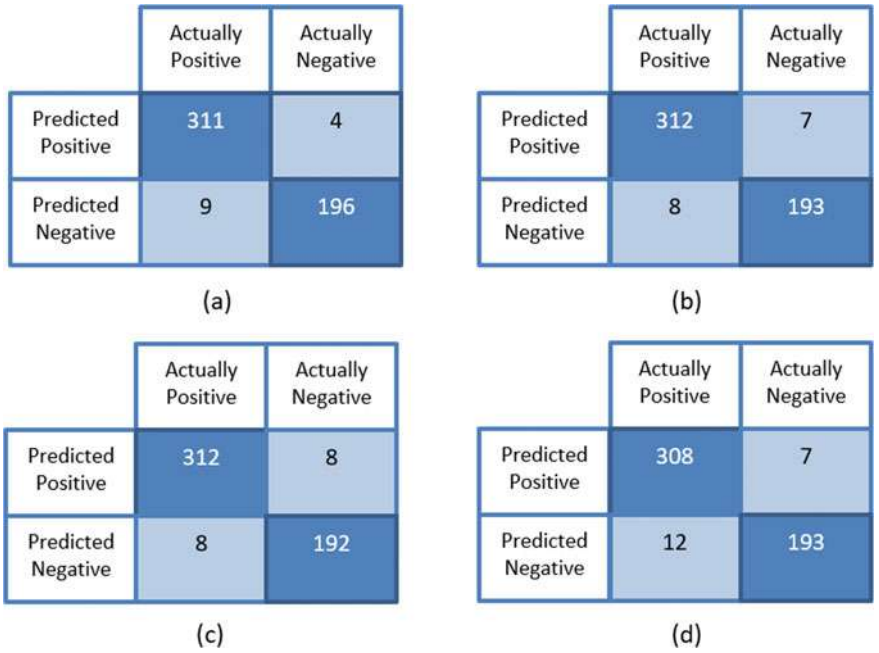


Fig. 3 Confusion matrices **a** XGBoost, **b** GradBoost, **c** Decision tree, and **d** Random forest

Table 4 Ranking of the models based on the performance metrics

Rank	TP	TN	Sensitivity	Specificity	Accuracy	ROC AUC
1	DT	XGBoost	GradBoost	XGBoost	XGBoost	XGBoost
2	GradBoost	GradBoost	DT	GradBoost	GradBoost	GradBoost
3	XGBoost	RF	XGBoost	RF	DT	DT
4	RF	DT	RF	DT	RF	RF
5	AdaBoost	KNN	AdaBoost	KNN	SVM	SVM
6	LR	SVM	LR	SVM	KNN	KNN
7	SVM	LR	SVM	LR	LR	LR
8	GNB	AdaBoost	GNB	AdaBoost	AdaBoost	AdaBoost
9	KNN	GNB	KNN	GNB	GNB	GNB

5 Conclusion

Several studies have been proposed to predict early stage diabetes, and accordingly, different comparative studies were performed to analyze the abilities of the machine learning models for this purpose. In this study, we extended the recent researches to perform a comparative evaluation for diabetes prediction and considered nine benchmark machine learning models on the common dataset.

Initially, the best hyperparameters were determined using fivefold cross-validation. Then, the experiments were performed using determined hyperparameters for each model. The results were analyzed using the accuracy, ROC AUC score, sensitivity, and specificity metrics.

The obtained results showed that the extreme gradient boosting model outperformed other models and achieved the highest accuracy, ROC AUC score, and specificity rates.

In addition, decision tree and tree-ensemble methods random forest and gradient boosting achieved more reasonable results than other considered models and demonstrated the efficiency of the ensemble models in predicting diabetes at an early stage.

Our future work will include developing a novel ensemble model using the superior models of this study to achieve higher rates.

References

1. Mahboob Alam T, Iqbal MA, Ali Y, Wahab A, Ijaz S, Imtiaz Baig T, Hussain A, Malik MA, Raza MM, Ibrar S, Abbas Z (2019) A model for early prediction of diabetes. Informatics Med Unlocked 16:100204
2. Oladimeji O, Oladimeji A, Oladimeji O (2021) Classification models for likelihood prediction of diabetes at early stage using feature selection. Appl Comput Informatics
3. World Health Organization. Diabetes fact sheet. www.who.int/news-room/fact-sheets/detail/diabetes. Accessed 2021-09-11
4. Metzger B, Lowe L, Dyer A, Trimble E, Chaovarindr U, Coustan D, Hadden D, McCance D, Hod M, Mclntyre HJO, Persson B, Rogers M, Sacks D (2008) Hyperglycemia and adverse pregnancy outcomes. New England J Med 358:1991–2002
5. Sneha N, Gangil T (2019) Analysis of diabetes mellitus for early prediction using optimal features selection. J Big Data 6
6. Sekeroglu B, Tuncal K (2021) Prediction of cancer incidence rates for the European continent using machine learning models. Health Informatics J 27(1):1460458220983878, 100204
7. Abiyev RH, Ma'aitah M (2018) Deep convolutional neural networks for chest diseases detection. Vision based computing systems for healthcare applications. J Healthcare Eng (4168538)
8. Ogedengbe M, Egbunu C (2020) Cse-dt features selection technique for diabetes classification. Appl Modelling Simul 4:101–109
9. Pei D, Yang T, Zhang C (2020) Estimation of diabetes in a high-risk adult Chinese population using j48 decision tree model. Diabetes Metab Syndr Obes 13:4621–4630, 100204
10. Hassan A, Malaserene I, Leema A (2020) Diabetes mellitus prediction using classification techniques. Int J Innov Technol Exploring Eng 9:2278–3075
11. Sarker I, Faruque M, Alqahtani H, Kalim A (2019) K-nearest neighbor learning based diabetes mellitus prediction and analysis for ehealth services. EAI Endorsed Trans Scalable Information Syst 162737
12. Zou Q, Qu K, Luo Y, Yin D, Ju Y, Tang H (2018) Predicting diabetes mellitus with machine learning techniques. Front Genetics 9:515, 100204
13. Kavitha R, Singh W (2020) A study on the effectiveness of machine learning algorithms in early prediction of diabetics among patients. Biosci Biotechnol Res Commun 13:99–104
14. Sekeroglu B, Hasan S, Abdullah S (2020) Comparison of machine learning algorithms for classification problems. In: Arai K, Kapoor S (eds) Advances in computer vision. Springer International Publishing, Cham, pp 491–499
15. Sadhu A, Jadli A (2021) Early-stage diabetes risk prediction: a comparative analysis of classification algorithms. Int Adv Res J Sci Eng Technol 8:193–201

16. Chaves L, Marques G (2021) Data mining techniques for early diagnosis of diabetes: a comparative study. Appl Sci 11(5)
17. Kareem A, Shi L, Wei L, Yongcai T (2020) A comparative analysis and risk prediction of diabetes at early stage using machine learning approach. Int J Future Generation Commun Netw 13:4151–4163
18. Islam MM, Ferdousi R, Rahman S, Bushra HY (2020) Likelihood prediction of diabetes at early stage using data mining techniques. In: Computer vision and machine intelligence in medical image analysis. Springer, Heidelberg, pp 113–125
19. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
20. Sekeroglu B, Emirzade E (2018) A computer aided diagnosis system for lung cancer detection using support vector machine. In: Jiang X, Chen Z, Chen G (eds) Third international workshop on pattern recognition, vol 10828. International Society for Optics and Photonics, SPIE, pp 143–150
21. Dougherty G (2013) Pattern recognition and classification. Springer, Heidelberg
22. Alpan K, Sekeroglu B (2020) Prediction of pollutant concentrations by meteorological data using machine learning algorithms. In: The international archives of photogrammetry, remote sensing and spatial information sciences, vol XLIV-4/W3-2020. Copernicus GmbH, pp 21–27
23. Schapire RE (2013) Explaining adaboost. In: Empirical inference. Springer, Heidelberg, pp 37–52
24. Friedman J (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232, 100204
25. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. arXiv arXiv:1603.02754
26. Cox D (1958) The regression analysis of binary sequences. J Royal Stat Soc: Ser B (Methodological) 20(2):215–232
27. Webb G (2010) Naïve Bayes. Springer US, Boston, MA, pp 713–714
28. Mucherino A, Papajorgji P, Pardalos P (2009) k-Nearest neighbor classification Springer, New York, pp 83–106

Chapter 28
Optimization of LoRa Networks Using Multi-armed Bandit Algorithms

Alexander Valach and Dominik Macko

1 Introduction

Scalability is defined by dictionary of computing as a term on how well a solution would sustain even when the problem increases [1]. In the terms of Internet of Things (IoT), it is equivalent to how well would devices be able to transmit and receive messages even when their numbers increase rapidly. This issue is addressed mainly by the simultaneous existence of heterogenous devices that use different technologies [2], but also similar physical layer frequencies. This results in inter-technology interference [3], and incapability of many protocols (usage of pure ALOHA-based solutions) to listen before transmission [4, 5]. Another issue is a usage of the license-free Industrial Scientific and Medical (ISM) band, which is jammed by many technologies. Two leading low-power wide area network (LPWAN) technologies (LoRa and Sigfox) use the ISM band for communication [6]. This only contributes to higher interference and increases a probability of a packet collision or a channel congestion.

Appropriate tools for the simulation of the physical layer properties in LoRa networks are missing [7, 8]. However, there has been some successful attempts to implement them [9–11]. They are considered essential building blocks in the deployment of LPWAN networks and a mandatory tool for further research in the field of effective communication and a rapid growth of a number of end devices.

To improve performance, we need to reduce collisions between messages, increase a packet delivery ratio (PDR), and ensure that a combination of parameters selection is battery efficient [12]. The latter is especially important in situations with a minimal number of collisions, high PDR, but high energy consumption. When communication

A. Valach (✉) · D. Macko
Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Ilkovičova 2, 842 16, Bratislava, Slovakia
e-mail: alexander.valach@stuba.sk

D. Macko
e-mail: dominik.macko@stuba.sk

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_29

parameters are set to the highest possible values, it leads to the shorter battery life. In some cases, there is no or limited possibility to recharge the battery; e.g., a sensor is put into an asphalt layer [13].

Effective communication parameters selection does not only refer to transmission in everyday environment with few devices. It is crucial to prepare IoT networks for rapid increase of connected devices. Semtech, the creator of LoRa, estimates that approximately 1.6 billion LPWAN devices will be connected in 2026 [14]. It means, we need to evaluate the performance of end devices in a harsh and dense environment, where a risk of collision increases. A node-based machine learning (ML) approach can potentially help to both mitigate collisions and save battery power [7, 15] by using a more effective process of communication parameters selection with only a limited knowledge of the environment [8, 16].

The rest of the paper is organized in the following way. Section 2 summarizes the recent research in the field of optimization of LoRa networks using ML algorithms. Section 3 introduces LoRa and LoRa@FIIT essentials focusing on the communication parameters. It further analyzes the different multi-armed bandit algorithms than can aid in mitigating collisions from the perspective of utilization in LoRa technology. Section 4 proposes a network testbed to evaluate a performance of Thompson sampling algorithm in a dynamic environment. In Sect. 5, the experimental results are presented. The paper is concluded in Sect. 6.

2 Related Works

One of the current challenges of the deployment of LoRa networks in densely populated urban areas with tall buildings and brick walls is to mitigate collisions. There has been a much research made in this field during the past years [2, 5, 17]. The researchers were focusing on several different aspects like an efficient communication parameters selection [7, 18–21], designing an energy-wise solution [22–24], and mitigating or predicting collisions [3, 4, 25–27].

A field of ML called a reinforcement learning (RL) has been found helpful in process of selecting communication parameters. Even in a dynamic environment where nodes join the network and leave it very often [7, 18]. In RL, a decision has to be made between an exploration and exploitation of environment with only a limited knowledge of the situation [28].

To simplify evaluation of various RL algorithms in LoRa networks, the LoRaMAB simulator was developed. It helped to show that the distributed learning outperforms simple heuristics in terms of network throughput [19]. The work [18] showed that even algorithms that do not take a switching (i.e., end nodes move between wireless access points) environment into account, namely upper confidence bound (UCB) and Thompson sampling (TS), achieve near-optimal performance even in non-stationary (i.e., rewards change in time) settings. Every RL algorithm proves to be more energy efficient than the current state of the art [7], primarily due to a collision reduction and possibility to quickly adapt to network changes.

A lot of research has been conducted in the recent years considering an optimization of LoRa technology and LoRaWAN protocol built on top of it. However, there are only a few papers addressing a utilization of RL algorithms considering devices that are constantly on the move and thus enforcing a non-stationary environment.

There is almost no paper considering a ML approach with other than LoRaWAN protocol. We proposed our own architecture to simulate a LoRa network and focus on the performance of LoRa@FIIT protocol, which was designed to be more energy-wise than LoRaWAN [29] and can become a promising solution for certain industry use cases; e.g., a quality of service is required.

3 Optimization of LoRa End Nodes Using Reinforcement Learning Techniques

In this section, we briefly introduce LoRa technology and LoRa@FIIT protocol and its network architecture derived from LoRaWAN industrial standard [30]. Furthermore, we briefly analyze multi-armed bandit algorithms that can be divided to three types depending on the environment they can adapt to [7].

3.1 LoRa Technology and LoRa@FIIT Protocol

LoRa is a physical layer modulation, also called LoRa PHY. It is a low-power wide area network (LPWAN) technology. It is very popular for battery-constrained devices due to its long-range communication, low power consumption and low deployment cost. It operates in the license-free ISM band [6, 31]. Transmitting in this band is free of charge, but it has major regulations. In LoRa, each device can occupy the medium only for certain time period. This time is called a duty cycle (DC) and is set depending on the frequency. In Europe, it is usually 1% of hour, which means a device can only transmit 36 s within one hour. The DC is refreshed after each hour [31].

The performance of LoRa networks depends on settings of each device. These settings can be dynamically adjusted and are called communication parameters (CP) [31, 32]:

1. **Carrier frequency (CF)** is also called a communication channel. When two devices use the same frequency at the same time, a collision occurs. There are also exceptions. When a capture effect (CE) is present [9] a message with higher (+6 dB) signal-to-noise ratio (SNR) value can be successfully retrieved [10]. Frequency is expressed in MHz and in Europe uses radio bands near 868 MHz [31].
2. **Transmission Power (TP)** has an impact on a battery lifetime. When an access point receives low-quality signal (below certain threshold), a node should increase

its TP. TP is expressed in dBm and can vary from -4 dBm to 20 dBm [17]. In the current implementation of network server, it can only be updated in 1 dBm steps [33].

3. **Spreading factor (SF)** has the most significant impact on communication efficiency. Lower SF value decreases communication range and time required to transfer message to the receiver (Time on Air, ToA), thus draining less battery power and decreasing a risk of collision, as the medium is occupied only for a short time interval. Higher SF value increases distance, a message can reach with lower SF values, and significantly (twice, by single-level SF increment) increases ToA, thus draining more battery power [6, 10].
4. **Coding Rate (CR)** expresses a number of redundant bits in LoRa messages. Possible values are 4/5, 4/6, 4/7, and 4/8. When CR is set to 4/5, 4 bits are encoded with 5 bits allowing communication to endure a short interference during transmission [31, 32]. A procedure of possible short interference recovery is called forward error correction (FEC). The value of 4/5 is the only possible choice in current LoRa@FIIT implementation [34]. However, it can be dynamically changed by making minor changes to the network server [33].
5. **Bandwidth (BW)** sets up a frequency width (expressed in kHz) for communication channel [29]. LoRa can operate with 125 kHz, 250 kHz, or 500 kHz bandwidth. Higher bandwidth values indicate higher data rates, but lower receiver sensitivity. If not explicitly written, a 125 kHz BW is assumed to be used in the whole paper.

LoRa@FIIT is a Media Access Control (MAC) layer protocol designed to overcome drawbacks of LoRaWAN. It uses corrected block tiny encryption algorithm (XXTEA) specifically designed for embedded devices with low memory and computational power [35]. It has a built-in quality of service (QoS) mechanism and supports three different acknowledgment types (no, optional, or mandatory ack). It requires significantly lower (42%) overhead for sending 1 B of data compared to LoRaWAN. However, it does not support roaming, so the owner of the network must be the same as the owner of the end devices [29].

LoRa@FIIT network architecture is derived from LoRaWAN architecture and illustrated in Fig. 1. LoRaWAN is another MAC layer protocol, widely deployed in LoRa networks and maintained by the LoRa Alliance [30, 36–38]. LoRa@FIIT networks consist of several types of devices:

1. **End node (EN)** is usually a battery-constrained device with limited computational power and memory. It is designed to measure a certain characteristic of an environment, e.g., air quality, humidity, or atmospheric pressure, and send the measured data via LoRa technology to nearest access points.
2. **Access point (AP)** receives LoRa packets from end nodes, extracts the content of the messages and sends it to the network server in a JSON format using Secure TCP for IoT (STIoT) [29].
3. **Network server (NS)** is a central decision-making point of LoRa@FIIT networks. It stores information about device duty cycle and manages communication parameter selection (SF and TP). It processes data from APs and manages end nodes'

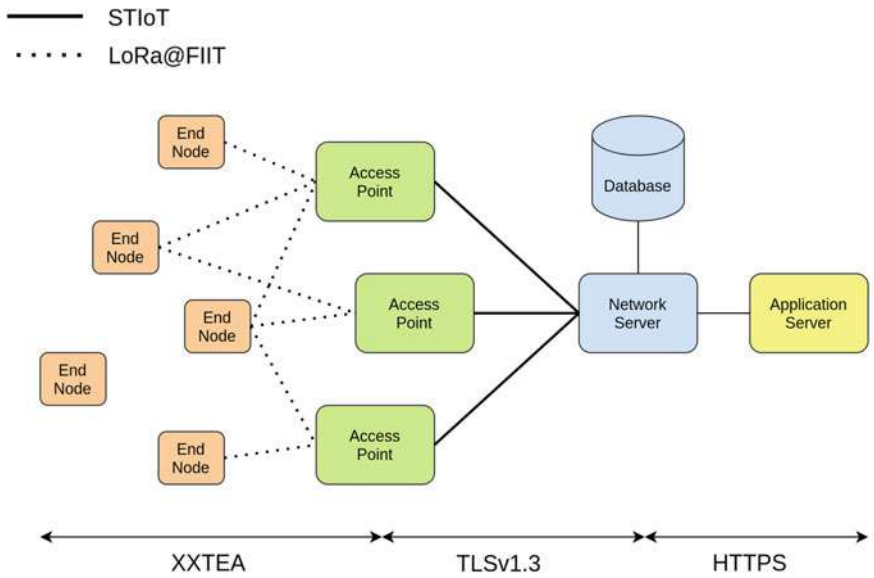


Fig. 1 Typical LoRa@FIIT network architecture derived from LoRaWAN architecture [30]

communication parameters selection process based on previous RSSI and SNR values.

4. **Application server (AS)** displays a collected data from the end nodes to the customers.

LoRa@FIIT introduces several message types that can be processed differently by network components. Message types are listed below:

1. **Data messages** carry an application payload with measured data.
2. **Emergency messages** are sent when critical conditions are measured or the level of the observed characteristics is below or above the acceptable value, e.g., water level too high or a blood oxygen saturation is too low.
3. **Registration messages** are sent when a device wants to join an existing network. They carry a Diffie–Hellman keys to derive a shared secret.
4. **Hello messages** are used as a keepalive mechanism and require an acknowledgment from the network server [29].

The current implementation of LoRaWAN or LoRa@FIIT both uses adaptive data rate algorithm (ADRA) to configure the proper communication parameters on the ENs. CPs are calculated on a network server and distributed to end devices.

The difference between best and worst CP selection combination can lead to 47% battery-life loss due to non-optimal decision-making process or lack of information from the network server [15]. The study shows that CP selection is not only crucial for further development, but it is also mandatory to provide long battery life, in term of several month to years.

3.2 Multi-armed Bandit Algorithms

Problem of selecting appropriate communication parameters can be compared to a problem of choosing a right arm with an unknown reward with goal of getting the largest cumulative reward [39]. A bandit, which is pulling arms, is not provided with the reward values and thus is forced to try different arms (exploration) or pull the exact same arm that gave him the largest discovered reward multiple times (exploitation). This problem is called a multi-armed bandit problem (MABP). Algorithms developed to cope with a MABP are called multi-armed bandit algorithms (MABA) [18].

MABAs are the part of a reinforcement learning, a field of machine learning. MABAs have been used to mitigate collisions and make the nodes more independent from network server in various network simulations [7, 8, 18, 40].

In a traditional MABP, we are given several slot machines, also called bandits, with the goal to minimize the cumulative regret [41] with no prior knowledge of which machine produces the highest reward. Depending on the characteristics, we differentiate between several environments [18]:

1. **Stationary environment** where rewards do not change in time. Once an optimal strategy is discovered, the low cumulative regret is achieved. However, this setup is very unrealistic in a dynamic environment of IoT.
2. **Non-stationary (dynamic) environment** where rewards do not change in time. A change point detection mechanism is required to signal nodes switching to a different environment (change of rewards).
3. **Adversarial environment** where rewards are set by an adversary and change when a new adversary is introduced. It is similar to the dynamic rewards but relies on adversary rather than a change point detection.

3.2.1 Stationary Multi-armed Bandit Algorithms

These algorithms are designed to deal with a MABP when rewards do not change in time. They are usually easy to implement as there are no additional requirements, only a reward calculation and pick-up of arm.

Upper Confidence Bound (UCB). Defines a confidence level for each arm. It is naive and greedy approach and depends heavily on the first draw [18]. This algorithm is designed for stationary environments. It proves to be the worst algorithm in non-stationary environment, based on the results from mathematical simulations [7]. The cumulative regret also tends to rise with larger number of trials [39].

A reward for each arm (combination of communication parameters) is computed based on the following formula:

$$\text{REWARD}_i = \mu_i + \sqrt{\frac{2\ln(n)}{n_i}} \tag{1}$$

In the formula above, μ_i represents the current (nth round) average reward returned from arm i, n represents the number of rounds (trials for any arm) so far and n_i represents number of pulls for arm i, e.g., how many times has the combination of communication parameters been selected.

Confidence bound is used as a mean to deal with an exploration versus exploitation dilemma. If multiple arms have the same reward, which is considered an initial state of the network, an arm with a higher confidence bound is drawn. After each draw, the confidence bound is decreased and the previously unexplored arms are preferred [18]. This strategy follows an optimistic approach in the favor of the uncertainty.

Thompson sampling (TS). It is a probability matching algorithm [42] that chooses an arm based on the shape parameters: α (successful attempts) and β (failed attempts). It uses Bayesian tools, assuming a prior distribution of each arm [39]. Unlike UCB, this is a sampling-based probabilistic approach and proved to achieve smaller cumulative regret than UCB [7, 16, 42].

The TS was designed to deal with a Bernoulli bandit problem where a reward is set to 1 or 0 [39]. Instead of a greedy approach, where a sub-optimal solution is usually found because the algorithm sticks to a local maximum, it uses a theta parameter. This parameter forces selection of a locally sub-optimal solution, which can lead to a globally optimal solution. The fundamentals of a TS algorithm are presented in Fig. 2 [39]:

In the above-mentioned algorithm, S_j and F_j represent a number of successful draws (a packet was delivered) and failures (a packet was lost), respectively. T is the number of total trials and N represents a number of arms (communication parameters combination) to choose between. θ_j is an array of rewards for each arm [39], which is unknown to the end node. $param_j$ represents arms (combination of SF and TP). Furthermore, a beta distribution is used to model the mean of each arm. It is considered a standard practice [39].

Require: α, β shape parameters from Beta distribution
  Sj=0, Fj=0
  for t=1, ..., T **do**
    for j=1, ..., N **do**
      Draw arm_j according to $Beta(S_j + \alpha, F_j + \beta)$
    end for
    Select $param_j = argmax_j \theta_j$ and get the reward r
    if $r = 0$ **then**
      $F_j = F_j + 1$
    else
      $S_j = S_j + 1$
    end if
  end for

Fig. 2 Pseudocode for Thompson sampling

A search space also has an important role. When it is very expansive, a TS is trying to prove a single option to be more efficient and pulls it several times. On the other hand, UCB uses an optimistic approach and is willing to take a quick but uncertain win option without further investigation.

3.2.2 Non-stationary Multi-armed Bandit Algorithms

In the non-stationary environment rewards change in time. This is more realistic environment than a stationary one, as it considers the change in the environment, also called a switching. The process of identifying a switching environment is also called a change point detection. We differentiate between two types of switches [7]:

1. **Global switching** where all the rewards are changed. This indicates a change in the environment; e.g., a mobile node moves from one area to another one, where network conditions are different.
2. **Per-arm switching** where only a reward for single arm has changed. It can indicate a congestion on a certain channel (SF). A mechanism to notify all the affected ENs is required [43].

Global Switching Thompson Sampling with Bayesian Aggregation (STSBA). It is a stochastic MABA. To the authors' best knowledge, it has not been evaluated in a real-world environment yet. Only mathematical simulation that take into account several physical modulation constraints were performed, both for LoRa [7] and for other wireless networks [16] facing a similar problem. It proved to be the most efficient from the stochastic algorithms, because its adaptability to non-stationary settings and a fast change point detection using a Bayesian aggregation [7].

It is a modified version of TS algorithm designed for non-stationary settings and is combined with a Bayesian online change point detector [42]. It is also an expert-based algorithm, meaning there is a TS procedure starting at time t. A Bayesian aggregation (BA) is used on the most likely expert (according to its weight), and then, a TS is run to choose an arm. After choosing an arm, weight for corresponding arm is updated [7].

In [7, 16], the researchers proved that a STSBA slightly outperformed TS at the cost of higher processor utilization and thus a higher energy consumption. TS performed similarly to STSBA but requires less computational power, which has a significant impact on battery life for low-power devices.

UCB has also its switching alternative, called Sliding Window UCB (SWUCB). However, it was outperformed by TS (stationary algorithm) and performs similarly to UCB [7], and thus, we will not examine it closer in this paper.

3.2.3 Adversarial Multi-armed Bandit Algorithms

Exponential Weights for Exploration and Exploitation (EXP3) is an adversarial MABA. It theoretically performs worse than stochastic bandits (TS and UCB) [16].

However, it can achieve similar results using any settings and thus providing similar results for different environments. It can also handle non-stationary settings [7, 19], which is essential for a real-world scenario with mobile nodes. EXP3.S is an enhanced version of EXP3 algorithm and performs better than the original one [22].

Disadvantage of these algorithms is the long conversion time. For EXP3 up to 200 kHours and for EXP3.S up to 20 kHours. Each new node should be able to communicate efficiently after conversion time [10]. Therefore, an improved solution is required to shorten the time, as it is not sufficient in dynamic and harsh network environment of license-free LPWANs.

3.2.4 Performance of Multi-armed Bandit Algorithms

The experiments showed that an ADRA was outperformed by MABAs, both in terms of an energy consumption and in terms of PDR. The energy consumption for a single communication parameter selection is lower in ADRA, as no additional overhead is required. However, the overall energy consumption is usually higher due to the node not being able to change CPs without a network server. In MABAs, an overhead is required for a CP selection, but an overall energy consumption can be lower due to an ability to choose alternative CPs without the interception from the network server.

When it comes to a comparison of MABAs, adversarial bandits (EXP3 and EXP3.S) are generally outperformed by the stochastic bandits [7, 16]. This is true especially when mobile nodes are introduced [7]. The stationary bandits (TS and UCB) are outperformed by the switching environment bandit (STSBA, SWUCB). The switching bandits are the most difficult to implement and tune properly, so an additional research in this field is also required.

The experiments also showed that the results of TS, a stationary bandit algorithm, are similar to its switching alternative [7]. This is quite surprising because a TS was not designed for a switching environment [39]. The lowest energy consumption was achieved with UCB and SWUCB. The lowest packet loss and overall total cost compared to ADRA has a STSBA, which is an improved version of TS for switching environments, modified by the researchers in [7]. ADRA naturally has the worst performance in examined scenarios [7, 11].

4 The Proposed Network Testbed

One of the most important aspects before a network is deployed are the performance metrics and ability to perform well even in a harsh real-world environment with densely located nodes and dynamic environment.

To simplify an assertion of different algorithms, we propose our own architecture, illustrated in Fig. 3. We have developed a simple STIoT (protocol for communication between APs and NS) packet generator [36] that simulates an operation of a single AP. This single AP simulator connects to a deployed remote NS as would be the case

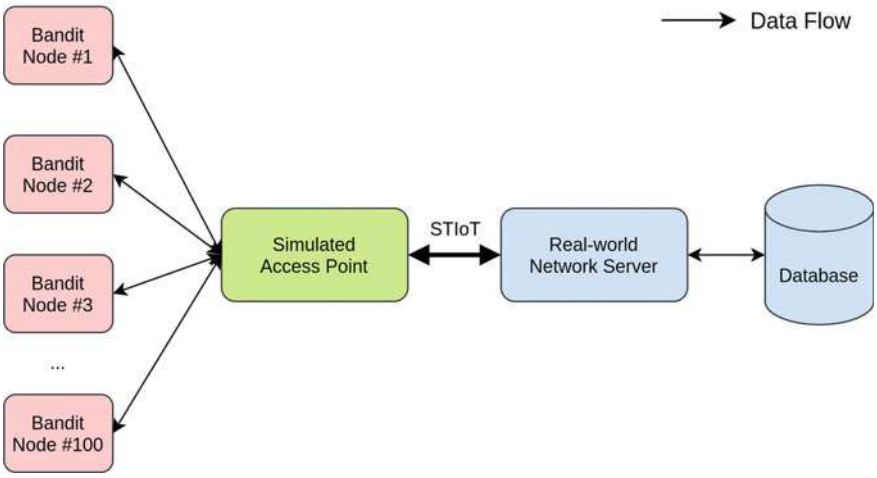


Fig. 3 Network testbed architecture consisting of several ENs, simulated AP and real NS

in a real-world scenario. Existing simulators would not be appropriate as they do not support LoRa@FIIT protocol and its architecture.

In the simulator, ENs do have their own configuration settings. However, LoRa physical properties are only calculated based on the rules described in the following subsection. As each EN support a communication parameter selection using TS and UCB (bandit algorithms), it is called a bandit node (BN). This term helps to differentiate the traditional ENs, that depend on the decision from NS, from the proposed BNs, which use MABAs.

4.1 Rules for the STIoT Packet Generator

The experiments will be performed using our own STIoT packet generator [44] and a real LoRa network server [33]. As the results of TS are very promising for mobile nodes, despite being primarily developed for a stationary setting, we will focus on implementing TS in our generator and simulate the process of up to 100 ENs connected to a single AP. Despite being evaluated only using simulations, some aspects from a real-world scenario were also implemented in the simulator and are listed below [44]:

1. **Rules for collisions and packet loss.** There are several types, which we refer to as a collision.
   (a) A frame is transmitted on the same SF as another frame at the same time or their receiving times overlap with each other.

(b) When a RSSI value is below the receiver's sensitivity, it is discarded and considered a collision.
(c) SNR value is randomly generated. If it is between certain threshold, it is considered unreadable (too much interference), cannot be demodulated, and is discarded.

2. **Movement of the end nodes.** Every second, each of the nodes is moving on the square area of 100 km^2. The signal is weakened by the free-space path loss (FSL) formula presented below. CF is a frequency in Hz and D is a distance in km. When we use Hz and km as units, C becomes close to 32.5:

$$\mathrm{FSL} = C + 20 \times \log \mathrm{CF} + 20 \times \log D \tag{2}$$

The directions for horizontal and vertical movements are randomly chosen at the beginning of the experiments and can only be positive or negative. The direction

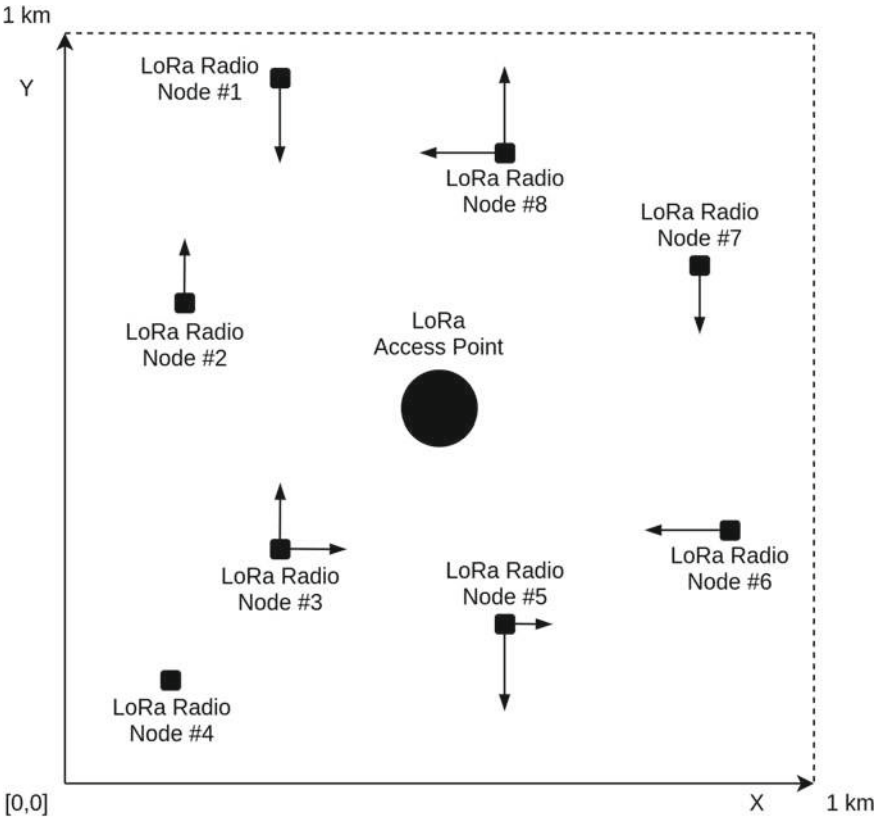


Fig. 4 LoRa access point is placed in the center of the observed area with end nodes moving constantly

Table 1 Experiment settings for both algorithms

Characteristic	Value
Frequency	866.1 MHz
Coding rate	4/5
Bandwidth	125 kHz
Duration of a single run	7 h
Number of access points	1
Number of end nodes	up to 100
Sleep time after each uplink	120 s
Payload length (range)	16–52 B
Number of runs for each simulation	5
Emergency message ratio	98.2%

is changed every time a device is getting close to a wall at the edge of the area. Every second, a 0–1 m movement of node can be made in both horizontal and vertical direction, resulting in 8 directions in total. This mechanism was design to force a frequent (but predictable) change of CPs in the environment.

3. **Placement of the nodes.** Each EN is placed randomly on the $[x, y]$ coordinates following a uniform distribution. Only a single AP can be found within the area, and it is placed in its center at coordinates [5000, 5000]. Distances are calculated in meters. The placement and the movement of the nodes around AP are presented in Fig. 4.
4. **Communication parameters.** Only SF and TP values can be updated by ENs or NS. Other values are fixed (presented in Table 1) during the experiments to simplify the process, as we focus mainly on the comparison of UCB and TS. The other parameters are CR, BW, and FREQ, which are set to 4/5, 125 kHz, and 866 MHz, respectively.
5. **Transmissions.** After sending an uplink message, two receiving windows are subsequently opened. However, only if the message type was set to emergency, a device is waiting for an acknowledgment. Otherwise, it is considered lost and is retransmitted.
6. **Retransmissions.** After three unsuccessful attempts to transmit an important (emergency or registration) message, a device enters a sleep mode. If a message does not require an acknowledgment, a retransmission does not occur. When an important data is being transmitted, a message can be retransmitted till the acknowledgment is received, or it is considered lost.
7. **Received signal strength indicator (RSSI)** is calculated based on the following formula, where $\mathrm{TP_{TX}}$ is a TP of EN, G_{TX} is a transmitter antenna gain, FSL is a free space loss and G_{RX} is an antenna gain of the receiver.

$$\mathrm{RSSI} = \mathrm{TP_{TX}} + G_{\mathrm{TX}} - \mathrm{FSL} + G_{\mathrm{RX}} \tag{3}$$

Cable loss at the transmitter and the receiver are both negligible in the case of very short (6 cm) low-loss cable.

8. **Payload.** Each packet contains a base64 encoded string with information about its current position within the simulation area. Those data are decoded and visualized during the evaluation of the experiments.
9. **Duty cycle restrictions.** Only a few papers have considered a duty cycle constraint. Our simulator has a built-in duty cycle management mechanism for each node. It calculates Time on Air [34] for each packet and subtracts this value from the actual duty cycle. Time on Air (ToA) is calculated using the following equations, where T_S is the time required to send a single symbol and N_{SYM} is number of symbols:

$$\mathrm{ToA} = T_S \times N_{\mathrm{SYM}} \tag{4}$$

$$T_S = \frac{2^{\mathrm{SF}}}{\mathrm{BW}} \tag{5}$$

5 Simulations and Results

We have compared results from experiments for UCB and TS algorithms. The current state of the art, ADR, was compared to UCB in authors' previous work [11]. It was clearly outperformed by UCB in each aspect (SF distribution, collision rate, PDR, active time). A comparison of performance of ADR to TS for mobile nodes would bring very similar results as in [11]. For this exact reason, only UCB and TS are compared in this paper.

Each simulation ran for five times. Each 100 end nodes were already registered before simulations. In each run, the ENs were joining network only when there was no other traffic to process by AP. This mechanism serves to avoid depletion of DC of a single AP. Connecting all 100 nodes within first hour leads to a depletion of DC, based on the results of authors earlier experiments [45]. However, not every end node was able to communicate with the AP due to a random placement of ENs possibly out of range and inability to find proper communication parameters.

The summary of common settings used in a network testbed is presented in Table 1. Each experiment lasted 7 h. Each node was randomly placed within a pre-defined area and sent 16–52 B (mean value of 43) long payload every 2 min.

Majority of messages (98.2%) had mandatory acknowledgment (emergency messages). They served as a good information source for reinforcement learning algorithms.

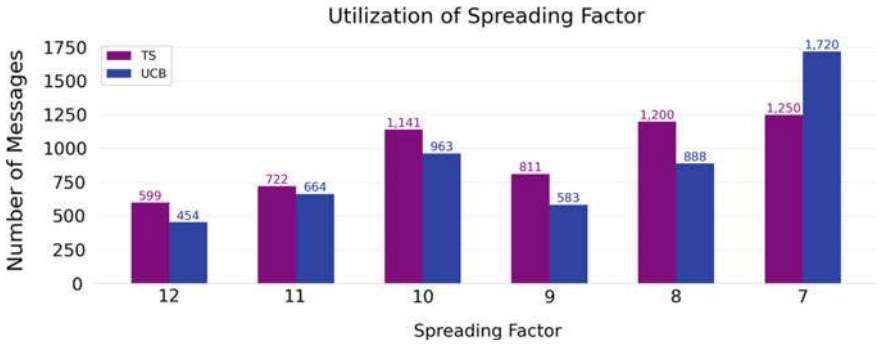


Fig. 5 Comparison of a number of messages sent on each SF

A list of evaluated performance metrics follows:

1. **SF distribution.** This metrics shows whether a majority of the end nodes transmit on the same SF. In that case, a risk of collision is higher because sub-channel (SF) becomes congested.
2. **Collision count.** A number of total collisions for each EN.
3. **PDR.** A number of successfully delivered messages to a number of all transmitted messages.
4. **Active mode ratio.** It represents a total time spent outside sleep mode to total uptime of each device.
5. **Total connected end nodes.** Number of ENs that were able to transmit at least 1 message to the NS.

For SF and path rendering, only messages that were received by NS are considered. For collision rate and PDR, also data from end nodes were evaluated as an estimation of exact packet loss can be challenging from the perspective of NS only.

Spreading factor was fairly distributed both in case of TS and UCB, as illustrated in Fig. 5. In UCB, there is a big difference between messages transmitted on SF 7 and SF 8. This is probably due to UCB sticking to the local maximum (SF 7). TS utilized both SF 7 and SF 8 almost equally. Equal distribution of SF is an intended behavior, as it is a way to avoid congestion on sub-channels (SF) and thus indirectly mitigate collisions. Other SF was fairly utilized with larger values for TS, as it has higher overall PDR.

Most of the ENs have up to five collisions, illustrated in Fig. 6. Higher numbers of collisions than 14 are left out for clarity. Overall, values were similar for both algorithms with UCB having lower mean value (4) compared to TS (5).

The most significant difference between UCB and TS is seen in PDR, illustrated in Fig. 6. UCB resulted in some nodes having very low values (below 40%) and some nodes having higher values (over 80%). TS has only 1 node that was unable to join the network. It was unable to send or receive a downlink registration message.

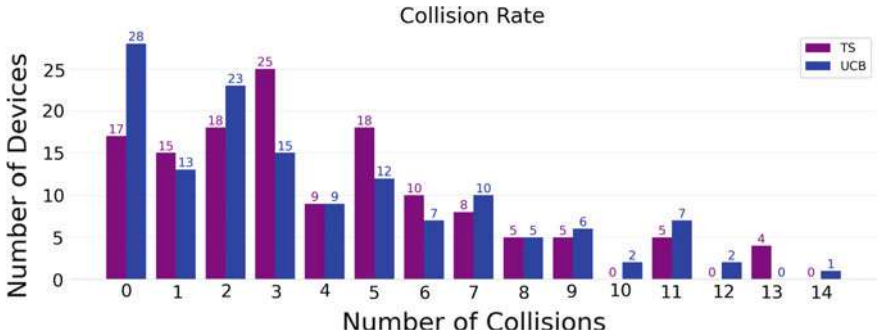


Fig. 6 Comparison of a number of collisions

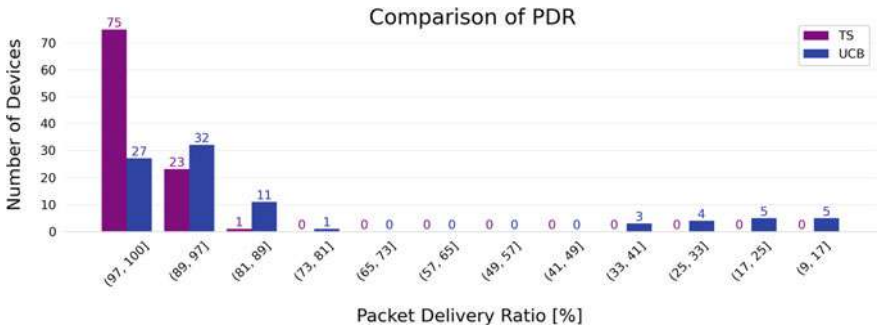


Fig. 7 Comparison of PDR values

UCB has 12 nodes that were unable to transmit messages properly, which reduces its performance. Most of the ENs for TS have high PDR values. Thus, in PDR, TS outperforms UCB (Fig. 7).

Each of the ENs was constantly moving, within a 100 km^2 area. At the beginning of the simulation, each end node was randomly assigned positive or negative direction for both x-axis and y-axis movements. This mechanism lead to some of the nodes getting closer to the AP and some of them getting away from the AP. Rendered paths for both algorithms are presented in Fig. 8.

A brief summary of performance metrics and characteristics for both algorithms is presented in Table 2. Mean value of PDR was higher for TS, but it requires also slightly longer active time for communication parameters selection overhead. This is justified by more complex process of selecting an arm for TS than UCB.

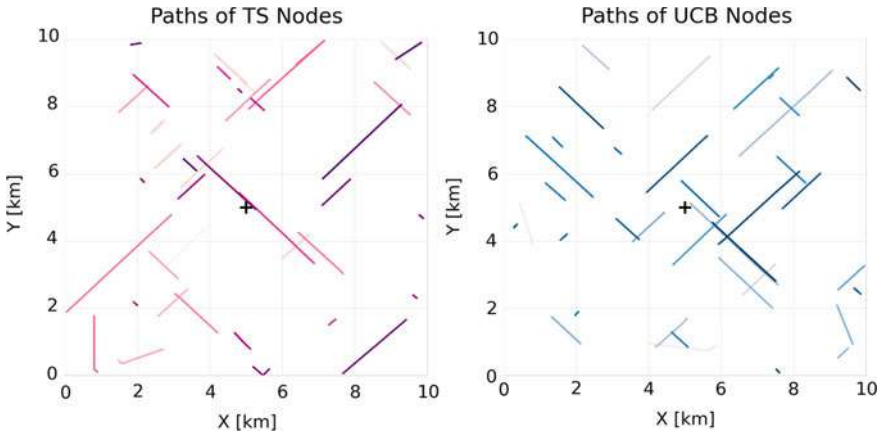


Fig. 8 Rendered paths of end nodes for both algorithms

Table 2 Performance metrics and characteristics for both evaluated algorithms

Characteristic	UCB	TS
Total connected end nodes	88	99
Collisions per end node (mean)	4	5
Packet delivery ratio (mean)	80.85%	97.69%
Time spent in active mode (mean)	2.50%	3.03%

6 Conclusion

LoRa is an emerging technology that has potential for LPWAN real-world deployment. However, it is not prepared for the harsh network environment of smart cities, where hundreds of nodes are connected. Adaptive data rate does not perform well with mobile nodes. As a result, the number of collisions in the network is rising, which is draining a limited power supply of end nodes. We propose replacing adaptive data rate algorithm with multi-armed bandit alternatives to be more effective both in terms of packet delivery (and overall network reliability) and energy efficiency. We propose a network architecture consisting of a real-world network server and simulated access point with around one hundred end nodes constantly on the move to evaluate simple, yet promising Thompson sampling algorithm. In the future, we plan to add an inter-SF collisions to make a scenario closer to reality and test other multi-armed bandit algorithms, not only using a simulator but using a real hardware and physical obstacles.

Acknowledgements This research was partially supported by the Slovak University of Technology in Bratislava and the Ministry of Education, Science, Research and Sport of the Slovak Republic, within the project ITMS 313021X329, co-funded by the European Regional Development Fund.

References

1. Howe D. FOLDOC—computing dictionary. https://foldoc.org/. Accessed: 08.06.2021
2. Akpakwu GA et al (2018) A survey on 5G networks for the Internet of Things: communication technologies and challenges. IEEE Access 6:3619–3647. https://doi.org/10.1109/ACCESS.2017.2779844
3. Pullmann J, Macko D (2010) Increasing energy efficiency by minimizing collisions in long-range IoT networks. In: 2019 42nd international conference on Telecommunications and Signal Processing (TSP), pp 178–181. https://doi.org/10.1109/TSP.2019.8768844
4. Rochester EM et al (2020) Lightweight carrier sensing in LoRa: implementation and performance evaluation. In: ICC 2020—2020 IEEE International Conference on Communications (ICC), pp 1–6. https://doi.org/10.1109/ICC40277.2020.9149103
5. Hou Y, Liu Z, Sun D (2020) A novel MAC protocol exploiting concurrent transmissions for massive LoRa connectivity. J Commun Netw 22(2):108–117. https://doi.org/10.1109/JCN.2020.000005
6. ABI Research. LoRaWANR and NB-IoT: competitors or complementary? https://lora-alliance.org/wp-content/uploads/2020/11/cr-lora-102_lorawanr_and_nb-iot.pdf
7. Kerkouche R et al (2018) Node-based optimization of LoRa transmissions with multi-armed bandit algorithms. In: 2018 25th International Conference on Telecommunications (ICT), pp 521–526. https://doi.org/10.1109/ICT.2018.8464949
8. Sandoval RM, Garcia-Sanchez A-J, Garcia-Haro J (2019) Optimizing and updating LoRa communication parameters: a machine learning approach. IEEE Trans Netw Service Manage 16(3):884–895. https://doi.org/10.1109/TNSM.2019.2927759
9. To T-H, Duda A (2018) Simulation of LoRa in NS-3: improving LoRa performance with CSMA. In: 2018 IEEE International Conference on Communications (ICC), pp 1–7. https://doi.org/10.1109/ICC.2018.8422800
10. Ta D-T et al (2019) LoRa-MAB: a flexible simulator for decentralized learning resource allocation in IoT networks. In: 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), pp 55-62. https://doi.org/10.23919/WMNC.2019.8881393
11. Valach A, Macko D (2021) Improvement of LoRa communication scalability using machine learning based adaptiveness. https://doi.org/10.36227/techrxiv.14627028.v1, https://www.techrxiv.org/articles/preprint/Improvement_of_LoRa_Communication_Scalability_using_Machine_Learning_Based_Adaptiveness/14627028/1
12. Bor M, Roedig U (2017) LoRa transmission parameter selection. In: 2017 13th international conference on Distributed Computing in Sensor Systems (DCOSS), pp 27–34. https://doi.org/10.1109/DCOSS.2017.10
13. 4TU.Federation. Smart sensors in asphalt. https://www.4tu.nl/bouw/en/Lighthouse/Smart
14. Lorrain R. The future of 5G and LoRaWAN. https://blog.semtech.com/the-future-of-5g-and-lorawan-friends-or-foes
15. Gupta A, Fujinami M (2019) Battery optimal configuration of transmission settings in LoRa moving nodes. In: 2019 16th IEEE annual Consumer Communications Networking Conference (CCNC), pp 1–6. https://doi.org/10.1109/CCNC.2019.8651707
16. Dakdouk H et al (2018) Reinforcement learning techniques for optimized channel hopping in IEEE 802.15.4-TSCH networks. In: Proceedings of the 21st ACM international conference on modeling, analysis and simulation of wireless and mobile systems. MSWIM '18. Association for Computing Machinery, Montreal, QC, Canada, pp 99–107. isbn: 9781450359603. https://doi.org/10.1145/3242102.3242110
17. Sandoval RM et al (2018) Optimal policy derivation for transmission duty-cycle constrained LPWAN. IEEE Internet Things J 5(4):3114–3125. https://doi.org/10.1109/JIOT.2018.2833289.
18. Bonnefoi R et al (2018) Multi-armed bandit learning in IoT networks: learning helps even in non-stationary settings. In: Marques P et al (eds) Cognitive radio oriented wireless networks. Springer International Publishing, Cham, pp 173–185. isbn: 978-3-319-76207-4

19. Ta D-T et al (2019) LoRa-MAB: curce allocation approach for LoRaWAN. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9013345
20. Sallum E et al (2020) Performance optimization on LoRa networks through assigning radio parameters. In: 2020 IEEE International Conference on Industrial Technology (ICIT), pp 304–309. 10.1109/ICIT 45562.2020.9067310
21. Valach A, Macko D (2020) Optimization of LoRa devices communication for applications in healthcare. In: 2020 43rd international conference on Telecommunications and Signal Processing (TSP), pp 511–514. https://doi.org/10.1109/TSP49548.2020.9163432
22. Li Y, Yang J, Wang J (2020) DyLoRa: towards energy efficient dynamic LoRa transmission control. In: IEEE INFOCOM 2020—IEEE conference on computer communications, pp 2312–2320. https://doi.org/10.1109/INFOCOM41043.2020.9155407
23. Orfanidis C et al (2019) Towards battery-free LPWAN wearables. In: Proceedings of the 7th international workshop on energy harvesting and energy-neutral sensing systems. ENSsys'19. Association for Computing Machinery, New York, NY, USA, pp 52–53. isbn: 9781450370103. https://doi.org/10.1145/3362053.3363488
24. Bouguera T et al (2018) Energy consumption model for sensor nodes based on LoRa and LoRaWAN. In: Sensors 18:2104. https://doi.org/10.3390/s18072104
25. Cui S, Joe I (2020) Collision prediction for a low power wide area network using deep learning methods. J Commun Netw 22(3):205–214. https://doi.org/10.1109/JCN.2020.000017
26. Ishida Y, Nobayashi D, Ikenaga T (2018) Experimental performance evaluation of the collisions in LoRa Ccmmunications. In: 2018 international conference on Computational Science and Computational Intelligence (CSCI), pp 1032–1035. https://doi.org/10.1109/CSCI46756.2018.00200
27. Moy C, Besson L (2019) Decentralized spectrum learning for IoT wireless networks collision mitigation. In: 2019 15th international conference on Distributed Computing in Sensor Systems (DCOSS), pp 644–651. https://doi.org/10.1109/DCOSS.2019.00117
28. Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. Mach Learn 47(2–3): 235–256. issn: 0885-6125. https://doi.org/10.1023/A:1013689704352
29. Perešíni O, Krajčovič T (2017) More efficient IoT communication through LoRa network with LoRa@FIIT and STIOT protocols. In: 2017 IEEE 11th international conference on Application of Information and Communication Technologies (AICT), pp 1–6. https://doi.org/10.1109/ICAICT.2017.8686837
30. LoRa Alliance. LoRaWAN R specification v1.0.4. https://lora-alliance.org/resource_hub/lorawan-104-specification-package/
31. Semtech. What are LoRa R and LoRaWAN R. https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan
32. Hill K, Gagneja KK, Singh N (2019) LoRa PHY range tests and software decoding—physical layer Security. In: 2019 6th international conference on Signal Processing and Integrated Networks (SPIN), pp 805–810. https://doi.org/10.1109/SPIN.2019.8711682
33. Perešíni O, Cagíň K, Valach A. LoRa Network Server. https://github.com/alexandervalach/lora-network-server
34. Simon Stefunko. LoRa@FIIT Arduino library. https://github.com/alexandervalach/lorafiit-library
35. Galas EM, Gerardo BD (2019) Feasibility assessment on the implementation of the enhanced XXTEA on IoT devices. In: 2019 IEEE 9th International Conference on System Engineering and Technology (IC-SET), pp 178–182. https://doi.org/10.1109/ICSEngT.2019.8906473
36. LoRa Alliance (2020) LoRaWAN certification protocol specification TS009-1.0.0
37. LoRa Alliance (2020) LoRaWAN 1.0.4 end device RF performance for all regions V1.0
38. LoRa Alliance. LoRaWAN TS1-1.0.4 L2 specification
39. Chapelle O, Li L (2011) An empirical evaluation of Thompson sampling. In: NIPS
40. Gloria A et al (2019) LoRa transmission power self configuration for low power end devices. In: 2019 22nd international symposium on Wireless Personal Multimedia Communications (WPMC), pp 1–6. https://doi.org/10.1109/WPMC48795.2019.9096197

41. Alami R, Maillard O-A, Feraud R (2017) Memory bandits: a Bayesian approach for the switching bandit problem. In: NIPS 2017—31st conference on neural information processing systems
42. Mellor J, Shapiro J (2013) Thompson sampling in switching environments with Bayesian online change detection. In: Carvalho CM, Ravikumar P (eds) Proceedings of the sixteenth international conference on artificial intelligence and statistics, vol 31. Proceedings of Machine Learning Research. PMLR, Scottsdale, Arizona, USA , pp 442–450. http://proceedings.mlr.press/v31/mellor13a.html
43. Sandoval Ruben M et al (2020) Deriving and updating optimal transmission configurations for Lora networks. IEEE Access 8:38586–38595. https://doi.org/10.1109/ACCESS.2020.2973252
44. Valach A (2021) LoRa@FIIT access point and end nodes simulator. https://github.com/alexandervalach/lora-ap-sim
45. Valach A (2021) LoRa notebooks. https://github.com/alexandervalach/lora-notebooks

Chapter 29
System for Management and Visualization of LoRa Network Components

Daniel Hroš and Alexander Valach

1 Introduction

Today we can see a substantial increase in the use of IoT devices. At the end of 2020, according to [1], around 11.7 billion IoT devices connected to the Internet were active worldwide. By 2030, the number of devices is estimated to reach 30 billion [1].

With a large number of devices comes the need to process the information they produce. IoT devices are often designed to operate for several years without charging. The combination of processing a large volume of information while also wanting to minimize the energy consumption of equipment creates a room for improvement of the current solutions. For this reason, we propose our data visualization and evaluation solution of LoRa@FIIT networks. LoRa@FIIT is a protocol, based on LoRa technology, operating on link layer of Open Systems Interconnection (OSI) model [2].

The rest of the work is organized as follows. Section 2 examines the existing solutions and analyzes them. Section 3 acquaints reader with LoRa technology and presents the design of our solution together with design of the screens. Section 4 discusses the method of verification of the proposed solution, and Sect. 5 summarizes the paper.

D. Hroš (✉) · A. Valach
Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Ilkovičova 2, 842 16, Bratislava, Slovakia
e-mail: xhrosd@stuba.sk

A. Valach
e-mail: alexander.valach@stuba.sk

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_30

2 Physical Modulation of LoRa and LoRaWAN MAC Protocols

LoRa is a wireless communication technology promoted by the LoRa Alliance [3]. The name LoRa is derived from the phrase Long Range. LoRa is a technology operating on the physical layer of the OSI model. LoRa uses Chirp Spread Spectrum (CSS) modulation to encode symbols, the signal of which is modulated by chirp pulses. Chirp represents the frequency of a signal that decreases and increases with time [4].

Devices communicating via LoRa technology use various communication parameters:

- **Spreading Factor (SF)**: The size of the SF affects the number of coded characters per symbol. In Europe, it may be between 7–12 [5]. The number of characters we can encode is equal to 2^{SF} [6]. The larger the SF, the slower the transmission, but the signal range increases [4].
- **Bandwidth (BW)**: Number of symbols occupying the spectrum [6].
- **Coding Rate (CR)**: Message coding rate represents a number of redundant bits used for error correction. For example, a value of 4/5 means that every 5th bit is used to check the previous four bits [7].
- **Transmission power (TP)**: Transmission power of end device [7].
- **Carrier Frequency (CF)**: Set to several values around 868 MHz in Europe and 915 MHz in the USA [7].
- **Packet Delivery Ratio (PDR)**: Value in % of the occupied gateway channel. If the PDR value is 90%, it means that 90% of the packets reached the network server.

The LoRaWAN Media Access Control (MAC) protocol is implemented on the top of the LoRa modulation. The LoRaWAN protocol is formalized in the LoRaWAN specification [8] and maintained by the LoRa Alliance [3]. LoRa operates in the unlicensed radio spectrum (ISM Band, Industrial Scientific and Medical Band). For LoRa frequencies, each country has specified its regional restrictions called Regional Parameters, which can be found on the LoRa Alliance's website [3, 9]. The parameters define how long can one device transmit a signal within a time period (usually 1 h). This restriction is called the duty cycle (DC). The most common restriction is to transmit a maximum of 1% of the time per hour per device [10].
Typical LoRaWAN or LoRa@FIIT network, in general, consists of four main components [2, 9]:

- **End Nodes (EN)**: Sensors or devices communicating with gateways using LoRa technology. The nodes are connected to the gates in a star topology.
- **Gateways (GW)**: Gateways are controlled by a network server and send data to it from ENs. Gateways communicate with the server via an Internet Protocol (IP) network using Secure TCP for IoT (STIoT) protocol.
- **Network Server (NS)**: The network server is responsible for managing the individual devices of the entire LoRa wireless domain. Servers route data from end

devices to the application server and back. The network server specifies the SF and TP for the end devices. It sends downlink messages and maintains DC of each node.
- **Application Server (AS)**: Software running on a server displaying data from a specific application.

STIoT is used to transfer JSON configuration from GW to the NS using a traditional IP network.

3 Related Works

There is currently no such a tool for visualizing network data and managing nodes in LoRa@FIIT networks. In order to analyze traffic, the network administrator must know the current implementation of the database and the NS in order to be able to manually display the data via SQL queries [11]. We therefore decided to use and analyze existing solutions in similar domains for inspiration. The LoRa@FIIT protocol is comparable to the LoRaWAN protocol. Both LoRa@FIIT and LoRaWAN protocols operate on MAC layer of OSI model. LoRa@FIIT is designed to be more energy-wise than LoRaWAN. We will focus on the architecture of these existing solutions for data management and visualization in LoRaWAN networks (The Things Network, TTN [12] and ChirpStack [13]), in our work. However, we will also look at the device management in WiFi network technology (UniFi [14]), as it may potentially contain functionalities that have not been implemented yet in IoT networks. WiFi is also, like LoRa, a wireless network. However, they operate in different frequency bands and have different energy consumption requirements. Their common characteristics are operating in a free band and being able to communicate bilaterally. However, LoRa is only able to receive messages after sending an uplink message. Both LoRa@FIIT networks and UniFi solution use a single central point where data is processed; thus, the WiFi network is conceptually similar to the LoRa@FIIT network.

In a TTN network, ENs send messages using a LoRaWAN protocol. The messages are received by all gateways in the range. The Router, which is connected to one or more Brokers, is responsible for managing the Gateway. The Broker forwards the messages to a specific Router, which forwards them to the ENs. In addition, the Broker connects a specific device to a specific application. At the application side, the data is encrypted (resp. decrypted) using AES-128 by the Handler that is connected to the Broker. The entire message routing operation is provided by the Discovery server. The described TTN network architecture can be seen in Fig. 1 [15]. TTN in the application environment displays a list of ENs in the network, a list of GWs, plots them on the map, and offers basic statistics about devices (e.g., when it was last seen, the payload). The TTN application environment focuses on decoding the payload of devices. ChirpStack is conceptually very similar to TTN but focuses more on displaying network statistics, e.g., packet delivery in time.

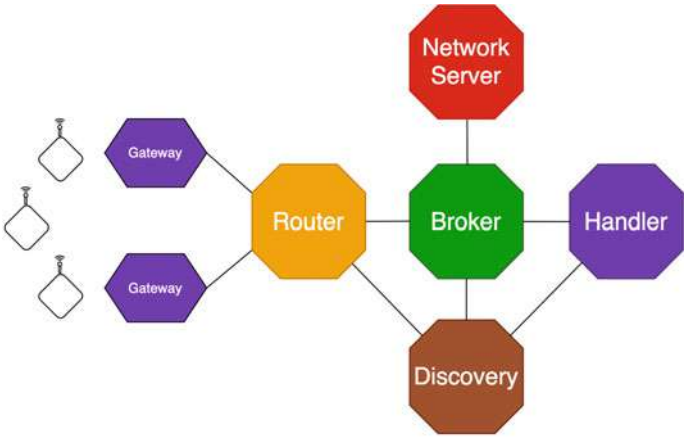


Fig. 1 TTN architecture [15] resembles recommended LoRaWAN architecture [16]

UniFi is one of the products of the US company Ubiquiti and focuses on the management of home or commercial WiFi networks. The graphical interface offers interesting functionalities that we could not find in the TTN or ChirpStack solutions. On the company website, they provide a functional demo that anyone can try [14]. In our work, we were mainly inspired by the following parts:

- **Dashboard**. Home screen with overall network performance and utilization summary.
- **Devices**. List of network devices connected to the network.
- **Clients**. List of end devices connected to the network.
- **Settings**. UniFi controller settings to manage network and end devices.

After a thorough analysis of the functionalities and the architectures of the existing solutions, we specified the most suitable components for a data visualization:

- **Line Chart**: A classic type of display of connected points on mutually perpendicular axes. For example, it displays the number of messages sent and received by the device every hour for a given time interval.
- **Bar Chart**: It displays the same information as line chart but in bar tables shape.
- **Heat Map**: It displays the geographically distributed signal quality. The map is exceedingly useful because it indicates areas with a strong and a weak signal.
- **Progress Bar**: A convenient method for visualizing what % of packets were received by the Network Server.

4 Design

While designing our solution, we focused on fulfilling the following goals:

- Provide information about the LoRa networks without the prior knowledge of the physical model of the database.
- Enrich the functionality of existing ChirpStack and TTN solutions.
- Provide the network administrator with a tool to help them set the parameters of the devices in the network, so that the network works as efficiently as possible in the given environment (rendering important network quality metrics using existing visualization methods).

The proposed system will simplify the work of the network administrator by:

- Displaying a clear view of the performance metrics (e.g., battery level or a packet delivery ratio) of the entire network and its components.
- Allowing the network administrator to configure GWs or ENs through a graphical interface and upload or download the configuration of a gateway.
- Displaying the list of ENs, GWs, and applications in a brief table with the possibility of filtering or sorting records and a link to the record detail. In a similar table, it displays the messages sorted by an application, a device, or a GW.
- Displaying the gate signal strength radius based solely on mathematical formulas.

To meet the design goals, we propose a system that will be available in the form of a graphical interface as a web application in the environment of common browsers. The graphical environment will support the user's login the first time the application is launched, in the classic way of combining a name and a password. The user environment must be so intuitive that anyone with basic knowledge of LoRa networks will be able to use it without the need for training.

The system consists of two parts (client and server). Both parts will communicate with each other using REST API and WebSocket technologies [17]. The server will be implemented in Node.js technology combined with the Express.js framework. We propose to implement the client part in the React.js framework.

In the next part of the design, we will describe the architecture of our solution, the proposed technologies, and present the implemented screens.

4.1 Architecture

The architecture of the solution consists of three main parts. LoRa NS, its web interface (also called the application server), and GW transmitting communication from ENs. The architecture can be seen in Fig. 3.

Only one of these parts is required to visualize the data, and it is database of NS. Any other LoRa NS [13, 29] can be used with our proposed solution as soon as it stores data in the proposed physical data model (illustrated in Fig. 2).

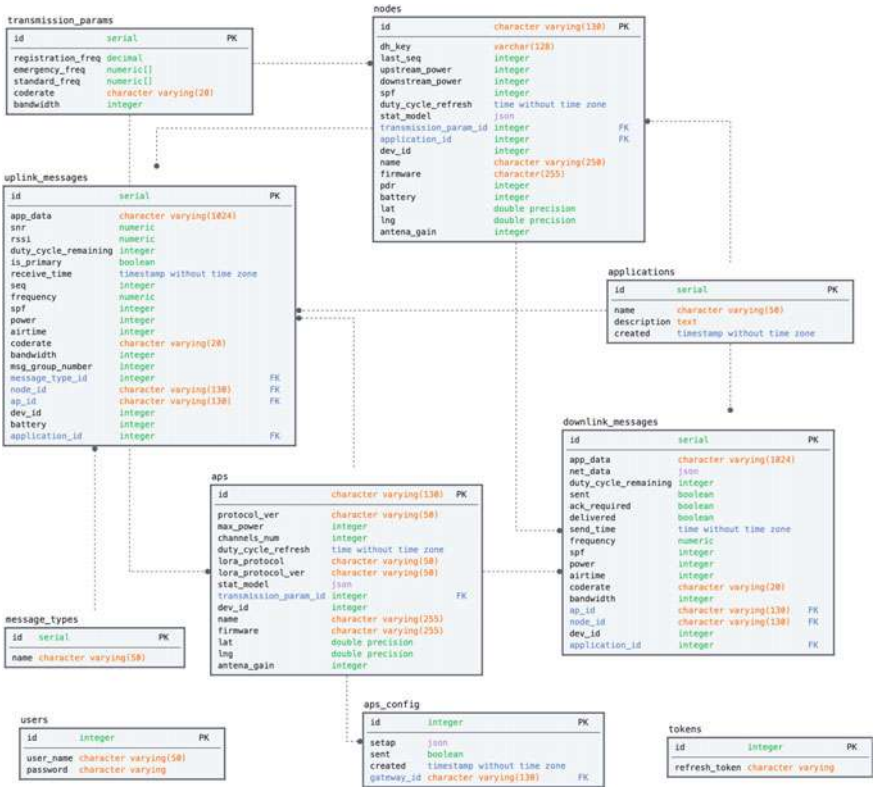


Fig. 2 Modified physical data model

Most of our work consists of the implementation of the LoRa web interface for NS. It exposes the graphical environment for the user and also communicates with the NS via HTTPS requests. The web interface reads data from the NS database. The web interface communicates directly with the NS only when the GW configuration changes. The configuration is changed by sending the message to a NS, which then sends it via the STIoT protocol to the GW. For the needs of two-way communication between the application and the network server, we propose to partially modify the implementation of the network server using the Redis publish/subscribe model [18]. To ensure two-way communication between the web server and browser, we propose to use WebSockets technology. Using Redis and WebSockets technologies allows us to view data in real time without the need of manual refresh of the page. The simulated GW generates packets, simulating the communication of devices on networks. The user accesses the environment of the application via the public address where the LoRa@FIIT web interface is installed (Fig. 3).

There are several technologies, but not all of them are suitable for our solution. It is important to distinguish which are adequate for the server and which for the client.

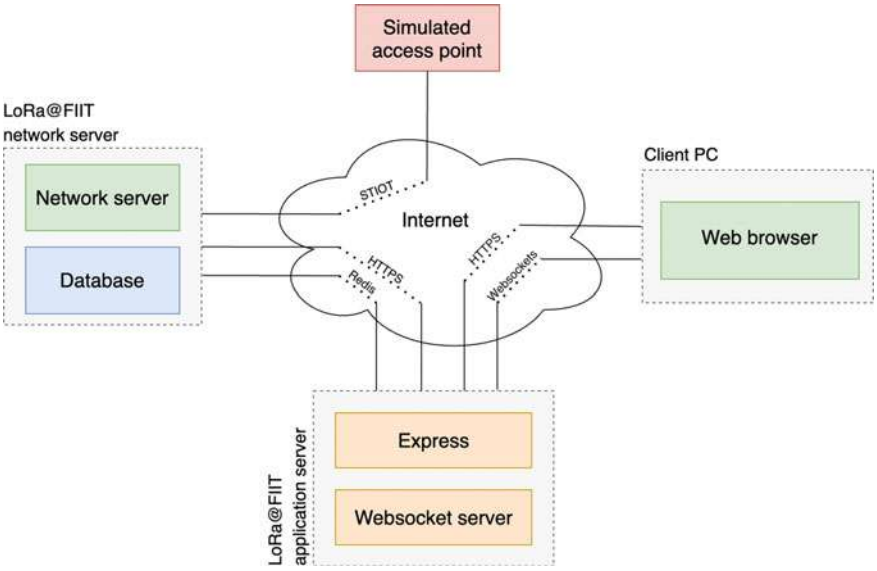


Fig. 3 Design solution architecture

When choosing the appropriate technologies, we took several things into account: fast response of the system, a single page feature, two-way communication, and a modern, minimalist and responsive design. After analyzing the technologies, we suggest using the following:

- **React**: It is JavaScript (JS) framework for building powerful user interfaces (UI) and provides a wide set of tools and libraries suitable for our solution [19].
- **Redux & React Redux:** The official React libraries for managing application state as a global repository for data, accessible across all application components [20].
- **React Router**: A collection of navigation components that make up the user's navigation apparatus in a user environment without the need to manually refresh the page [21].
- **Material-UI**: Front end library obeying to the design conventions introduced by Google [22]. We have chosen the library as the most practical choice because it is easy to implement and maintained. It is also implemented by large companies such as Netflix, Spotify, Amazon, and many others [23].
- **Formik**: A library designed to work with forms. Its task is to handle the input in the form from the user [24].
- **Node.js**: Runtime environment that executes JavaScript outside the web browsers and can handle a lot of requests due to a single-core event Loop mechanism [25].
- **Express**: A flexible framework for Node.js is providing a set of features to make working with an HTTP server easier. It is a clear choice for its minimality and practicality combined with a robust API [26].

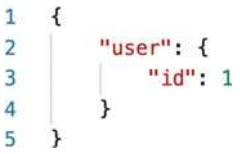


Fig. 4 Data structure in JWT

- **Socket.IO**: WebSocket technology enables event-driven two-way communication between server and client. The Socket.IO library makes it easy to work with WebSockets [27].
- **Redis**: It provides a publish/subscribe model that we propose to use to notify the application server of any change in the database state due to the packet generation by the simulated GW [28].

4.2 User Interface

In this section, we present our design of the LoRa@FIIT network server's web interface. We were inspired by the aforementioned preexisting technologies. To secure the application server, we use a combination of name and password, which the user fills in when the application is started for the first time. The user can change the credentials later in the application settings. The entered data will be stored in the database during registration.

Validation of data when registering, logging in, and adjusting login data will be done with JSON Web Token (JWT) approach. When registering or logging in to the application, the server returns a JWT, which contains the data in the structure as seen in Fig. 4.

After the user logs in, the JWT client is saved to local storage of the browser. The token structure contains an id, which is used to identify the user at each request to the server.

For user authentication, a JWT is sent with each HTTPS request to the server. Because a pre-agreed string is used when signing the JWT, only the network and application server can verify the token. If the JWT is successfully verified, then request is processed. The procedure for verifying user authentication when requesting a server is shown in the sequence diagram in Fig. 5.

When designing screens, we tried to preserve the conventions in interactive systems by following the ten principles of Jakob Nielsen [29] as well as the eight golden rules of Schneiderman [30] for the best possible user experience. We placed emphasis mainly on the fact that the user should always know the state of the application. It is very important because the application consists of many different screens and states. The user can return from any state of the application, in case he inadvertently got into a state in which he does not want to be. In addition to designing a desktop

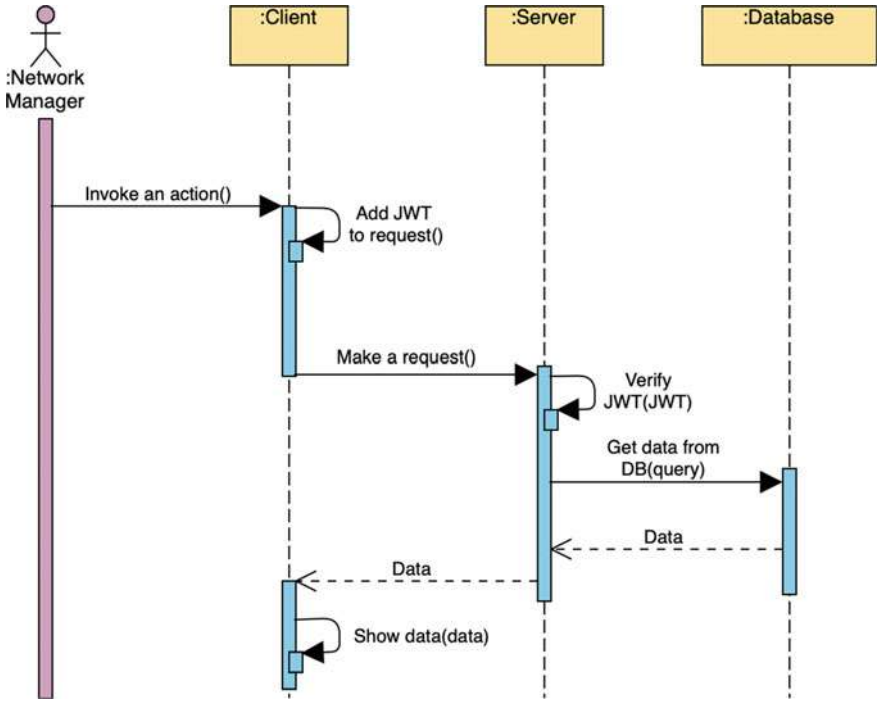


Fig. 5 Client authentication sequence diagram

application, we have also developed a mobile version of the UI which can be seen in Fig. 7. We paid attention to a consistent and intuitive design across all screens.
In Fig. 6, we can see the basic layout of the application. The design can be divided into three main parts:

1. **Header**: It displays the users' path (breadcrumb) to easily switch between application states. In the right corner, there is a button to refresh the content of the page together with the profile icon. The user can change the login details or log out of the application by clicking on the profile icon.
2. **Sidebar**: It serves as the main navigation in the application. The active link has a different background color. Click on the arrow to hide the panel to enlarge the content of the application.
3. **Application content**: The main area that is filled with content depending on the current state of the application (Fig. 7).

To update the configuration of nodes (both GWs and ENs), we propose a new STIoT message called SETAP which can be seen in the JSON structure [2] (Fig. 8): The STIoT configuration message is identified by the *message_name* attribute and the device identifier *ap_id*. After authentication, the web interface sends a message

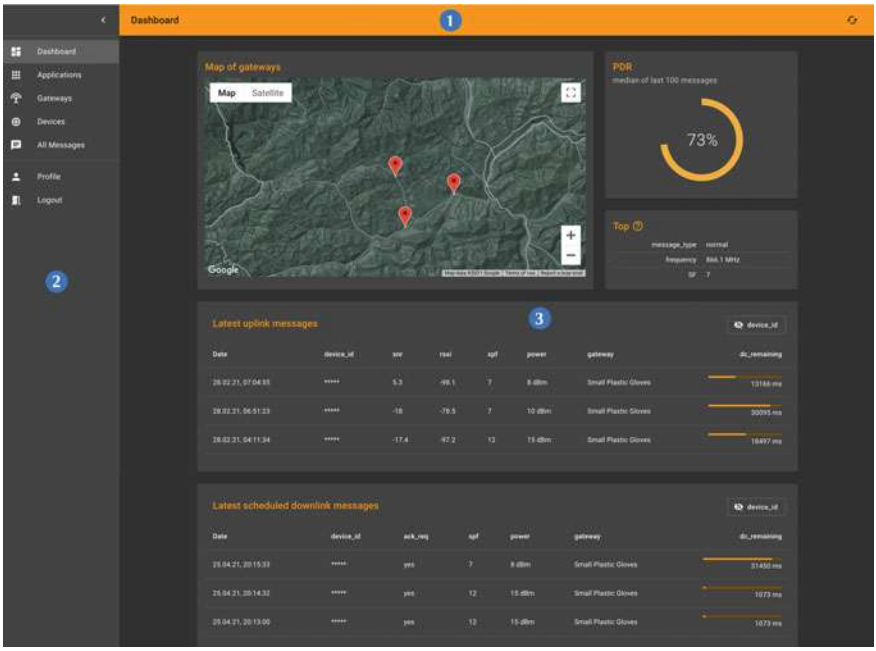


Fig. 6 Screen layout

to the network server's HTTPS endpoint. If no error occurs, the user is notified that the GW has been successfully configured.

The GW detail at Fig. 9 includes the signal radius component, which shows on the map the estimated signal coverage transmitted it. The calculation of the coverage range of the GW is expressed in the following paragraph [31]. 32.5 is a constant value used when distance is in km and frequency in Hz:

$$\text{GTX} = \text{TP} - \text{Cable Loss}_{\text{EN}} + \text{Antenna Gain}_{\text{EN}} \tag{1}$$

Free Space Path Loss (FSPL) calculation:

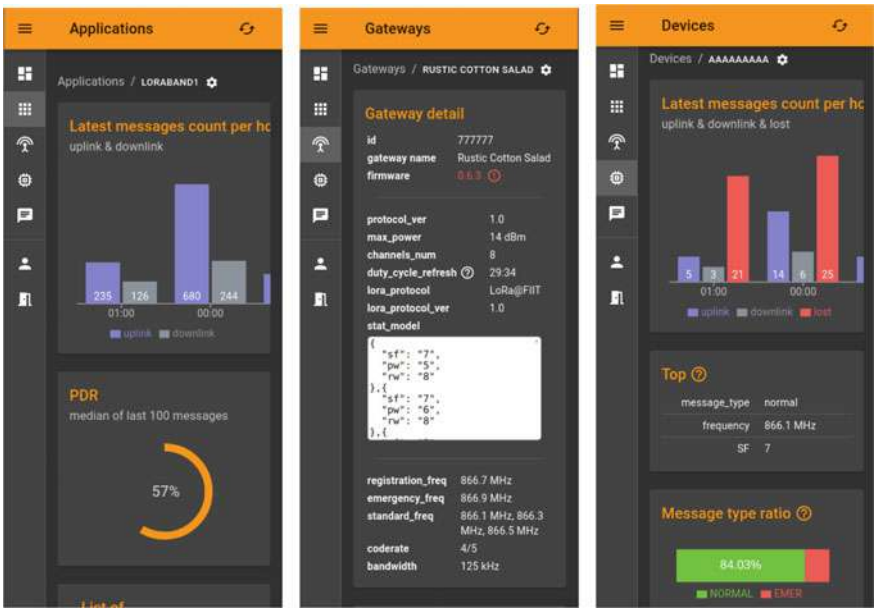
$$\text{FSPL} = 32.5 + 20 \times \log_{10} \text{Frequency} + 20 \times \log_{10} \text{Distance} \tag{2}$$

Gate transmission power calculation:

$$\text{RSSI} = \text{GTX} - \text{FSPL} - \text{Cable Loss}_{\text{GW}} + \text{Antenna Gain}_{\text{GW}} \tag{3}$$

It follows (for the sake of clarity, abbreviations are used in the formula):

$$\text{Distance} = 10^{\frac{\text{TP}+\text{CL}_{\text{EN}}+\text{AG}_{\text{EN}}-32.5-20\times\log_{10}\text{Frequency}-\text{CL}_{\text{GW}}+\text{AG}_{\text{GW}}-\text{RSSI}}{20}} \tag{4}$$



(a) Application detail mobilescreen

(b) Gateway detail mobile screen

(c) Device detail mobile screen

Fig. 7 Mobile screens

```
{
    "message_name": "SETAP",
    "ap_id": 9876,
    "message_body": [
        {
            "type": "NORMAL",
            "cr": "4/5",
            "freqs": [
                869600000,
                869800000
            ],
            "band": 12500,
            "power": 14,
            "sf": 0
        }
    ]
}
```

Fig. 8 SETAP message JSON format

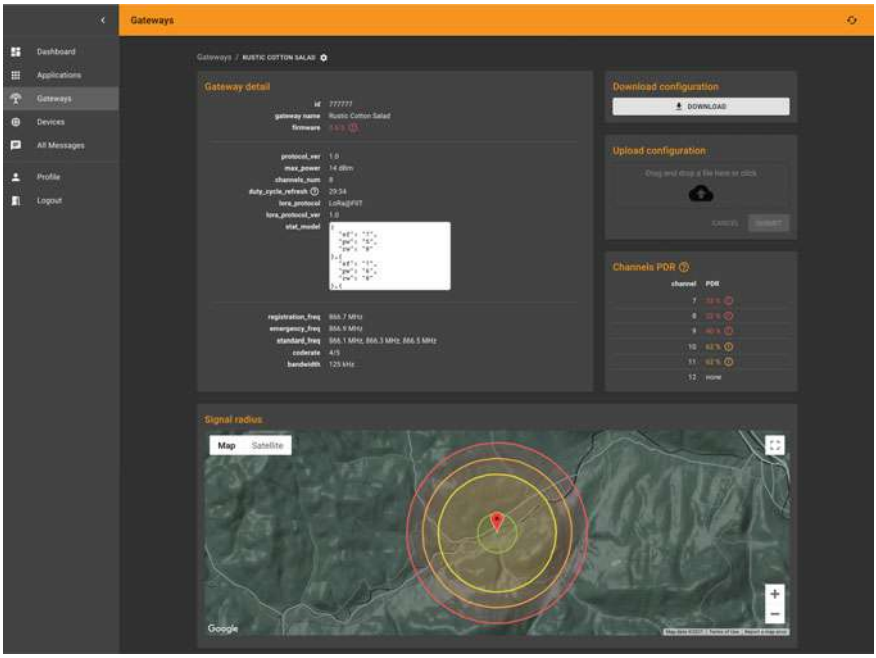


Fig. 9 Screen of gateway detail

The final distance is in kilometers. We omit Cable Loss (CL) in the calculation, because this value would be negligibly small for a very short (5 cm) low loss antenna cable. We also do not consider possible objects between the GW and the EN, which could shorten the signal range. The display of the gate signal radius is therefore only indicative and may not reflect reality.

Like a GW, EN settings can be updated using a web interface. Changing the settings in the web interface schedules a downlink configuration message (only SF and TP can be updated). After an uplink from a particular EN, the NS sends the message to the EN.

Throughout the development process, we propose continuous writing of test to minimize the errors in the system. We suggest using unit testing to test auxiliary functions, or simple parts of the system that work in isolation independently of the other parts. Implementing integration tests is more time-consuming but it also emphasizes dependency testing. UI tests are aimed at testing changes in the user environment with which the user interacts. We therefore suggest testing the correct response of UI components when the system state changes. The last type of testing that we propose to use is manual testing, which is the slowest to perform.

Our proposed solution was successfully deployed and is available online at lora.fiit.stuba.sk/demo. Using our tool, a network administrator can easily identify connected devices, message types and estimated number of collisions, overall PDR,

radius of a single GW, and display the latest uplink messages or scheduled downlink messages. For GWs to be fully configurable, changes to firmware have to be made [32]. ENs are configurable using current LoRa@FIIT library [11].

5 Conclusion

There is currently no tool for managing and displaying components in the LoRa@FIIT network. The network administrator must know the physical model of the database and be familiar with SQL to display the important metrics. There are currently solutions on the market that provide component visualization and device management for LoRaWAN networks. We analyzed the existing solutions, and based on them, we created our own components for data visualization in LoRa@FIIT networks. In addition to LoRaWAN solutions, we were also inspired by an application for managing and displaying WiFi networks. We have focused on displaying the performance metrics. Based on the analysis of existing data management solutions, we proposed our own solution. We plan to implement inserting obstacles to map in order to improve the signal propagation model to be more realistic. We also plan to give recommendations based on the collected results. We presented proposed design of the application's UI screens, briefly explained the method of user authentication, proposed the system architecture, defined a new type of STIoT message called SETAP for GW management, and proposed a way to verify the solution.

Acknowledgements This research was partially supported by the Slovak University of Technology in Bratislava and the Ministry of Education, Science, Research and Sport of the Slovak Republic, within the project ITMS 313021X329, co-funded by the European Regional Development Fund.

References

1. IoT Analytics. State of the IoT 2020: 12 Billion IoT Connections. https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/
2. Perešíni O, Krajčovič T (2017) More efficient IoT communication through LoRa network with LoRa@FIIT and STIOT protocols. In: 2017 IEEE 11th international conference on Application of Information and Communication Technologies (AICT), pp 1–6. https://doi.org/10.1109/ICAICT.2017.8686837
3. LoRa Alliance. https://lora-alliance.org/
4. Phung K-H et al (2018) Analysis and assessment of LoRaWAN. In: 2018 2nd international conference on recent advances in Signal Processing, Telecommunications Computing (SigTelCom), pp 241–246. https://doi.org/10.1109/SIGTELCOM.2018.8325799
5. Yousuf AM, Rochester EM, Ghaderi M (2018) A low-cost LoRaWAN testbed for IoT: implementation and measurements. In: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), pp 361–366. https://doi.org/10.1109/WF-IoT.2018.8355180
6. Abdelfadeel KQ, Cionca V, Pesch D (2018) Fair adaptive data rate allocation and power control in LoRaWAN. In: 2018 IEEE 19th international symposium on "A World of Wire-

less, Mobile and Multi- media Networks" (WoWMoM), pp 14–15. https://doi.org/10.1109/WoWMoM.2018.8449737
7. LoRa Alliance R. Specification v1.0.3. https://lora-alliance.org/resource-hub/lorawanr-specification-v103
8. LoRa Alliance R. LoRaWAN R for Developers. https://lora-alliance.org/lorawan-for-developers
9. The Things Industries. LoRaWAN Architecture. https://www.thethingsnetwork.org/docs/lorawan/architecture.html
10. El Rachkidy N, Guitton A, Kaneko M (2019) Collision resolution protocol for delay and energy efficient LoRa networks. IEEE Trans Green Commun Netw 3(20:535–551. https://doi.org/10.1109/TGCN.2019.2908409
11. Valach A, Stefunko S. LoRa@FIIT library. https://github.com/alexandervalach/lorafiit-library
12. The Things Industries. The Things Network. https://thethingsnetwork.org
13. Orne Brocaar. ChirpStack open-source LoRaWAN R Network Server. https://www.chirpstack.io/
14. Ubiquiti. UniFi-Build networks more powerful than ever before. https://unifi-network.ui.com
15. The Things Industries. Network architecture. https://www.thethingsnetwork.org/docs/network/architecture.html
16. LoRa Alliance. LoRaWAN R 1.0.4 specification package. https://lora-alliance.org/resource
17. IETF. RFC 6455—The WebSocket protocol. https://datatracker.ietf.org/doc/rfc6455/?include
18. Redis Labs. Pub/Sub-Redis. https://redis.io/topics/pubsub
19. Facebook Inc. React—a JavaScript library for building user interfaces. https://reactjs.org
20. Redux. Redux—a predictable state container for JavaScript apps. https://redux.js.org
21. React Training. React router: declarative routing for react. https://reacttraining.com/react-router
22. Google. Material design. https://material.io
23. Google. Material-UI: a popular react UI framework. https://material-ui.com
24. Formium Inc. Formik. https://formik.org
25. OpenJS Foundation. The Node.Js Event Loop. https://nodejs.dev/
26. OpenJS Foundation. Express—Node.js web application framework. https://expressjs.com
27. Damien Arrachequesne. Socket.IO. https://socket.io/index.html
28. Redis Labs. Redis. https://redis.io
29. Nielsen J (2005) Ten usability heuristics
30. Shneiderman B (2004) Designing for fun: how can we design user interfaces to be more fun? Interactions 11(50:48–50. issn: 1072-5520. https://doi.org/10.1145/1015530.1015552
31. Linka H et al (2018) Path loss models for low-power wide-area networks: experimental results using LoRa, May 2018
32. Valach A, Skuta M. LoRa access point. https://github.com/alexandervalach/lorafiit-access-point

Chapter 30
Deep Elephant Herding Optimization-Based Attack Detection for Securing Virtualized Infrastructures of Cloud

Bhavana Gupta and Nishchol Mishra

1 Introduction

Virtualization technology of cloud has been implemented as modern-day cloud infrastructures, where a growing amount of users make an intelligent move to different cloud benefits [1, 2]. This ensures that cloud services are available to everyone. Security, on the other hand, remains a major worry for cloud customers' and services' important needs [3]. As a result, security becomes the most irrefutable aspect in cloud architecture [4, 5] and should be prioritized to ensure the integrity and confidentiality of user data [6, 7]. Furthermore, the security situation might be called into question on various levels. Some of the issues in the virtualization system are as follows: (i) the possibility of multi-tenancy necessitates access to virtualized resources and complete network access to cloud resources [8, 9], (ii) the increasing data protection factor places more conditions on Cloud Service Providers (CSPs) to ensure the resources' integrity and confidentiality, and (iii) the cloud SLA defines the resource availability of virtualized infrastructure [10, 11]. However, since the linked goods and technologies are not fully developed, the deployment and development of these virtualization approaches are still in the early stages [12, 13]. As a result, determining "what type of security concerns may develop" is difficult. In general, three elements have a substantial influence on virtualization's attack surface. Containerization, cross-layer invocation and hypervisor isolation are the three. As a result, we need a comprehensive approach to addressing these security concerns [14–16].

Notably, from a vertical standpoint, programmes in VMs are often integrated with additional components and layers, spanning from the hypervisor to the guest OS [17, 18]. As a result, the misconfiguration of the VMs may ultimately enable the attacker

B. Gupta (✉) · N. Mishra
Department of Computer Science and Engineering, School of Information Technology, UIT-RGPV, Bhopal, India
e-mail: bhavanagupta2506@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_31

to infiltrate the apps. Furthermore, each layer is made up of multiple heterogeneous components from diverse suppliers' horizontal perspectives. However, in software or hardware, preserving the trust connection amongst the pieces is harder [19]. As a result, in order to evaluate the risk and security of this situation, a probabilistic framework capable of dealing with inherent uncertainties must be used [20, 21]. In contrast, research on these models has not yet been conducted in a systematic manner in order to cope with security vulnerabilities in a virtualized environment. The following are the primary contributions of the suggested model:

- Presents a unique weighted probability-based holoentropy feature.
- Proposes an improved DBN model for detecting attacks, with weights ideally set using an EHO approach.

In this paper, Sect. 2 of this research offers a literature survey on traditional threat detection techniques in cloud computing. Section 3 describes the suggested threat detection mechanism in cloud computing's virtualized infrastructure. Section 3 depicts the feature set, which includes an enhanced holoentropy-based feature. The deep belief network-based attack detection is shown in Sect. 4. Section 5 depicts the outcomes and debates of suggested projects. Section 6 brings the paper to a close.

2 Literature Review

A. *Related works*

In 2020, Asvija et al. [22] created a new Bayesian attack graph model in 2020 to analyse security concerns in cloud computing infrastructure, which is often used for designing cloud services. The benefit of BAGs is that they may be used to emulate the uncertainties that are inherent in security breaches. They extend the reference conditional probability tables for the BAG nodes by using known attacks on virtualized systems from the Common Vulnerabilities and Exposures (CVE) database. To showcase the results of our model, we use Bayesian probabilistic inference and show the conclusions that system architects may use to assess the risk of infrastructures. They have also offered a deterministic technique for attack graphs that is compatible with a probabilistic model, as well as extracted the security metrics used for the simulated BAG, which can be compared against other designs.

In 2020, Maxime et al. [23] describe in-depth at framework-based virtualization architectures and their effects on cloud security. Virtualization of the underlying architecture is achieved in such a manner as to perform software inspection on selected instructions. The VMM utilizes the trapping mechanism to oversee and manage the operating system. Another thing to keep in mind is that the privileged instructions (the technology) were also part of the sensitive instructions (the instructions used for accessing sensitive data). In addition, intercepting instructions in unprivileged mode were enabled, and sensitive instructions were handled as needed. At long last, the results of modified method implementation

have shown to provide a higher level of security in this kind of architecture. In 2019, Patil et al. [24] have proposed a secure method to detect and track breaches in cloud computing virtual networks. An HLDNS, which is implemented on each cloud computing server, is suggested as a possible solution. To prevent intrusions, it takes network traffic from/to the virtualized environment, internal network and external network on all nodes and correlates that traffic to the VMs to determine if any traffic occurred. Researchers generalized a BBA with two new fitness factors, including cloud network traffic functionality, for easy extraction of viable cloud network traffic functionality. The suggested system's final performance proved to show improved cloud security network analysis.

In 2020, Alaluna et al. [25] have developed the multi-cloud virtual network that has been established via embedding. There are two ways to handle the primary distribution problem in network virtualization. First, virtual networks encoded to find safe routing. The MILP was designed to help solve this problem. We obtained findings that help us learn about the costs and trade-offs of implementing security requirements in network virtualization. Finally, the suggested method's performance showed that virtual networks were being put to efficient use by businesses for security. In 2019, Amit et al. [26] addressed the issue of virtual machine security in cloud computing by proposing the use of virtual machine hosts in cloud computing. Previously Co-located Users First, a novel VM placement technique, was suggested for minimal likelihood of harmful co-location in VMs. A cloud service provider using a safe VM placement method helps to mitigate some co-location-related risks. According to the suggested technique, the results of the cloud's co-location resistance tests have demonstrated better results than conventional methods.

In 2018, Win et al. [27], cloud computing is vulnerable to attack, and our big data-based security analytics approach has been used to protect it. The HDFS keeps track of the network information, as well as device programme information, which is received frequently from the virtual machines' guest operating systems. Once the threat is detected, the extraction of security characteristics is obtained by locating possible attack routes via the MapReduce parser and event correlation in the graph. The experimental findings confirmed that the suggested approach is more efficient than existing models. In 2018, Joseph et al. [28] have shown malware attacks in cloud storage by using VM snapshots on virtual machines in order to make the network self-healing. The VM's self-healing capability on the local cloud network allows VMs to benefit from snapshots without impacting any VM in the backup server. Additionally, a machine learning technique could tell the difference between the projected images that were damaged and those that were not. Finally, the chosen method's performances indicate that it is more effective at identifying and detecting different types of malware.

In 2019, Benkhelifa et al. [29] suggest using cloud service providers to safeguard and defend virtual environments. Non-deterministic device testing technique had previously made it clear that pricing models of the device were

complex. This study has the aim of supporting both corporations and universities involved in the development of cloud-based virtualized application ecosystems. A service that offers a software testing platform for virtual network testing is under development. As a result, the suggested method's performance proved to be superior to previous approaches.

B. *Review*

Table 1 illustrates the reviews on cloud computing virtual infrastructures. Early on, Bayesian inference was used in [22], which enhanced scalability, efficiency and security. However, to get more accurate risk assessment, one must analyse post-incident evidence of past security breaches. Virtualization techniques were used in the book, [23], which offers reduced security threats, the ability to use new possibilities in safeguarding the cloud and lower complexity, but it needs attention to be given to the integration of IoT resources into the cloud.

Additionally, HLDNS was implemented in [24] which employs identification and development of new attack signatures in the distributed attack cloud and an improved network security. Despite this, it requires additional development in terms of network assaults, in particular while operating under various system levels. It has also been used in [25] to solve a more efficient, cost-revenue-efficient and expressive trade-off between user demands. On the other hand, it should be formulated in the style of chebychev style with the aid of reference points for addressing problems like these. The VM placement algorithm was successfully used in [26] that were shown to boost cloud locations' resistance to co-location attacks, reduce the risk of malicious VM co-location and improve resource efficiency and security. However, the technology requires continued investigation and refinement to further strengthen and secure live migration. The addition to this was that BDSA (Best Digital Security Approach) was used, which provides higher visitor numbers and constant performance overhead and is successful in detecting assaults. However, it will have to spend a lot of time training the logistic regression classifiers, who need on-the-fly retraining. Using the SVM method in [28, 29] produced faster, more accurate detection and had higher performance. Despite that, it should place greater emphasis on the SVM method, which results in zero false negatives. The problems which apply to virtualized infrastructures in cloud computing have to be taken into consideration for successful work in the current paper.

3 Proposed Framework for Attack Detection Framework in Cloud computing's Virtualized Infrastructure

In this article, we propose a novel method to cloud computing security analytics that detects threats in virtualized infrastructure. Attack may be detected by examining the features of the network's structure. For these purposes, the suggested model includes two distinct phases: the first being feature extraction, and the second being classification. The goal is to use Python's multi-process architecture, with Python

Table 1 Detailed overview of traditional virtualized infrastructures in cloud computing models, including their features and challenges

Author (citation)	Implemented scheme	Features	Challenges
Asvija et al. [22]	Bayesian inference technique	• Enhanced scalability • Enhanced efficiency • Enhanced security	• Security breaches are best evaluated using a posteriori evidence data, which is best obtained by increasing the use of a posteriori evidence data
Maxime et al. [23]	virtualization methods	• Abridged security concerns • New potential for cloud service security • Reduced complexity	• Integrating IoT resources based on the dispersion and heterogeneity and providing services at the edge are all elements that should be considered
Patil et al. [24]	HLDNS method	• Recognize the scattered assaults • Create a new attack signature cloud • Improved network security	• The protocol should be extended to identify attacks using the packet data portion • It is essential to verify the fitness functions with BBA that have been implemented to ensure that the system level assaults have been detected
Alaluna et al. [25]	SecVNE model	• Better efficiency solution • Better cost-revenue trade-offs • Refining request expressiveness	• To meet these requirements, we must use reference points to access and solve problems that have the characteristics of Chebyshev-like functions
Amit et al. [26]	placement algorithm for VM	• Cloud co-location resistance has been increased • Reduced likelihood of harmful VM co-location • Improvements in resource efficiency and security	• An algorithm that focuses on designing and analysing safe live migration should be developed

(continued)

Table 1 (continued)

Author (citation)	Implemented scheme	Features	Challenges
Win et al. [27]	Approach BDSA	• Increased visitor count • Reliable performance overhead • Effective in identifying attacks Consistent performance overhead	• Concentrate on retraining the logistic regression classifiers on-the-fly while dealing with an unanticipated growth of the classifiers
Joseph et al. [28]	SVM algorithm	• Extra efficient • High-level detection effectiveness • Enhanced performance	• SVM must make up for the 0 false negatives that it produces
Benkhelifa et al. [29]	Servitization Process	• Enhanced test coverage • Low complexity • Improved reliability, accuracy and stability	• A key point of growth will be an emphasis on a legal framework

doing several tasks, to further segregate the amount of data to analyse depending on velocity, volume and veracity and to reduce the time taken for processing. The Feature extration takes place for each individual dataset. Here, holoentropy-based features are extracted. This procedure was performed on the aggregated features from the multi-process system to determine whether there was an attack in the cloud. In order to increase the model's ability to anticipate an attack, the DBN model's weights are adjusted using EHO method, which is shown in Fig 1.

3.1 Feature Set: Improved Hole Entropy-Based Feature

Improved hole entropy-type features are extracted from the input data.

Improved hole entropy feature

In this paper, a total of four holoentropy features are extracted.

Conventional Holoentropy: "The holoentropy $\mathrm{HE}_X(Y)$ is defined as the sum of entropy, and the total correlation of the random vector is indicated as the sum of the entropies of all attributes" [30].

$$\mathrm{HE}_X(Y) = E_X(Y) + C_X(Y) = \sum_{i=1}^{m} \mathrm{HE}_X(y_i) \tag{1}$$

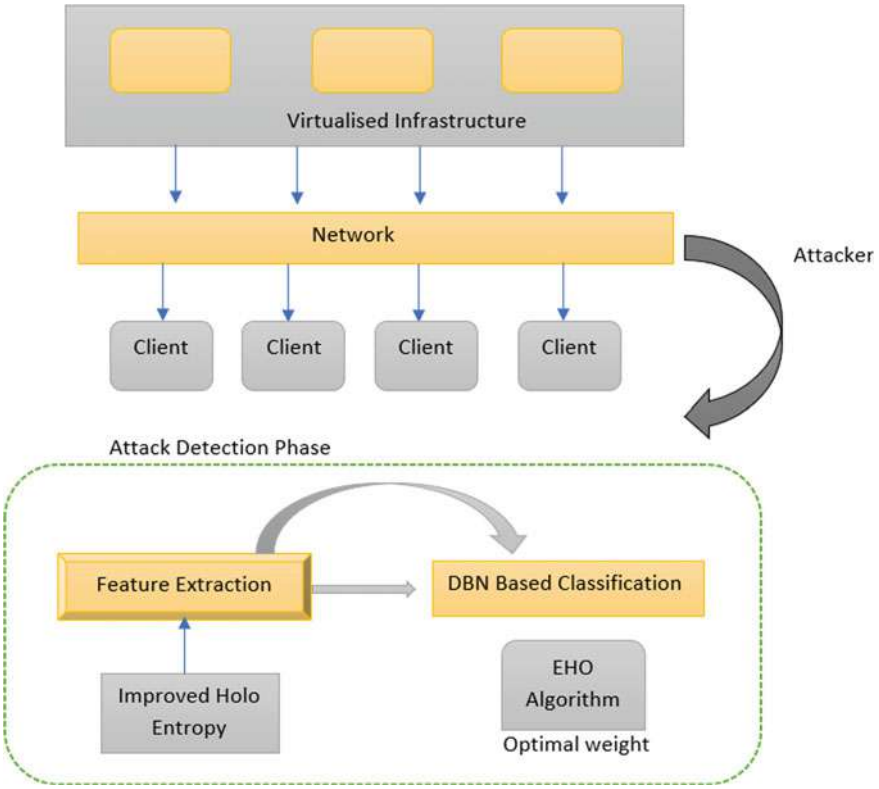


Fig. 1 The proposed architecture of attack detection framework in cloud virtualized infrastructure

In Eq. (1), $E_X(Y)$ indicates the entropy, $C_X(Y)$ implies the correlation, X denotes the dataset, and Y specifies the random vector.

Weighted holoentropy: The weighted hole entropy [30, 31] is defined as the sum of each attribute's weighted entropy. Equation (2) expresses the weighted holoentropy.

$$\mathrm{WHE}_X(Y) = \sum_{i=1}^{m} w_X(y_i)\mathrm{HE}_X(y_i) \tag{2}$$

The reverse sigmoid function is used as the weighting factor $w_X(y_i)$ in Eq. (3).

$$w_X(y_i) = 2\left(1 - \frac{1}{1 + \exp(-\mathrm{HE}_X(y_i))}\right) \tag{3}$$

The reverse sigmoid is a maximizing function in Eq. (3) that spans from (0, 2).

Weighted Probability-based Proposed Holoentropy: The holoentropy is calculated as follows.

$$\mathrm{HL}_E(X_i) = w \times E(X_i) \tag{4}$$

$$w = 2 \times \left(1 - \frac{1}{1 + \exp(-E(X_i))}\right) \tag{5}$$

$$E(X_i) = \sum_{t=1}^{\mu(X_i)} p_t \log p_t \tag{6}$$

Equation (6) signifies the unique characteristic in the feature attribute and represents the probability of feature attributes. Similarly, Eq. (7) gives the suggested holoentropy for both feature and label characteristics (7).

$$\mathrm{HL}_E(X_i, y_i) = w \times E(X_i, y_i) \tag{7}$$

$$w = 2 \times \left(1 - \frac{1}{1 + e^{-E(X_i, y_i)}}\right) \tag{8}$$

$$E(X_i, y_i) = \sum_{t=1}^{\mu(X_i)} p_t \tag{9}$$

Differential holoentropy: Consider an object x_0 of X, the difference of weighted holoentropy $h_X(x_0)$ between X and the $X \big/ \{x_0\}$ is defined as the differential holoentropy [30] of the object x_0 as per Eq. (10).

$$h_X(x_0) = \mathrm{WHE}_X(Y) - \mathrm{WHE}_{X/\{x_0\}}(Y) \tag{10}$$

$$h_X(x_0) = \sum_{i=1}^{m} \left[w_X(y_i)\mathrm{HE}_X(y_i) - w_{X/\{x_0\}}(y_i)\mathrm{HE}_{X/\{x_0\}}(y_i) \right] \tag{11}$$

In Eq. (11), $w_X(y_i)$ is denoted as a reverse sigmoid function of entropy $\mathrm{HE}_X(y_i)$, and the difference between $w_X(y_i)$ and $w_{X/\{x_0\}}(y_i)$ is lower than the entropy $\mathrm{HE}_X(y_i)$. The differential holoentropy is shortened in Eq. (12).

$$\hat{h}_X(x_0) = \sum_{i=1}^{m} w_X(y_i)\left[\mathrm{HE}_X(y_i) - \mathrm{HE}_{X/\{x_0\}}(y_i)\right] \tag{12}$$

4 Deep Belief Network-Based Attack Detection Using Optimization

4.1 DBN-Based Classification

Once all the relevant features have been identified, the DBN is used to distinguish amongst invaders in the network. Using the DBN [32] structure, which is made up of visible neurons, hidden neurons and many output layers, will provide an effective system for presenting data. This means that hidden and input neurons have a strong, intimate connection; however, visible neurons are disconnected, and association rules do not exist in hidden neurons [33]. A symmetric and exclusive connection is appeared amongst the visible and hidden neurons. The output of neurons in a Boltzmann network is probabilistic, in which the output $\overline{O}$ is based on the probability function $\overline{Q}_p(\delta)$ in Eq. (14). The DBN model is defined in Eq. (16), where q denotes the pseudo-temperature.

$$\overline{Q}_p(\delta) = \frac{1}{1 + e^{\frac{-\delta}{q}}} \tag{13}$$

$$\overline{O} = \begin{Bmatrix} 1 \text{ with } 1 - \overline{Q}_p(\delta) \\ 0 \quad \text{with } \overline{Q}_p(\delta) \end{Bmatrix} \tag{14}$$

$$\lim_{q \to 0^+} \overline{Q}_p(\delta) = \lim_{q \to 0^+} \frac{1}{1 + e^{\frac{-\delta}{q}}} = \begin{cases} 0 \text{ for } \delta < 0 \\ \frac{1}{2} \text{ for } \delta = 0 \\ 1 \text{ for } \delta > 0 \end{cases} \tag{15}$$

The feature processing operation in the DBN framework is carried out by a collection of RBM layers, while the classification procedure is carried out by MLP. The mathematical design depicts Boltzmann machine energy in the form of neuron or binary state in Eqs. (16) and (17). In this case, Mc,l denotes the weights of the neurons, which are ideally adjusted by a new DEHO and θ_a specify the biases.

$$N(b) = -\sum_{c<l} b_c M_{c,l} - \sum_{c} \gamma_c b_c \tag{16}$$

$$\Delta N(b_c) = \sum_{l} b_c M_{c,l} + \gamma_c \tag{17}$$

Equations (18–20) explain the growth of energy in terms of joint composition in hidden and visible neurons (u, v), respectively, and v_c and u_c depict the binary state of hidden l and visible c units. And $M_{c,l}$ represents the weight between them, as well as the g_c and L_l biases.

$$N(u, v) = -\sum_{(c,l)} M_{c,l} u_c v_l - \sum_c g_c u_c - \sum_l L_l v_c \tag{18}$$

$$\Delta N(u_c, \vec{v}) = \sum_l M_{c,l} v_l + g_c \tag{19}$$

$$\Delta N(\vec{u}, v_c) = \sum_l M_{c,l} u_c + L_l \tag{20}$$

The weight allocation is achieved using RBM training, and the dispersed probabilities are represented as in Eq. (21). In the RBM model, the probability distribution for the hidden and visible vectors pair $\left(\overline{u}, \overline{h}\right)$ is given in Eq. (23). Equation (24) specifies the partition function.

$$\hat{M}_{\left(\hat{C}\right)} = \max_{\hat{M}} \prod_{\vec{u} \in F} a(\vec{u}) \tag{21}$$

$$a\left(\vec{u}, \overrightarrow{h}\right) = \frac{1}{S} e^{-N(\vec{u}, \vec{v})} \tag{22}$$

$$S = \sum_{\vec{u}, \vec{v}} e^{-N(\vec{u}, \vec{v})} \tag{23}$$

The DBN model makes use of the CD learning model, and the processes involved in the CD model are as follows.

Step 1: For training, choose Samples $x_.$

Step 2: Compute the probability of hidden neurons a_v by differentiating the product of visible vector u and $\hat{M}$ weight matrix according to $a_v = \chi\left(u.\hat{M}\right)$ as per Eq. (24), which indicates the activation function.

$$\hat{M}_{\left(\hat{C}\right)} = \max_{\hat{M}} \prod_{\vec{u} \in F} a(\vec{u}) \tag{24}$$

Step 3: The hidden states v derived from c_v probabilities are investigated.

Step 4: Determine the exterior vectors product u and a_v which is evaluated as a positive gradient $\phi^+ = u.a_v^q$.

Step 5: Eq. (25) examines the transformation of visible states u' into v concealed states. Furthermore, it is critical to investigate any concealed states v' that may have resulted from the u' refurbishmentsate.

$$a(\vec{u}_l \to 1|\vec{v}) = \chi\left(g_c + \sum_c u_l M_{c,l}\right) \tag{25}$$

Step 6: Determine the outer product u and v using its negative gradient $\phi^{-} = u'.v'^{q}$.

Step 7: The updated weight is defined by Eq. (26). Here, κ refers to the learning rate.

$$\Delta \hat{M} = \kappa\left(\varphi^{+} - \varphi^{-}\right) \tag{26}$$

Step 8: The weight update is defined in Eq. (27) with new values.

$$M'_{c,l} = \Delta M_{c,l} + M_{c,l} \tag{27}$$

Let us assume training patterns $\left(G^{\hat{C}}, B^{\hat{C}}\right)$ in the MLP model learning process, where $1 \leq \hat{C} \leq \overline{P}\,\hat{C}$ denotes the number of trainings patterns and $G^{\hat{C}}$ and $B^{\hat{C}}$ denote the actual and anticipated outputs, respectively. The error evaluation is defined by Eq. (28).

$$e_l^{\hat{C}} = G^{\hat{C}} - B^{\hat{C}} \tag{28}$$

As a result, Eq. (28) depicts the squared error of $\hat{H}$ pattern success to MSE in Eqs. (29) and (30), where $\tilde{P}$ denotes the number of training patterns.

$$\mathrm{SE}_{\hat{C}}^{\mathrm{mean}} = \frac{1}{\tilde{J}_v}\sum_{l=1}^{\tilde{J}_v}\left(e_l^{\hat{C}}\right)^2 = \frac{1}{\tilde{J}_v}\sum_{l=1}^{\tilde{J}_v}\left(G^{\hat{C}} - B^{\hat{C}}\right)^2 \tag{29}$$

$$\mathrm{SE}_{\mathrm{avg}} = \frac{1}{\tilde{P}}\mathrm{SE}_{\hat{C}}^{\mathrm{mean}} \tag{30}$$

4.2 *Elephant Herding Optimization Algorithm for Optimal Weight Selection*

4.2.1 Solution Encoding and Fitness *Assessment*

DBN's weight is properly calibrated because fine-tuning of learnable weights increases DBN's efficiency in detecting the existence of assaults in the cloud. The input solution to the EHO [34] method is shown in Fig. 2, where $M_1\ M_2,\ldots.M_N$ denotes the DBN weights, and N denotes the total weight counts. Eq. expresses the

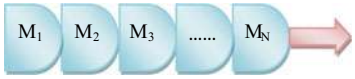


Fig. 2 Solution encoding

fitness assessment (objective) of the proposed detection framework (31). Here, $e_l^{\hat{C}}$ indicates the MSE as described in Eq. (28).

$$\mathrm{OF} = \mathrm{Min}\left(e_l^{\hat{C}}\right) \tag{31}$$

4.3 Elephant Herding Optimization model

This study will use the EHO algorithm [34, 35]. It was decided to use the Elephant Herding algorithms to resolve a variety of search issues [36, 37]. The flowchart depicts this operating concept (4).

Following are the three rules of the EHO algorithm that have been broken down into a more simple manner. Elephants may be joined together at random to form new clans. Each family uses an elephant with a fitness rating depending on their relative capabilities. Fitness is calculated using the Eq. (31).

- Elephants belong to many clans, and in each clan, the female and male elephants have an established number.
- In certain cases, males go on their own, preferring to be alone.
- Every clan has an elephant matriarch who functions as the clan leader.

Clan updating operator: Clan updates are done one at a time. The next place in clan operator updates is affected by the matriarch j and each elephant in the clan j. The elephant z in clan j is updated in Eq. (32).

$$A_{n,j,z}^{\mathrm{EHO}} = A_{j,z}^{\mathrm{EHO}} + \lambda.r.\left(A_{\mathrm{best},j}^{\mathrm{EHO}} - A_{j,z}^{\mathrm{EHO}}\right) \tag{32}$$

In Eq. (32), $A_{n,j,z}^{\mathrm{EHO}}$ is the most recently updated clan and $A_{j,z}^{\mathrm{EHO}}$ represents the elephant's prior place z in clan j, $\lambda \in [0, 1][0, 1]$ is the scaling factor, and it specifies the effect of matriarch on j on $A_{j,z}^{\mathrm{EHO}}$. $A_{\mathrm{best},j}^{\mathrm{EHO}}$ represents matriarch j with the fittest elephant individuals of clan j, $r \in [0, 1]$. In each clan, the fittest elephant is not calculated using Eq. (32), (i.e.) $A_{j,z}^{\mathrm{EHO}} = A_{\mathrm{best},z}^{\mathrm{EHO}}$. Rather, for the fittest elephant, the update process is performed as per Eq. (33).

$$A_{n,j,z}^{\mathrm{EHO}} = \eta \times A_{\mathrm{cen},j}^{\mathrm{EHO}} \tag{33}$$

In Eq. (33), $\eta[0, 1]$ is the centre of clan j. The new individual $A_{n,j,z}^{\mathrm{EHO}}$ is expressed in Eq. (16) from the information obtained by all elephants in clan j. $A_{\mathrm{cen},j,D}^{\mathrm{EHO}}$ represents the centre of clan j for Dth dimension, which is given in Eq. (34).

$$A_{\mathrm{cen},j,D}^{\mathrm{EHO}} = \frac{1}{N_j} \times \sum_{z=1}^{N_j} A_{j,z,D}^{\mathrm{EHO}} \tag{34}$$

In Eq. (34), $1 \leq D \leq d$ denotes the Dth dimension and d indicates the total dimensions. N_j specifies the number of elephants in clan j. $A^{\text{EHO}}_{j,z,D}$ refers to the Dth elephant individual $A^{\text{EHO}}_{j,z}$.

Separating operator: Male elephants are allowed to fend for themselves as adults and do not socialize with other elephants of their kind. In this scenario, the separating operator is provided from the separating method to avoid any optimization issues. Consider the elephant individuals with the lowest fitness at each generation for calculating the separation operator, as stated in Eq. (35).

$$A^{\text{EHO}}_{\text{worst},j} = A^{\text{EHO}}_{\text{min}} + r.\left(A^{\text{EHO}}_{\text{max}} - A^{\text{EHO}}_{\text{min}} + 1\right) \tag{35}$$

Here, $A^{\text{EHO}}_{\text{min}}$ and $A^{\text{EHO}}_{\text{max}}$ are the lowest and maximum boundaries in single elephant positions. $A^{\text{EHO}}_{\text{worst},j}$ identifies the clan's weakest elephant members, and j and r relate to a random number $\in [0, 1]$. The complete proposed approach is summarized in Fig. 3.

5 Results and Discussion

5.1 Simulation Procedure

The attack detection model DEHO (DBN + EHO), written in Python, was run on AWS cloud infrastructure. The results were recorded. Also, in comparison to other standard models similar to Support Vector Machine, K-Nearest Neighbour and Naive Bayes, the suggested DBN + EHO-based attack detection system was evaluated for the different parameters:

5.2 Performance Analysis

As shown in Fig. 4, the DEHO attack detection method's performance is compared to other schemes by calculating "accuracy, precision, recall, specificity and F1-Score." Positive metrics in Fig. 4 are highlighted in the study's findings. According to Fig. 4a, the accuracy of adopted DEHO model at learning rate performs the task of learning rate 70, which is 14.58%, 29.16% and 14.58% to a far greater extent than conventional models such as SVM, KNN and Naive Bayes, respectively. If the learning rate is increased, the suggested model predicts attack with more reliability. Specifically, with regards to learning, the model's accuracy rate of about ~96% was verified to be accurate by almost 90% of learning. Additionally, in Fig. 4b, DEHO is 8.89%, 20% and 89% better than SVM, KNN and Naive Bayes, respectively, in terms of precision.

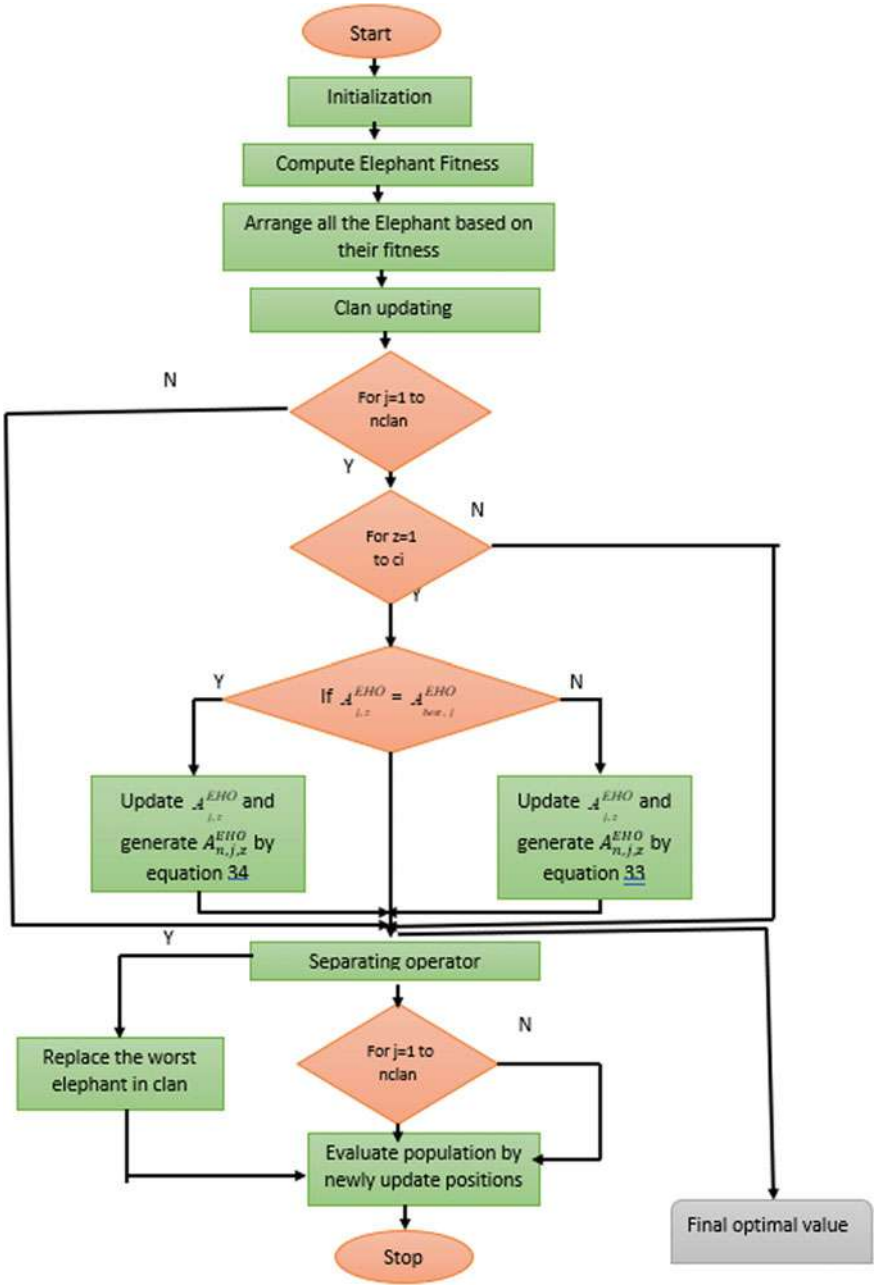
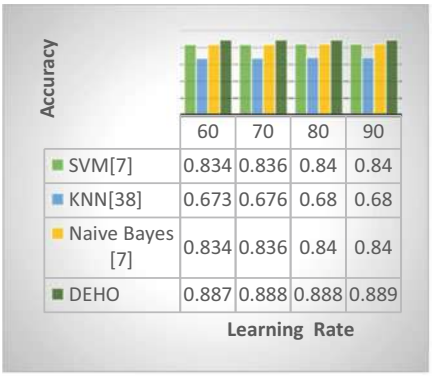
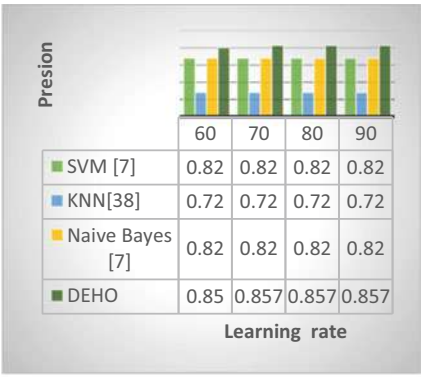


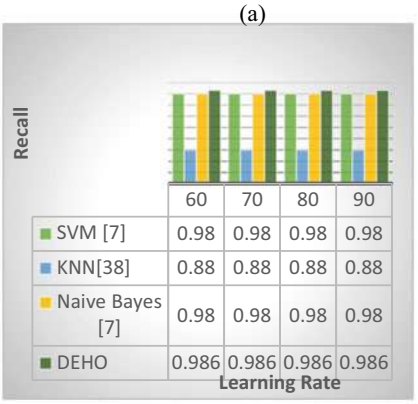
Fig. 3 Flow chart of proposed model



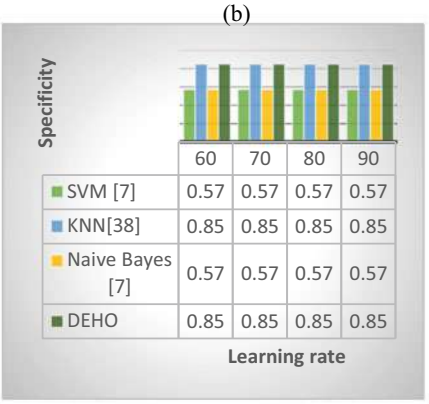
(a)



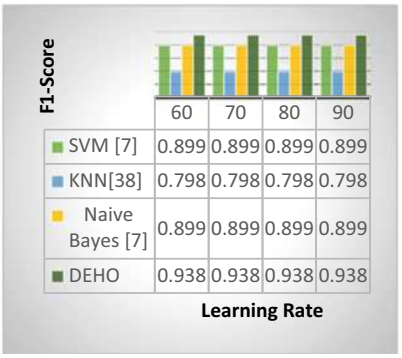
(b)



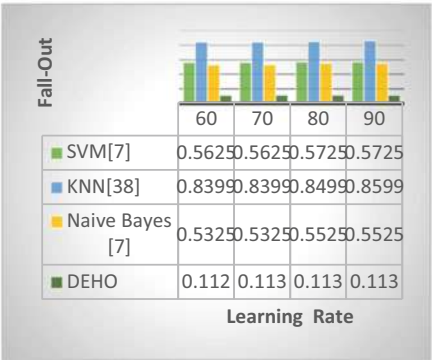
(c)



(d)



(e)



(f)

Fig. 4 Performance analysis of DEHO method over the conventional models for positive measures like **a** accuracy, **b** sensitivity, **c** specificity, **d** precision

Table 2 Overall performance analysis of adopted method over existing models

Metrics	SVM [28]	KNN [38]	Naive Bayes [28]	DEHO
Accuracy	0.83.5775	0.676882	0.836775	0.887688
Precision	0.825723	0.729072	0.825723	0.85786
Recall	0.985468	0.881325	0.985468	0.986092
Specificity	0.5725	0.859903	0.5725	0.106
F1-score	0.898551	0.798001	0.898551	0.938083
Fall out	0.5725	0.859903	0.5725	0.107

The study is shown in Fig. 4c, and the new DEHO technique has a maximum recall of 98.6% while the current SVM, KNN and Naive Bayes methods have minimum recalls of 98%, 88% and 98%, respectively. In terms of F1-score for learning rate 80, the suggested DEHO approach outperforms current models like SVM, KNN and Naive Bayes by 4%, 14.58% and 4%, respectively, in Fig. 4e.

Finally, the results show that the chosen DEHO model outperforms other existing models in identifying the existence of an attack in a network.

5.3 Overall Performance Analysis

Table 2 highlights the overall performance of the chosen DEHO technique when compared to traditional models.

When compared to other existing methods, the suggested DEHO approach outperforms the other models. According to the table, the suggested DEHO approach outperforms conventional models such as SVM, KNN and Naive Bayes in terms of accuracy by 5.2%, 21.123% and 5.1%, respectively. Furthermore, the accuracy of the DEHO model is 0.8876, whereas the precision of current models such as SVM, KNN and Naive Bayes is 0.825723, 0.729072 and 0.825723, respectively. Furthermore, the suggested DEHO achieves a low fall out, while traditional models have a large error value, which indicates incorrect detection. Thus, the overall performance of the new DEHO approach has shown that it outperforms other conventional methods.

6 Conclusion

This article has developed a novel virtualization security analytics approach for locating cloud threats. This project takes use of big data problems based on network behaviour features. The detection phase was processed under two major phases: The process of extracting features and classifying data occurred over two phases: I feature extraction and (ii) classification. Proposed holoentropy features were retrieved during

the feature extraction step. In order to classify if an attack was present in the network, the characteristics that were extracted were put through the classification process, which utilized the optimized DBN to search for an attack. To improve detection accuracy, the weights were adjusted for the DBN using EHO, which assisted the learning process. The suggested DEHO approach yielded a learning rate of 14.58, 29.16 and 14.58% better than the other conventional models such as SVM, KNN and Naive Bayes. Observations of the provided table revealed that, while the traditional models indicate poor performance with minimum values, the suggested model indicates improved recall. Thus, the DEHO method's improvement was successfully proven.

https://cse-cic-ids2018.s3.ca-central-1.amazonaws.com/Processed%20Traffic%20Data%20for%20ML%20Algorithms/Friday-02-03-2018_TrafficForML_CICFlowMeter.csv.

References

1. Mc Donnell N, Howley E, Duggan J (2020) Dynamic virtual machine consolidation using a multi-agent system to optimize energy efficiency in cloud computing. Future Gener Comput Syst 108:288–301
2. Jeddi S, Sharifian S (2020) A hybrid wavelet decomposer and GMDH-ELM ensemble model for Network function virtualization workload forecasting in cloud computing. Appl Soft Comput 88(105940)
3. Alsmadi D, Prybutok V (2018) Sharing and storage behavior via cloud computing: Security and privacy in research and practice. Comput Hum Behav 85:218–226
4. Brammya G, Deepa TA (2019) Job sceduling in cloud environment using lion algorithm. J Netw Commun Syst 2(1):1–14
5. Devagnanam J, Elango NM (2020) Optimal resource allocation of cluster using hybrid grey wolf and cuckoo search algorithm in cloud computing. J Netw Commun Syst 3(1):31–40
6. Hussain SA, Fatima M, Saeed A, Raza I, Shahzad RK (2017) Multilevel classification of security concerns in cloud computing. Appl Comput Inform 13(1):57–65
7. Xu X, Zhang Q, Maneas S, Sotiriadis S, Bessis N (2019) VMSAGE: a virtual machine scheduling algorithm based on the gravitational effect for green Cloud computing. Simul Model Practice Theory 93:87–103
8. Tummalapalli SR, Chakravarthy AS (2020) Optimal container resource allocation using hybrid SA-MFO algorithm in cloud architecture. Multimedia Res 3(1):11–20
9. Ashok Kumar C, Vimala R (2020) Load balancing in cloud environment exploiting hybridization of chicken swarm and enhanced raven roosting optimization algorithm. Multimedia Res 3(1):45–55
10. Mavridis I, Karatza H (2019) Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing. Futur Gener Comput Syst 94:674–696
11. Fernández-Cerero D, Jakóbik A, Grzonka D, Kołodziej J, Fernández-Montes A (2018) Security supportive energy-aware scheduling and energy policies for cloud environments. J Parallel Distrib Comput 119:191–202
12. Namasudra S, Devi D, Kadry S, Sundarasekar R, Shanthini A (2020) Towards DNA based data security in the cloud computing environment. Comput Commun 151:539–547
13. Kumar V, Ahmad M, Mishra D, Kumari S, Khan MK (2020) RSEAP: RFID based secure and efficient authentication protocol for vehicular cloud computing. Veh Commun 22(100213)
14. Hudic A, Smith P, Weippl ER (2017) Security assurance assessment methodology for hybrid clouds. Comput Secur 70:723–743

15. Luo L, Xing L, Levitin G (2019) Optimizing dynamic survivability and security of replicated data in cloud systems under co-residence attacks. Reliab Eng Syst Safety 192(106265)
16. Paulraj GJ, Francis SA, Peter JD, Jebadurai IJ (2018) Resource-aware virtual machine migration in IoT cloud. Future Gener Comput Syst 85:173–183
17. Zhu W, Zhuang Y, Zhang L (2017) A three-dimensional virtual resource scheduling method for energy saving in cloud computing. Future Gener Comput Syst 69:66–74
18. Shrestha M, Johansen C, Noll J, Roverso D (2020) A Methodology for Security Classification applied to Smart Grid Infrastructures. Int J Crit Infrastruct Prot 28(100342)
19. Ficco M, Choraś M, Kozik R (2017) Simulation platform for cyber-security and vulnerability analysis of critical infrastructures. J Comput Sci 22:179–186
20. Bazm M-M, Lacoste M, Südholt M, Menaud J-M (2019) Isolation in cloud computing infrastructures: new security challenges. Ann Telecommun 74(3–4):197–209
21. Nowakowski P, Bubak M, Bartyński T, Gubała T, Meizner J (2018) Cloud computing infrastructure for the VPH community. J Comput Sci 24:169–179
22. Asvija B, Eswari R, Bijoy MB (2020) Bayesian attack graphs for platform virtualized infrastructures in clouds. J Inform Secur Appl 51:102455
23. Compastié M, Badonnel R, Festor O, He R (2020) From virtualization security issues to cloud protection opportunities: an in-depth analysis of system virtualization models. Comput Secur 97(101905)
24. Patil R, Dudeja H, Modi C (2019) Designing an efficient security framework for detecting intrusions in virtual network of cloud computing. Comput Secur 85:402–422
25. Alaluna M, Ferrolho L, Figueira JR, Neves N, Ramos FM (2020) Secure multi-cloud virtual network embedding. Comput Commun 155:252–265
26. Agarwal A, Duong TNB (2019) Secure virtual machine placement in cloud data centers. Futur Gener Comput Syst 100:210–222
27. Win TY, Tianfield H, Mair Q (2018) Big data based security analytics for protecting virtualized infrastructures in cloud computing. In: IEEE transactions on big data, vol 4, no 1, 1 Mar 2018, pp 11–25. https://doi.org/10.1109/TBDATA.2017.2715335
28. Joseph L, Mukesh R (2018) Detection of malware attacks on virtual machines for a self-heal approach in cloud computing using VM snapshots. pp 249–257
29. Benkhelifa E, Bani Hani A, Welsh T, Mthunzi S, Ghedira Guegan C (2019) Virtual environments testing as a cloud service: a methodology for protecting and securing virtual infrastructures. IEEE Access 7:108660–108676. https://doi.org/10.1109/ACCESS.2019.2912957
30. Wu S, Wang S (2013) Information-theoretic outlier detection for large-scale categorical data. IEEE Trans Knowl Data Eng 25(3):589–602. https://doi.org/10.1109/TKDE.2011.261
31. Krishnan H, Elayidom MS, Santhanakrishnan T (2021) Weighted holoentropy-based features with optimised deep belief network for automatic sentiment analysis: reviewing product tweets. J Exp Theor Artif Intell 1–29
32. Wang HZ, Wang GB, Li GQ, Peng JC, Liu YT (2016) Deep belief network based deterministic and probabilistic wind speed forecasting approach. Appl Energy 182:80–93
33. Thomas R, Rangachar MJ (2018) Hybrid optimization based DBN for face recognition using low-resolution images. Multimedia Res 1(1):33–43
34. Wang GG, Deb S, Coelho LD (2015) Elephant herding optimization. https://doi.org/10.1109/ISCBI.2015.8
35. Li W, Wang G-G, Alavi AH (2020) Learning-based elephant herding optimization algorithm for solving numerical optimization problems. Knowl-Based Syst 195:105675
36. Marsaline Beno M, Valarmathi IR, Swamy SM, Rajakumar BR (2014) Threshold prediction for segmenting tumour from brain MRI scans. Int J Imaging Syst Technol 24(2):129–137. https://doi.org/10.1002/ima.22087
37. Gayathri Devi KS (2019) Hybrid genetic algorithm and particle swarm optimization algorithm for optimal power flow in power system. J Comput Mech, Power Syst Control 2(2):31–37
38. Zhang W, Chen X, Liu Y, Xi Q (2020) A distributed storage and computation k-nearest neighbor algorithm-based cloud-edge computing for cyber-physical-social systems. IEEE Access 8:50118–50130

Chapter 31
MRF–PSO: MultiRoot Finding Particle Swarm Optimization Algorithm for Nonlinear Functions

Diogo Nuno Freitas, Luiz Guerreiro Lopes, and Fernando Morgado-Dias

1 Introduction

Finding roots or zeros of nonlinear functions is a task that is often required in many areas and is still a field of study that needs attention and further research in view of its importance in many applications in science and engineering, including, e.g., signal processing, filter design, speech processing, cryptography, among many others [14, 42]. Although many iterative methods for finding such roots exist, such as the well-known Newton's method, they usually present some drawbacks, such as (i) the need for good initial approximations to the roots to assure convergence, (ii) the need for computation of first or second order derivatives of the function, or even (iii) the necessity of repeated deflations—which leads to the accumulation of rounding errors and inaccurate results—or the execution of several runs of the root finding algorithm with different initial approximations in order to allow finding all the roots [12]. The solution of systems of nonlinear functions is even more complicated, possibly being the most challenging problem in all of numerical mathematics [20, 34], since it is not known a priori whether the system admits solutions and how many solutions there are.

D. N. Freitas (✉) · F. Morgado-Dias
Madeira Interactive Technologies Institute (ITI/LARSyS/M-ITI), 9020-105 Funchal, Portugal
e-mail: diogo.freitas@m-iti.org

F. Morgado-Dias
e-mail: morgado@uma.pt

D. N. Freitas
NOVA Laboratory for Computer Science and Informatics (NOVA LINCS), 2829-516 Caparica, Portugal

D. N. Freitas · L. G. Lopes · F. Morgado-Dias
Faculty of Exact Sciences and Engineering, University of Madeira, 9020-105 Funchal, Madeira Is., Funchal, Portugal
e-mail: lopes@uma.pt

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_32

Unlike the typical numerical root finding methods, the particle swarm optimization (PSO) algorithm [9, 22] is a population-based stochastic optimization algorithm. PSO uses particles, placed randomly in the search space, intending to find the optimal solution(s) in a (possibly complex) search space S defined by a given problem. Particles then move stochastically in S by exchanging information among all other particles in the search space, or among the neighborhood of particles to which they belong, seeking the best global position(s) regarding minimization or maximization problems.

Due to the search strategy implemented in PSO, the algorithm can explore all search space and progressively converge to the most propitious areas of it. Thus, there is no need to have good initial approximations to the roots or zeros of a given function for the algorithm to converge. Moreover, the PSO algorithm does not need the derivative information to be computed, since the cost of each particle is given by evaluating the function on the particles' position, nor is it a deflation technique; therefore, high-quality solutions may be obtained. Considering all this, the PSO algorithm is suggested here as an alternative to the traditional iterative root finding methods.

With that goal in mind, a novel concurrent PSO-based algorithm for the simultaneous approximation of all the roots of a given nonlinear equation or system of nonlinear equations, named MultiRoot Finding Particle Swarm Optimization (MRF–PSO) algorithm, is introduced and tested in this paper.

A brief state-of-the-art review on the use of PSO for root finding is presented in Sect. 2. Afterward, in Sect. 3, it is explained precisely what the PSO algorithm is, and how to use it for root finding, and then the MRF–PSO algorithm is presented. Its innovative features are: (i) a novel architecture for information sharing; (ii) a technique for detecting equal roots, and (iii) an intelligent particle positioning strategy. Finally, in Sect. 4, it is shown that the MRF–PSO algorithm is effective in finding all roots.

2 Related Work

To the authors knowledge, the first approach related to the use of PSO for root finding was made by Brits and colleagues [6] in 2002 (see also [5]). These authors introduced the neighborhood best PSO (or simply nbest PSO) as a PSO algorithm variant capable of solving systems of nonlinear equations. Besides using a dynamic neighborhood approach in order to solve a given system of equations, the objective function was redefined as the minimum of the set of all combinations of two fitness functions. However, although their strategy has revealed some promising results when compared to the gbest and some lbest architectures, the number of combinations increases exponentially as the number of equations increases, causing the algorithm to face performance issues.

Wang and collaborators [40], also intending to solve systems of equations, chose to control the adjustment of the inertia term (ω) during the execution of the PSO algorithm. In order to adjust the inertia weight, those authors used a PI and a PID controller combined with a single neuron network to tune the parameters of those controllers.

Mo et al. [26] showed that combining PSO with the conjugate direction method can help the resulting algorithm to jump over local minima when solving systems of nonlinear equations. In contrast, other approaches simply opt to modify the equations of motion [16, 33, 39]. The work developed by Reyes-Sierra and collaborators [33] should be highlighted, since they proposed a PSO variant able to find both real and complex roots of a given nonlinear system of equations. In this PSO variant, the velocity update equation is changed to make the particles simultaneously explore the real and the complex plane of the space of solutions.

Abraham et al. [1] used the PSO algorithm to find the integer numerical approximations for the solutions of the Diophantine equation, which represents an elliptic curved function. Later, Pérez et al. [30] proposed a discrete version of the PSO algorithm for solving identical problems.

Amaya et al. [2] tested the PSO algorithm for solving systems of nonlinear equations and studied the effect of considering different swarm sizes. They found that, as the swarms get bigger, the PSO algorithm required fewer iterations to achieve the same accuracy level, and thus have a faster convergence speed; however, the algorithm's execution time increased. It is important to emphasize that in this case, the PSO algorithm had to be restarted at different positions in S in order to find all the multiple roots.

Zhao et al. [43], on the other hand, applied a speciation strategy, where particles are grouped in neighbors according to the Euclidean distance between them. Thus, the information sharing is reserved for particles belonging to the same species, making the algorithm able to find multiple solutions, more specifically a solution per species.

Finally, hybrid PSO-based approaches were also suggested for solving systems of nonlinear equations, involving hybridization of PSO with the Nelder–Mead method [28], the artificial bee colony (ABC) algorithm [18], and the bacterial foraging optimization (BFO) algorithm [24], and more recently Ibrahim and Tawhid [15] hybridized the cuckoo search optimization (CSO) algorithm with PSO with the same purpose.

3 Methodology

3.1 Particle Swarm Optimization

The PSO algorithm was proposed and developed by Eberhart and Kennedy [9, 22] in 1995. This optimization algorithm aims to optimize a given problem by using particles (in other words, candidate solutions), and changing the current velocity and

position in the space S of possible solutions, considering the cognitive knowledge and collective information provided by the swarm or by the neighboring particles. In this view, until a predefined stopping criterion is satisfied, each particle's velocity and position should be updated (the number of particles in the swarm is defined by the user).

The equations of motion for each particle i in the swarm, at every iteration t, were originally defined as follows [9, 22]:

$$\begin{cases} \mathbf{V}^{i}_{t+1} = \mathbf{V}^{i}_{t} + \varphi_1 R_1{}^{i}_{t}(\mathbf{p}^{i}_{t} - \mathbf{x}^{i}_{t}) + \varphi_2 R_2{}^{i}_{t}(\mathbf{l}^{i}_{t} - \mathbf{x}^{i}_{t}), \\ \mathbf{x}^{i}_{t+1} = \mathbf{x}^{i}_{t} + \mathbf{V}^{i}_{t+1}, \end{cases} \tag{1}$$

where φ_1 and φ_2 are, respectively, the cognitive and social weights chosen by the user in order to control how much the particle's own experience and the neighbors' experience should influence the particle's movement. R_1 and R_2 are uniformly distributed random vectors with values between the search space's boundaries, being responsible for adding diversity to the swarm and thus avoiding the premature convergence of the algorithm to a local minimum or maximum. Lastly, $\mathbf{p}^{i}_{t}$ and $\mathbf{l}^{i}_{t}$ denote the personal best position of particle i and the local best position, at iteration t, between all the particles belonging to the neighborhood of the particle i.

Although PSO is simple to implement and has few parameters to tune, the algorithm suffers from convergence problems caused by its weak exploration capabilities; that is, the algorithm has a high probability of converging to a position that is not granted to be the global best position for the problem but only a local optimum.

For that reason, a constriction factor K [8] was suggested as a means for controlling the local and global search strategy, ensuring the convergence of the algorithm by making a trade-off between exploration and exploitation. With this new parameter, the equations of motion were modified in order to accommodate it in such a way that [10]:

$$\begin{cases} \mathbf{V}^{i}_{t+1} = K\left[\mathbf{V}^{i}_{t} + \varphi_1 R_1{}^{i}_{t}(\mathbf{p}^{i}_{t} - \mathbf{x}^{i}_{t}) + \varphi_2 R_2{}^{i}_{t}(\mathbf{l}^{i}_{t} - \mathbf{x}^{i}_{t})\right], \\ \mathbf{x}^{i}_{t+1} = \mathbf{x}^{i}_{t} + \mathbf{V}^{i}_{t+1}, \end{cases} \tag{2}$$

where

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \tag{3}$$

with $\varphi = \varphi_1 + \varphi_1$ and $\varphi_1 = \varphi_2 = 2.05$. In this work, $\varphi = 4.1$, and thus $K \approx 0.7298$.

Although other authors (see, e.g., [7, 8]) have suggested different constriction factors and deeply studied their effect in the particles search process, as well as combined the constriction factor and the inertia factor ω [35] in the velocity update equation [19], Equation (2) is still the most used approach when combining the original PSO algorithm with a constriction factor.

Another approach for improving the exploration capabilities of PSO is to change the communication structure. The communication structure, or swarm topology, defines which particles share information with other particles. Consequently, the swarm topology determines how particles explore the search space since the global best position can be shared among a restricted set of neighbors (also known as lbest models) or with the entire swarm (the so-called gbest model).

The task of choosing the best swarm topology has been shown to be complex since no swarm topology is superior to all others [23]. Nevertheless, tests performed by the authors using PSO for root finding showed that the gbest model was the architecture with the fastest convergence speed, as well as the one that found more accurate roots; however, lbest architectures (such as mesh, pyramid, random, ring, star, and toroid) were better in escaping from local minima, and thus increasing the number of times that the PSO algorithm ends successfully (i.e., when a root is found before the stop criterion is met).

The number of particles in the swarm is one of the first parameters to be considered when using PSO. If on the one hand a small number of particles is chosen, then particles tend to take longer (i.e., require more iterations) to explore the entire search space, which may also make harder for the algorithm to escape from local optima [11]. On the other hand, when a higher number of particles is used, there are more interactions and exchanges of information in the swarm; thus, PSO has a higher probability of increasing the quality of the solution(s) found when compared to a smaller number of particles.

Tests performed by the authors with a different number of particles (24, 36, 48, and 60) revealed that increasing or decreasing the number of particles does not significantly affect the number of iterations. On the other hand, as one would expect, increasing the number of particles impose an increase in the execution time. Although swarms with 24 particles found more accurate roots, swarms with 48 and 60 were able to escape from local minima and found more roots. (This is because there are more sources of information about the most promising areas of the search space and, consequently, more information sharing.)

Setting the PSO's parameters is thus dependent on the practical application of the algorithm, being the user responsible for choosing between effectiveness (giving in this way preference to accuracy) and efficiency (prioritizing the number of [possible not accurate] roots found).

3.1.1 Adaptation of PSO for Root Finding

Using the PSO algorithm for root finding is in essence very similar to the use of PSO for minimization or maximization problems. Changes were made specifically in the way of computing the cost of each particle—now given by the absolute value of the evaluation of the function at the current position of the particle—as well as ensuring that the particles stay within the search space bounds.

It is important to note that since the PSO algorithm was designed for optimization problems, where particles with lower cost (considering minimization problems) have

a higher probability of being at the best global position, the same may not happen with the roots of a given function. This means that if the absolute value of the result of the function evaluation at a certain position was not taken as the cost of the corresponding particle, particles far from the function root with negative costs would have more influence than particles that may even be closer to the root but have a higher fitness value.

Besides that, even though in the first iteration of the PSO algorithm, particles are inevitably positioned within the boundaries of the search space, during the later iterations, some particles can leave the search domain and thus the feasible region. Consequently, particles may take some iterations to return to the feasible region again, or they may not even come back and influence other particles to move toward invalid positions, affecting the quality of the solutions found by the algorithm. Thus, it is important to ensure that particles stay within their bounds during the execution of the algorithm.

In this work, boundary control was achieved by resetting the particle's dimensions that exceed its bounds to the nearest available boundary, such that:

$$(\mathbf{x}^{\,i}_{t+1})_d = \begin{cases} (\mathbf{x}_{\min})_d & \text{if } (\mathbf{x}^{\,i}_{t+1})_d < (\mathbf{x}_{\min})_d, \\ (\mathbf{x}_{\max})_d & \text{if } (\mathbf{x}^{\,i}_{t+1})_d > (\mathbf{x}_{\max})_d, \\ (\mathbf{x}^{\,i}_{t+1})_d & \text{otherwise.} \end{cases} \tag{4}$$

The remaining search strategy was not directly modified, since particles should always move toward the best fit particle, in this case the particle with the lowest root value.

3.2 MultiRoot Finding Particle Swarm Optimization

The traditional PSO algorithm finds, at each execution, only one root approximation. However, in many problems in Engineering, Physics, Chemistry, and Economics, the common situation is to have and find multiple roots. Although the PSO algorithm could be executed several times in order to find the multiple roots (in the sense of at least a few, regardless of their multiplicities) of a given nonlinear function, the algorithm will tend to converge to the same roots, while others are seldom found.

The MRF–PSO algorithm, introduced in this section, was developed to find the distinct roots of a given nonlinear equation or system of nonlinear equations by using multiple swarms that explore different areas of the same search space simultaneously.

In the proposed PSO-based algorithm, multiple swarms are placed in the search space and the algorithm stops when no swarms are exploring it. Swarms do not exchange information among them, but between each of them and the master. The master, as can be seen in Fig. 1, stores the information about the roots found by each swarm and is responsible for synchronizing the execution of each swarm, and also for spreading the knowledge gained during the execution of the algorithm. In this figure, r denotes the list of roots found and r_n the root found by swarm n.

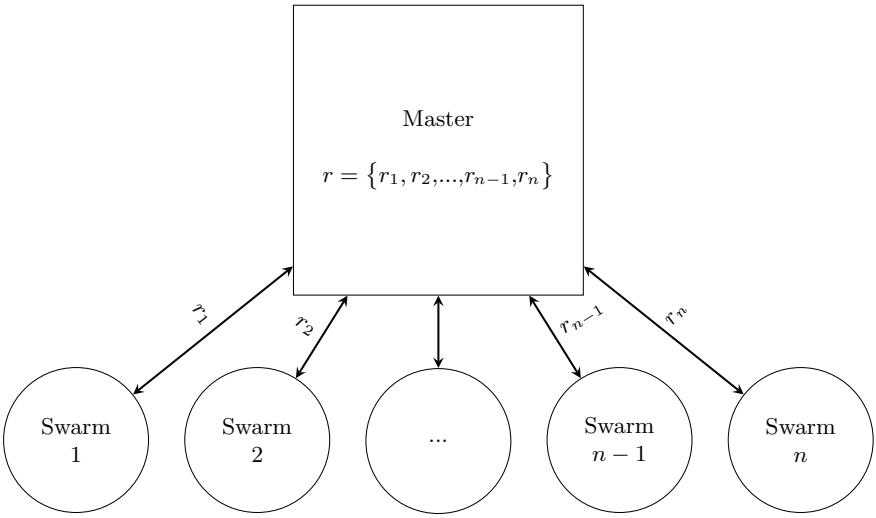


Fig. 1 MRF–PSO architecture for sharing information about the roots found during the search process of each swarm

The MRF–PSO algorithm can be seen as multiple instances of the PSO algorithm running in parallel, where each swarm explores the search space and waits for the other swarms to finish in order to exchange information. Some strategies, however, must be applied for detecting equal roots and to prevent particles to search in the same areas of the search space where a root was previously found.

It is important to note here that every instance of the PSO algorithm can be used with any number of particles, different equations of motion or even distinct swarm topologies. However, since the authors focus is on the algorithm accuracy, and also taking into account the tests previously performed, Equation (2) was used to update both the velocity and position of particles, being the swarms formed with 24 particles organized in a gbest model. These settings were applied to every instance of the PSO algorithm.

This algorithm was developed and tested using the Python general-purpose programming language at version 3.7. In order to leverage the execution time, the Ray framework [27] was used to create a system process for each swarm. Thus, in multicore processor systems, several swarms can explore the search space simultaneously. Generally speaking, this means that the time required to find a root is approximately the same as that required to find multiple roots.

3.2.1 Detecting Equal Roots

In order to allow the master to detect equal roots, a new parameter was added to the PSO algorithm: the absolute tolerance parameter (e), given by:

$$nr(\mathbf{x}, r, e) = \begin{cases} 1, & \left|(\mathbf{x})_i - (r_j)_i\right| > e \ , \forall i \in 1, ..., d, \\ & \qquad\qquad j \in 1, \ldots, z, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where d is the number of dimensions of the search space, r is the set of roots found by the MRF–PSO algorithm, and z is the number of roots in r, i.e., $z = |r|$. Thus, if $nr(\mathbf{x}, r, e) = 1$, then $\mathbf{x}$ is appended to r, meaning that a new distinct root was found.

Since the execution of the nr function on all swarms is dependent on the order of evaluation, swarms are sorted by ascending order of its cost value (i.e., the root value), such that swarms that found a lower cost value have a higher probability of being appended to r.

To clarify that, consider that $z = 0$, e.g., after the first iteration. (In the MRF–PSO algorithm, the iteration number is incremented after the execution of the PSO algorithm in all swarms.) Now consider that four swarms were placed in a two-dimensional search space and $e = 0.1$. Swarm 1 found a root at $(1, 2)$ with a root value of 0.01. On the other hand, Swarm 2 found a root at $(1.01, 2.01)$, but with a root value of 0.001. Swarms 3 and 4 found distinct root positions, then the root positions found by those two swarms will be appended to r. Nevertheless, Swarm 1 and Swarm 2, according to the absolute tolerance parameter, found the same root.

In this example, the position found by Swarm 2 represents a more accurate root position than the position found by Swarm 1. However, if Swarm 1 was the first swarm to be evaluated in nr, then its root position will be appended to r, and the root found by Swarm 2 will be discarded, since a similar root was already found (in this case, by Swarm 1). Thus, not considering Swarms 3 and 4, if the swarms were sorted by their cost value, then the root found by Swarm 2 would be appended to r, and Swarm 1 would be discarded since a similar root (found by Swarm 2) was already in r.

The example presented before is intended to show the importance of sorting swarms by their cost value after running the PSO algorithm in each swarm. It is important to note that the root values presented before were chosen to illustrate why ordering swarms is important and do not represent by any means the required accuracy imposed to the MRF–PSO algorithm.

3.2.2 Particle Positioning

Besides the absolute tolerance parameter, another parameter was added to the MRF–PSO algorithm, namely the maximum time to live parameter. This parameter defines how many times should the particles of a given swarm be repositioned in the search space.

Swarms (and their particles) are repositioned in the search space when they find a root that was already found before; however, unlike the traditional PSO algorithm initialization, particles are not distributed uniformly in all search space.

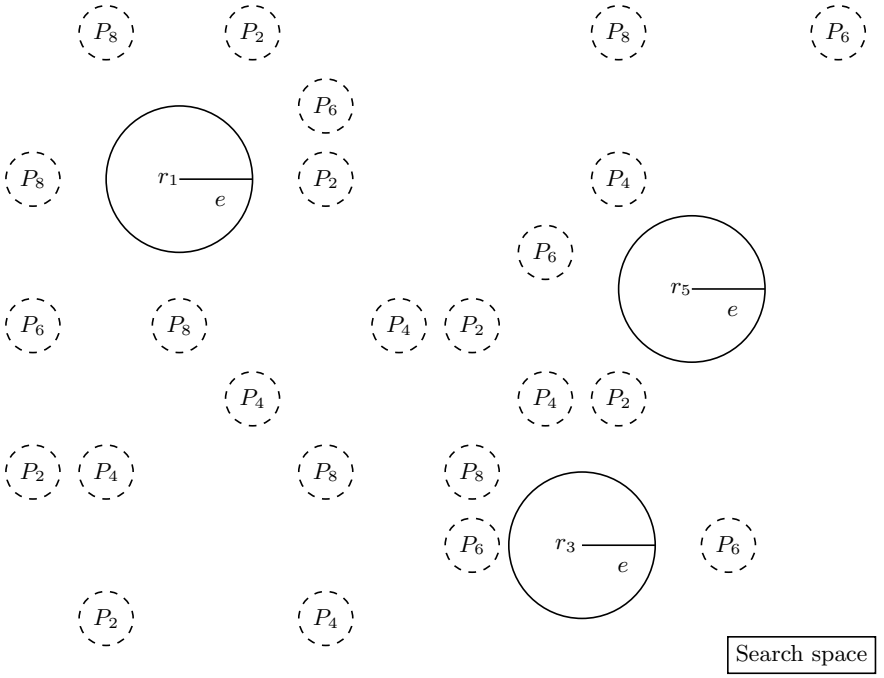


Fig. 2 Example of particle repositioning when three roots were discovered by Swarms 1, 3, and 5

As mentioned before, the master is the mean for sharing information about the roots found by other swarms. This information is used to not place particles at locations in the search space that have already been detected as roots. Thus, the algorithm has a higher probability of converging to a root that was not found before. It is noteworthy that swarms that found distinct roots are removed from the search space.

As an example, consider Fig. 2. Three roots were detected by Swarm 1, Swarm 3, and Swarm 5. Since those swarms successfully found roots that were not found before, they were removed from the search space. Particles of the remaining swarms, in turn, are repositioned uniformly in the search space in areas where roots were not detected, i.e., in the remaining search space.

Besides that, the number of iterations of the repositioned swarms is reset since their particles will explore a new search area. However, when the number of times that each swarm is repositioned in the search space reaches the maximum time to live, swarms are removed from the search space and are not repositioned again.

Each PSO instance iterates until the root value is lower than a preestablished tolerance ε. In addition to setting the stopping criterion by the value of the objective function at a given point in order to ensure that the algorithm will end, it was also defined that the algorithm can have a maximum of 5 000 iterations for every example.

It is important to note here that unsuccessful swarms, i.e., swarms that were not able to find any root until the maximum number of iterations was reached are directly removed from the search space without being repositioned, even though they have never before been repositioned.

The flowchart of Fig. 3 summarizes the workflow of the MRF–PSO algorithm.

4 Results and Discussion

A set of well-known test functions [17, 32] was chosen to test if the MRF–PSO algorithm is able to find, within a reasonable execution time, all roots of a given nonlinear equation or system of nonlinear equations. The convergence of the algorithm was also tested by the average number of roots found.

It is important to note that the chosen test functions are nonlinear equations or systems of nonlinear equations, and that finding approximations for roots of nonlinear functions is one of the most complicated and challenging problems in numerical mathematics, as mentioned before.

It is also noteworthy that, for root finding, systems of nonlinear equations were considered as a single nonlinear equation, such that:

$$f(\cdot) = |f_1(\cdot)| + |f_2(\cdot)| + \cdots + |f_n(\cdot)|, \tag{6}$$

where $f_i(\cdot)$ denotes the i-th nonlinear equation of a given system of nonlinear equations.

Each test function was tested 500 times, and the results are presented below in terms of the average values of all runs. Except for Price 4 Function (Example 6) where $\epsilon = 10^{-5}$, for all other test functions $\epsilon = 10^{-12}$. Finally, $e = 0.1$, and the time to live parameter was set to be equal to 5. The tests were executed in a laptop machine with a Dual-Core Intel Core i7 CPU running at 1.70 GHz, and 8 GB of RAM.

The effectiveness was computed by dividing the average number of roots found by the total number of known roots. The execution time, given in seconds, is the time required by the algorithm to finish its execution, that is, until no more swarms are exploring the search space, even though the algorithm may have discovered all the roots at the end of the first few seconds. This metric is useful to study since in most situations the number of roots is not known. Likewise, it is also reported the execution time required by the algorithm to find all the known roots, even if the algorithm continues to run.

Example 1 (Himmelblau's objective function)

$$f(x, y) = \begin{cases} 4x^3 + 4xy - 42x + 2y^2 - 14 = 0 \\ 2x^2 + 4xy + 4y^3 - 26y - 22 = 0 \end{cases} \tag{7}$$

- Number of swarms placed in the search space: 18 swarms.

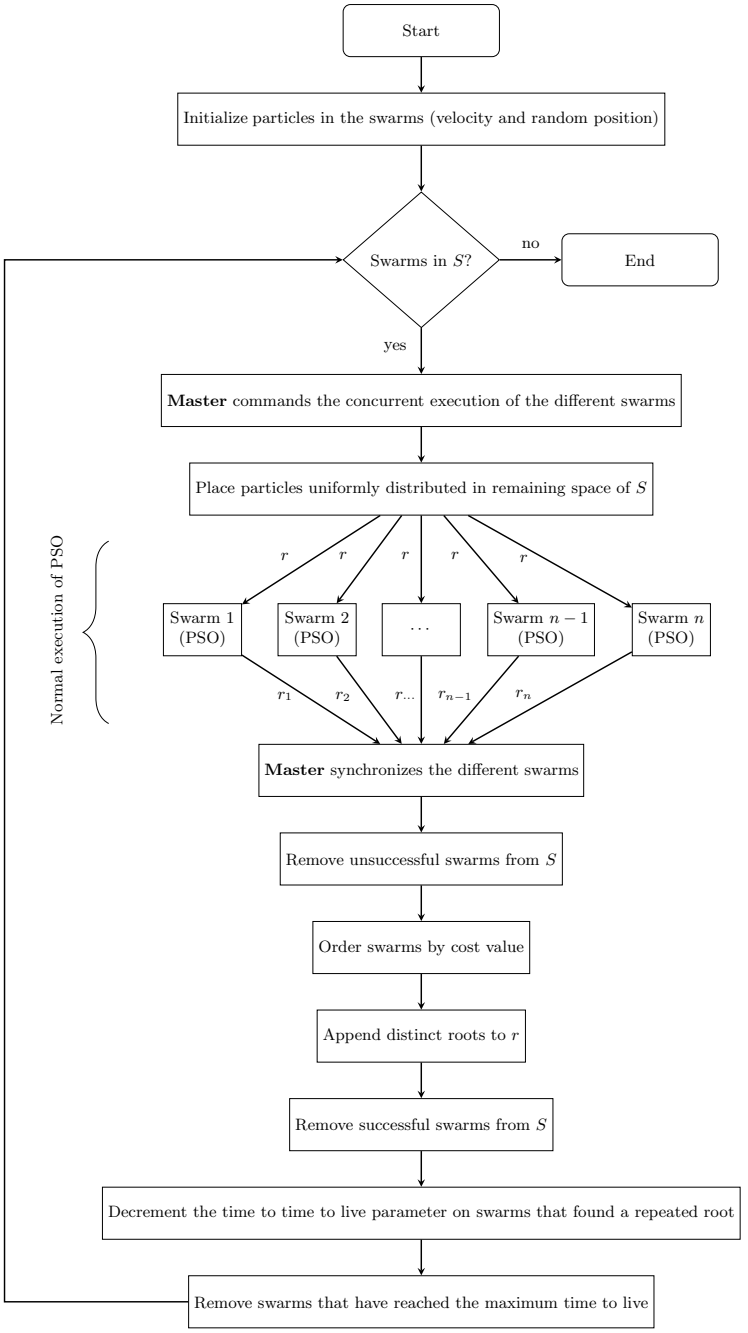


Fig. 3 Flowchart of the MRF–PSO algorithm

- Average number of roots found: 8.81 (effectiveness: 98 %).
- Number of know roots: 9.
- Average total execution time: 31.62 s (22.74 s to find all roots).

Example 2 (Merlet problem)

$$f(x, y) = \begin{cases} -\sin(x)\cos(y) - 2\cos(x)\sin(y) = 0 \\ -\cos(x)\sin(y) - 2\sin(x)\cos(y) = 0 \end{cases} \tag{8}$$

- Number of swarms placed in the search space: 26 swarms.
- Average number of roots found: 13 (effectiveness: 100 %).
- Number of know roots: 13.
- Average total execution time: 39.03 s (20.05 s to find all roots).

Example 3 (Floudas problem)

$$f(x, y) = \begin{cases} 0.5\sin(xy) - 0.25\dfrac{y}{\pi} - 0.5x = 0 \\ \left(1 - \dfrac{0.25}{\pi}\right)\Big(\exp(2x) - e\Big) + e\dfrac{y}{\pi} - 2ex = 0 \end{cases} \tag{9}$$

- Number of swarms placed in the search space: 4 swarms.
- Average number of roots found: 2 (effectiveness: 100 %).
- Number of know roots: 2.
- Average total execution time: 6.40 s (4.03 s to find all roots).

Example 4 (Robotics–Planar Parallel Manipulators)

$$f(x) = k_0 + k_2x^2 + k_4x^4 + k_6x^6 + \left(k_1x + k_3x^3 + k_5x^5\right)\sqrt{1 - x^2} = 0, \tag{10}$$

where:

$$k_0 = 3.9852, k_1 = -8.8575, k_2 = -10.039, k_3 = 20.091,$$

$$k_4 = 7.2338, k_5 = -11.177, \text{and } k_6 = -1.17775.$$

- Number of swarms placed in the search space: 12 swarms.
- Average number of roots found: 4.68 (effectiveness: 78 %).
- Number of know roots: 6.
- Average total execution time: 23.77 s (22.52 s to find all roots).

Example 5 (Wayburn Seader 1 Function)

$$f(x, y) = \left(x^6 + y^4 - 17\right)^2 + (2x + y - 4)^2 \tag{11}$$

- Number of swarms placed in the search space: 2 swarms.
- Average number of roots found: 2 (effectiveness: 100 %).

Table 1 Comparison of MRF–PSO with some of the results available in the literature for Example 1

Metric	Value
Effectiveness MRF–PSO	98%
Effectiveness [32]	82%
Effectiveness [36]	100%
Execution time MRF–PSO	31.62 (22.74)
Execution time [32]	19.917
Execution time [36]	5

- Number of know roots: 2.
- Average total execution time: 5.62 s (2.70 s to find all roots).

Example 6 (Price 4 Function)

$$f(x, y) = \left(2x^3y - y^3\right)^2 + \left(6x - y^2 + y\right)^2 \tag{12}$$

- Number of swarms placed in the search space: 6 swarms.
- Average number of roots found: 3 (effectiveness: 100 %).
- Number of know roots: 3.
- Average total execution time: 7.91 s (4.59 s to find all roots).

The MRF–PSO algorithm showed to be a viable approach for finding the roots of a given nonlinear equation or system of nonlinear equations. Even though it presented difficulties with Example 4, the effectiveness was always 98% or more for the other examples.

Results of Examples 1 to 4 were compared in terms of the number of roots found and execution time with some results available in the literature about the use of some stochastic optimization algorithms different from PSO for solving systems of nonlinear equations [13, 31, 32, 36, 38]. These results were summarized in a paper authored by Ramadas et al. [32], that also proposed a metaheuristic algorithm for finding multiple roots of systems of nonlinear equations. The results of the comparison between MRF–PSO and other approaches are presented in Tables 1, 2, 3, and 4.

As can be seen, despite not being able to be superior in all considered approaches, the MRF–PSO revealed to be a promising approach in terms of effectiveness, especially when compared to the results presented by Ramadas and collaborators [32].

However, it is possible to note that, in most cases, the MRF–PSO required a higher execution time when compared with the other algorithms. Possible justifications for this result may arise from the different algorithm accuracy considered and from the programming language used to develop both the algorithms. Nevertheless, it is still a limitation of the MRF–PSO algorithm to be considered in the future work.

Table 2 Comparison of MRF–PSO with some of the results available in the literature for Example 2

Metric	Value
Effectiveness MRF–PSO	100%
Effectiveness [32]	61%
Effectiveness [36]	100%
Effectiveness [13]	100%
Effectiveness [38]	100%
Execution time MRF–PSO	39.03 (20.05)
Execution time [32]	0.881
Execution time [36]	46
Execution time [13]	20
Execution time [38]	0.03

Table 3 Comparison of MRF–PSO with some of the results available in the literature for Example 3

Metric	Value
Effectiveness MRF–PSO	100%
Effectiveness [32]	100%
Effectiveness [36]	100%
Effectiveness [13]	100%
Effectiveness [38]	100%
Execution time MRF–PSO	6.40 (4.03)
Execution time [32]	0.505
Execution time [36]	0.607
Execution time [13]	0.461
Execution time [38]	0.03

Table 4 Comparison of MRF–PSO with some of the results available on the literature for Example 4

Metric	Value
Effectiveness MRF–PSO	78%
Effectiveness [32]	88%
Effectiveness [31]	100%
Execution time MRF–PSO	23.77 (22.52)
Execution time [32]	5.345
Execution time [31]	n/a

5 Conclusion

The MRF–PSO algorithm described in this paper is a bio-inspired stochastic algorithm that uses the search strategy implemented in the PSO algorithm in order to compute approximations to the roots of a given nonlinear equation or system of nonlinear equations.

Multiple swarms are placed in the search space, and following a parallel architecture, they are able to explore the search space at the same time, using different cores of a multicore processor, and report the results to the master. The master, in turn, stores all the roots that were found during the search process and is responsible for randomly positioning the particles of the swarms uniformly in the search areas where a root was not found yet.

The MRF–PSO algorithm was tested with different, commonly used nonlinear functions. The results of the executions were compared with other results available in the literature, and although the MRF–PSO can be considered an effective algorithm for finding the multiple roots of a given function, it showed some issues related to the time required to compute approximations for all roots. Nevertheless, this contribution can certainly be exploited in real-world problems.

As a work for future direction, it would be interesting to try the use of dynamic neighbors with the concept of stereotyping [3, 21, 37], fully informed particle swarm (FIPS) [25], and the adaptive PSO (APSO) [41]. These ideas would, for instance, improve the convergence behavior of each PSO instance and thus enhance the overall effectiveness and efficiency of the MRF–PSO algorithm.

The way the algorithm is designed does not enable it to solve nonlinear equations or systems of nonlinear equations subject to a set of constraints. A penalty function strategy, like the one introduced by Parsopoulos and Vrahatis [29], can be a good starting point.

Finally, the authors are also considering rewriting the MRF–PSO algorithm in another computer language, such as the Julia programming language [4], in order to reduce the total execution time.

Acknowledgements This work was supported by the Project MITIExcell (Project—UIDB/50009/2020), co-financed by Regional Development European Funds for the "Operational Program Madeira 14–20"—Priority Axis 1 of the Autonomous Region of Madeira, Number M1420-01-0145-FEDER-000002. In addition, the funding from LARSyS–FCT Pluriannual funding 2020–2023 is acknowledged.

References

1. Abraham S, Sanyal S, Sanglikar M (2010) Particle swarm optimisation based Diophantine equation solver. Int J Bio-Inspired Comput 2(2):100–114
2. Amaya I, Cruz J, Correa R (2011) Real roots of nonlinear systems of equations through a metaheuristic algorithm. DYNA-Revista de la Facultad de Minas 78(170):15–23

3. van den Bergh F (2002) An analysis of particle swarm optimizers. Ph.D. thesis, University of Pretoria, Pretoria, South Africa
4. Bezanson J, Karpinski S, Shah VB, Edelman A (2012) Julia: a fast dynamic language for technical computing. arXiv:1209.5145
5. Brits R (2002) Niching strategies for particle swarm optimization. Master's thesis, Faculty of Natural & Agricultural Science, University of Pretoria, Pretoria, South Africa
6. Brits R, Engelbrecht AP, van den Bergh F (2002) Solving systems of unconstrained equations using particle swarm optimization. In: IEEE international conference on systems, man and cybernetics, SMC, vol 3. Yasmine Hammamet, Tunisia, pp 6:1–6:6
7. Bui LT, Soliman O, Abbass HA (2007) A modified strategy for the constriction factor in particle swarm optimization. In: 3rd Australasian Conference on Artificial Life and Computational Intelligence. ACAL. Geelong, Australia, pp 333–344
8. Clerc M (1999) The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: IEEE Congress on Evolutionary Computation, CEC, vol 3. Washington, WA, USA, pp 1951–1957
9. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: 6th international symposium on Micro Machine and Human Science, MHS. Nagoya, Japan, pp 39–43
10. Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: IEEE Congress on Evolutionary Computation, CEC, vol 1. La Jolla, CA, USA, pp 84–88
11. El-Gallad A, El-Hawary M, Sallam A, Kalas A (2002) Enhancing the particle swarm optimizer via proper parameters selection. IEEE Canadian Conference on Electrical and Computer Engineering, vol 2. CCECE, Winnipeg, Canada, pp 792–797
12. Freitas D, Lopes LG, Morgado-Dias F (2018) A neural network based approach for approximating real roots of polynomials. In: International Conference on Mathematical Applications. ICMA. Funchal, Portugal, pp 44–47
13. Hirsch MJ, Pardalos PM, Resende MG (2009) Solving systems of nonlinear equations with continuous GRASP. Nonlinear Anal: Real World Appl 10(4):2000–2006
14. Huang DS (2004) A constructive approach for finding arbitrary roots of polynomials by neural networks. IEEE Trans Neural Netw 15(2):477–491
15. Ibrahim AM, Tawhid MA (2019) A hybridization of cuckoo search and particle swarm optimization for solving nonlinear systems. Evol Intelli 12(4):541–561
16. Jaberipour M, Khorram E, Karimi B (2011) Particle swarm algorithm for solving systems of nonlinear equations. Comput Math Appl 62(2):566–576
17. Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. Int J Math Modelling Numer Optimisation 4(2):150–194
18. Jia R, He D (2012) Hybrid artificial bee colony algorithm for solving nonlinear system of equations. In: 8th international conference on Computational Intelligence and Security, CIS. Guangzhou, China, pp 56–60
19. Kar R, Mandal D, Bardhan S, Ghoshal SP (2011) Optimization of linear phase FIR band pass filter using particle swarm optimization with constriction factor and inertia weight approach. In: IEEE symposium on Industrial Electronics and Applications, ICIEA, Langkawi, Malaysia, pp 326–331
20. Karr CL, Weck B, Freeman LM (1998) Solutions to systems of nonlinear equations via genetic algorithms. Eng Appl Artif Intelli 11(3):269–375
21. Kennedy J (2000) Stereotyping: improving particle swarm performance with cluster analysis. In: IEEE Congress on Evolutionary Computation, CEC, vol 2. La Jolla, CA, USA, pp 1507–1512
22. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: International Conference on Neural Networks, vol 4. ICNN, Perth, Australia, pp 1942–1948
23. Liu Q, Wei W, Yuan H, Zhan ZH, Li Y (2016) Topology selection for particle swarm optimization. Information Sci 363:154–173
24. Mai XF, Li L (2013) Bacterial foraging algorithm based on PSO with adaptive inertia weigh for solving nonlinear equations systems. Adv Mater Res 655:940–947

25. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. IEEE Trans Evol Comput 8(3):204–210
26. Mo Y, Liu H, Wang Q (2009) Conjugate direction particle swarm optimization solving systems of nonlinear equations. Comput Math Appl 57(11):1877–1882
27. Moritz P, Nishihara R, Wang S, Tumanov A, Liaw R, Liang E, Elibol M, Yang Z, Paul W, Jordan MI, Ray IS (2018) Ray: a distributed framework for emerging AI applications. In: 13rd USENIX symposium on Operating Systems Design and Implementation, OSDI, Carlsbad, CA, USA, pp 561–577
28. Ouyang A, Zhou Y, Luo Q (2009) Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations. In: IEEE international conference on Granular Computing, GrC, Nanchang, China, pp 460–465
29. Parsopoulos KE, Vrahatis MN (2002) Particle swarm optimization method for constrained optimization problem. In: Frontiers in artificial intelligence and applications, vol 76. IOS Press, Fairfax, VA, USA, pp 214–220
30. Pérez O, Amaya I, Correa R (2013) Numerical solution of certain exponential and non-linear Diophantine systems of equations by using a discrete particle swarm optimization algorithm. Appl Math Comput 225:737–746
31. Pourjafari E, Mojallali H (2012) Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. Swarm Evol Comput 4:33–43
32. Ramadas G, Fernandes E, Rocha A (2018) Finding multiple roots of systems of nonlinear equations by a hybrid harmony search-based multistart method. Appl Math Information Sci 12(1):21–32
33. Reyes-Sierra S, Plata-Rueda J, Correa-Cely R (2012) Real and/or complex roots calculation of nonlinear equations systems through modified particle swarm optimization. Ingeniería y Universidad 16(2):349–362
34. Rice JR (1993) Numerical methods, software, and analysis, 3rd edn. Academic Press, Boston
35. Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: IEEE World Congress on Computational Intelligence, WCCI. Anchorage, AK, USA, pp 69–73
36. Silva RM, Resende MG, Pardalos PM (2014) Finding multiple roots of a box-constrained system of nonlinear equations with a biased random-key genetic algorithm. J Global Optimization 60(2):289–306
37. Suganthan PN (1999) Particle swarm optimiser with neighbourhood operator. In: IEEE Congress on Evolutionary Computation, CEC, vol 3. Washington, WA, , pp 1958–1962
38. Tsoulos I, Stavrakoudis A (2010) On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods. Nonlinear Anal: Real World Appl 11(4):2465–2471
39. Turgut OE, Turgut MS, Coban MT (2014) Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations. Comput Math Appl 68(4):508–530
40. Wang Q, Zeng J, Jie J (2007) Modified particle swarm optimization for solving systems of equations. In: International Conference on Intelligent Computing, ICIC, Qingdao, China, pp 361–369
41. Zhan Z, Zhang J, Li Y, Chung HS (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybernetics Part B (Cybernetics) 39(6):1362–1381
42. Zhang Y, Zeng Z (2009) A new method for simultaneous extraction of all roots of algebraic polynomial. In: International conference on Computational Intelligence and Security, CIS, vol 1. Beijing, China, pp 197–200
43. Zhao J, Sun J, Lai CH, Xu W (2011) An improved quantum-behaved particle swarm optimization for multi-peak optimization problems. Int J Comput Math 88(3):517–532

Chapter 32
Investigation of Impact of Mobile Phone Radiations on EEG/ECG and Modeling of Their Coherence Using ANN Ensemble

Suman Pattnaik, Balwinder S. Dhaliwal, and Shyam Sundar Pattnaik

1 Introduction

The demand of the mobile communication devices is increasing continuously due to requirement of seamless sharing of information and to stay connected. The continuous and prolonged usage of mobile phones is making us to be under the impact of RF radiations. The exposure to these radiations leads to various biological, physiological, and psychological effects as reported in the literatures by many researchers. Wessapan et al. [1] analyzed the effect of near field exposure on biological tissues in terms of flow of fluid, transfer of heat and SAR. They found that the velocity field, temperature distribution, and SAR are influenced by exposure distance. Rad et al. [2] investigated the effect of radiations from cellular phones on the blood viscosity. A, B, AB, and O group bloods were taken from healthy subjects and stored in EDTA as anticoagulant. The samples of these bloods were radiated with the EM fields for one hour with an interval of 15 min. It was found that the viscosity of blood got changed with the time of exposure. Lu and Huang [3] discussed the biological effects of mobile phone radiations on the human beings and found that headache was the more visible effect and was a reversible physiological effect. Lin [4] observed that the students who frequently used cellular phones showed less focus and also delayed and inaccurate responses. The study of effects of radiations of mobile phone studied by Mat et al. [5] and Kaur et al. [6] reported that anxiety, sleep disorders, loss in memory, ear pain, and temperature increase in adjoining nerves, problematical birth in case of pregnant woman, and effect on eyes are the major problems due to the usage of mobile phones. Repacholi [7] reported that headache, memory loss, insomnia,

S. Pattnaik (✉)
ECE Department, SSIET, Dera-Bassi, Punjab, India
e-mail: sumanbhagat1199@gmail.com

B. S. Dhaliwal · S. Sundar Pattnaik
National Institute of Technical Teachers' Training and Research, Chandigarh, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_33

dizziness, rashes, tingling, fatigue, heating of ear, irritation, and many other psychological, behavioral, and biological effects have been observed among the users of the mobile phones. Guillet et al. [8] presented that the interactions among organs like heart and brain are significant in physiological processes and analyzed ECG, EEG, and respiration rhythms recorded during sleep from polysomnography databases to track rhythms coherence. The time-averaged coherence has been compared with heart rate variability spectra to investigate the common and altered features.

In this proposed work, the heart rate variability and other parameters have been analyzed to know the effects of radiations on the heart of the mobile users. Also, the EEG recorded simultaneously with ECG during the usage of mobile phones has been analyzed using various linear and non-linear parameters to study the effects of radiations in brain. The interaction between heart and brain has been assessed by the coherence which has been calculated between ECG and EEG signals to study the extent of synchronization and relation among the signals. The rest of the paper has been organized as follows. The Sect. 2 describes the experimental setup and methodology of the proposed work. The impact of mobile phone radiations on ECG and EEG signals has been discussed in Sect. 3. The Sect. 4 presents the effect of mobile phone radiations on the coherence between ECG and EEG signals. The development of an ANN ensemble-based model to predict the impact of mobile phone radiations on EEG from the impact on ECG has been described in Sect. 5. The presented work has been concluded in Sect. 6 which is followed by the list of the references cited in this work.

2 Experimental Setup and Methodology

While doing the experiment to study the effects of mobile phone radiations, various precautions have been taken regarding the subjects and the experimental setup. For this study, 75 healthy M. Tech. students of National Institute of Technical Teachers Training and Research, Chandigarh, India have been considered. The age, height, weight, and BMI index were almost same for all the subjects. The average age of the students was 22.2 years with standard deviation of 2.27 years. All the subjects were healthy, and no one was taking any kind of medicine. These students used their mobile phones while recordings were done. The mobile phones used by the subjects in the experiment were of reputed manufactures like Samsung, Nokia, and Motorola having SAR values between 0.67–1.14 W/kg. During measurements, it was ensured that there is no any other type of electromagnetic exposure present in the lab. The participation of all individuals in the experiments was voluntary and with full knowledge of motive of study. The subjects were given necessary directions so that motive of the experiment is fulfilled. The measurement of electromagnetic field in the laboratory was done, and this information was used in the pre-processing stage.

The ECG and EEG signals of each subject were acquired simultaneously. ECG signal has been acquired using Biopac MP100 system fixed at a 200 Hz sampling frequency and notch filter at 50 Hz. The recording has been done with three electrodes

placed on left arm wrist with positive polarity, on right arm wrist with negative polarity, and one ground electrode at right leg ankle. EEG data have been obtained by recording waveforms using 20 electrodes placed as per the 10–20 international system configuration using RMS maximus 24–32 setup. For all subjects, the ECG and EEG signals were measured in morning when they were not using mobile phones, i.e., when no radiation exposure was there, and subjects were in relax mood. The ECG and EEG signals measured in this setup have been named as ideal mode signals. The second mode of measurement was under the exposure from the standard mobile phone when the call was underway, and the call was initiated by the subject. It has been denoted by Tx mode, i.e., transmission mode. The maximum call duration was fixed as 5 min. This ECG and EEG acquisition of subjects correspond to their communication in their daily routine. The experimental setup and methodology have been described in more details in [9].

3 Results of Investigations on ECG and EEG Signals

3.1 Impact of Mobile Phones Radiations on ECG

To investigate the impact of electromagnetic radiations of mobile phones on ECG, the time domain parameters have been examined. The analyzed time domain parameters include mean heart rate (HR), root mean square successive RR interval differences (RMSSD), standard deviation of heart rate (STD HR), standard deviation of all normal sinus RR intervals (SDNN) in ms, RR triangular index, triangular interpolation of RR interval histogram (TINN), and the percentage of the number of RR interval differences (pNN50) which are equal to or more than 50 ms [10]. The mean values of these parameters along with the units, standard deviation, the *t*-value, and *p*-value from the statistical results for transmission mode of mobile communications (denoted as Tx) have been shown in Table 1. These values have been compared with the mean and standard deviation values for the ideal mode as shown in Table 1.

As evident from Table 1, there are variations in mean values of all time domain parameters due to radiations from the mobile phones. This change is expressive as though the duration of the call is small, and the *p*-value is also showing that the change is statistically significant for two cases.

3.2 Impact of Mobile Phones Radiations on EEG

The impact of mobile phones' radiations on the human EEG has been studied by analyzing delta, theta, alpha, beta, and gamma brain waves under Tx communications mode. The recorded EEG signals have been examined with the help of different parameters, namely correlation dimension (CD), Hurst exponent (HE),

Table 1 Time domain parameters values

Parameter	Mode of measurement	Mean	SD	*t*-value	*p*-value
Mean HR (ms)	Ideal mode	80.488	11.394	–	–
	Tx mode	80.744	11.376	−0.324	0.748
RMSSD (ms)	Ideal mode	47.064	25.80	–	–
	Tx mode	40.28	21.69	2.740	0.011*
STD HR (bpm)	Ideal mode	5.134	1.4316	–	–
	Tx mode	5.466	1.725	−1.019	0.318
SDNN (ms)	Ideal mode	54.6667	15.8346	–	–
	Tx mode	55.7157	13.6020	−0.390	0.700
RR triangular Index	Ideal mode	10.782	3.386	–	–
	Tx mode	10.81	3.262	−0.051	0.959
TINN (ms)	Ideal mode	257.5	96.60	–	–
	Tx mode	248.33	94.05	0.716	0.482
pNN50 (%)	Ideal mode	18.67	15.74	–	–
	Tx mode	14.35	11.60	2.217	0.036*

* *p-values which are significant*

largest Lyapunov exponent (LLE), approximate entropy (ApEn), and the power values of the brain waves. The EEG recorder employed in this measurement has 16 channels, and the values of these parameters corresponding to one channel, namely C3P3 (which is at the central part of the brain) are depicted in Table 2, however, such results have been calculated for all 16 channels. The average values of all these parameters have been computed, and a comparison with the ideal mode values of the same has been performed to analyze the effect of mobile phone radiations.

Table 2 EEG parameters values for C3P3 channel

Parameter	Mean		*p*-value
	Ideal mode	Tx mode	
Largest Lyapunov exponent (LLE)	0.009708	0.017508	0.07
Approximate entropy (ApEn)	0.619731	0.62055	0.968
Hurst exponent (HE)	0.67078	0.69472	0.33
Correlation dimension (CD)	2.081912	2.079019	0.893
Delta power	70.04701	74.89917	0.801
Theta power	18.31366	21.96054	0.325
Alpha power	111.6407	125.6605	0.031*
Beta power	5.340967	6.746587	0.208
Gamma power	0.318761	0.343972	0.477

* *p-values which are significant*

The visible measurable variations in the non-linear parameters and power values of the brain waves have been observed due to use of mobile phone. Relatively large variation has been noticed on the side of mobile phone, and also, impact on the other internal segments of the brain has been observed. The human emotions and features like visual processing, memory, and attention are largely dependent on this segment of brain. Also, the analysis of delta, theta, alpha, beta, and gamma waves of other channels has shown statistically significant variations in some cases.

4 Coherence Analysis ECG and EEG Signals

The coherence between the ECG and EEG signals has been investigated by examining the correspondence between the P, QRS, and T waves of the ECG signal and the five brain waves, i.e., delta, theta, alpha, beta, and gamma waves of the EEG signal. The coherence of ideal ECG waves has been calculated with the ideal brain waves. Similarly, the coherence of ECG waves acquired during Tx mode has been calculated for Tx mode brain waves. The coherence coefficient has been calculated using the expression presented by Golinska [11]. The mean value has been considered, and then, the maximum, minimum, mean, and standard deviation values have been calculated for all channels of EEG. In Table 3, the coherence coefficients between P, QRS, and T waves of the ECG signal and the five brain waves for ideal and Tx modes are given corresponding to C3P3 channel. The coherence coefficient values show that there exists a trend in the change in the values from the ideal modes to

Table 3 Maximum coherence values at C3P3 channel

ECG parameter	EEG parameter	Ideal mode	Tx mode
P	Delta	0.2042	0.2091
	Theta	0.1903	0.1778
	Alpha	0.2138	0.2166
	Beta	0.2089	0.1993
	Gamma	0.2007	0.2244
QRS	Delta	0.2381	0.2421
	Theta	0.2606	0.2312
	Alpha	0.2278	0.2045
	Beta	0.2321	0.2350
	Gamma	0.2928	0.2297
T	Delta	0.2116	0.1968
	Theta	0.2037	0.2474
	Alpha	0.2006	0.2336
	Beta	0.2128	0.2294
	Gamma	0.2400	0.2207

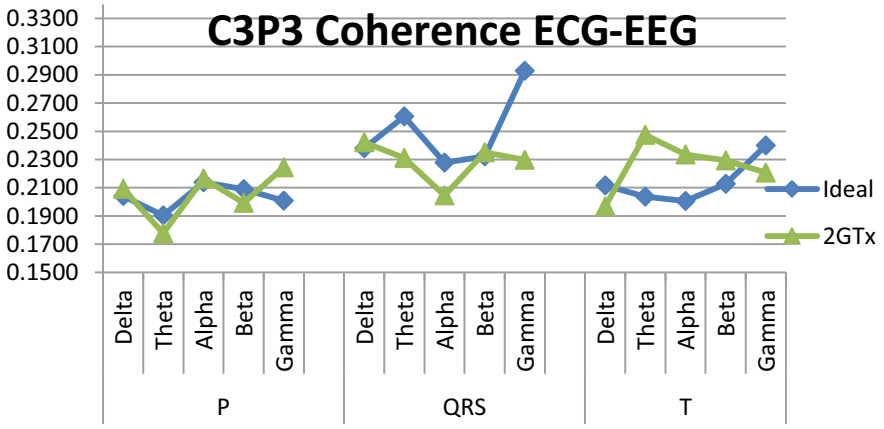


Fig. 1 Coherence plot at C3P3 channel

the communication mode under the influence of the mobile phone radiations. The presented preliminary study has been undertaken under a normal lab setup for short duration only, so the coherence coefficient values are up to 0.3, however, it is expected that if the similar experiment is performed under clinically controlled environment, then better coherence coefficient values shall be observed. As evident from Table 3, at C3P3 channel, the P wave has highest coherence value with gamma wave during Tx mode, so the gamma wave can be controlled with the variation in P wave. QRS complex is having highest similarity with the delta wave in the Tx mode, and the T wave is mainly influencing the theta wave during Tx mode. Similarly, during ideal mode, alpha, theta, and gamma waves are synchronized with the P wave, the QRS complex, and T wave, respectively. So, it shows that during the ideal as well as Tx modes, notable synchronization is detected between ECG and EEG signals. Figure 1 shows the coherence plot at C3P3 channel.

5 Design of ANN Ensemble Model to Predict the Impact of Mobile Phone Radiation on EEG from the ECG Signal

As discussed above, there exists coherence between the mobile phone radiations impacted ECG and EEG signals. So, ECG signal can be mapped to EEG signal, i.e., it is possible to predict the EEG values if we have the ECG signal and an efficient predictor. As the mapping between ECG and EEG is highly complex, so ANN ensemble approach has been proposed for modeling of this mapping. ANN ensemble approach performance in terms of generalization has been reported as superior [12], and it is suitable for many complex applications where single ANN models are not appropriate.

ANNs are the networks used to learn and train a particular task and then updating its weight according to some learning rule. These are based on trial-and-error method-based iterations [13], and the learning depends on the weights initialization and leaning rate. There is also a problem of getting stuck into local minima. These local minima problems have been solved by other optimization algorithms like PSO, etc., or by using the hybrid approaches. Making use of multiple possible neural network models with different specifications in parallel and then combining their output by a suitable method [14] is called an ANN ensemble technique. In ANN ensembles, the performance is better as many different ANN's outputs are considered to get the best result. Ensembling of ANN takes two important factors into consideration. First is the efficient training of the ANN models in the ensemble (generally known as constituent ANN models), and second is the method of combining the outputs of constituent ANN models.

Various steps involved in the design of ANN ensemble are given by Yang and Browne [15]. First step is to select the subsets from the original training data so that it involves diversification in it. The popular methods for making subsets are bagging, noise rejection, cross-validation, stacking, boosting, etc. The next step is to create different constituent ANN models, i.e., different neural networks. These constituent ANN model members should be diverse and must be initialized at different weights, different hidden layers, different structure, and different training algorithm to add to the diversity. The last step is the combination of the outputs from the ensemble members. It can be either assigning weight to each output or averaging the output or through voting or ranking, etc.

ANN ensemble model, as shown in Fig. 2, has been designed to predict the power of beta wave at FP2F4 channel from the low SDNN and LF power of ECG signal during Tx mode. The proposed ANN ensemble has four members, each being the multi-layer perceptron neural network (MLPNN) model having two hidden layers. The numbers of neurons in first and second hidden layers are 25 and 30, 15 and 25, 20 and 25, and 20 and 20 in MLPNN1, MLPNN2, MLPNN3, and MLPNN4 models, respectively. The tan-sigmoid transfer function has been used in all hidden layers and the output layers employed linear activation functions. The algorithm for the

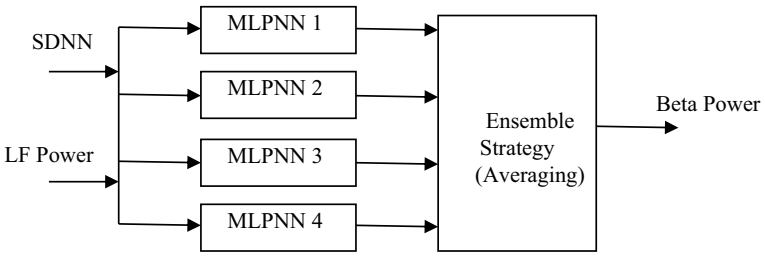


Fig. 2 ANN ensemble model to predict EEG beta power from ECG LF power and SDNN

training and testing of the ANN ensemble model is as follows.

For i=1:4
Apply the inputs
Set the target
Set the parameters
Define the activation functions
Set the learning rate
Set the error and the number of epochs
Compare the output and find the errorout
Adjust the weights
If errorout < error or epochs reached
Calculate output
Stop
end
Ensemble output
ensemble_error = (error1 + error2 + error3 + error4)/4
Average_ensemble_error = (sum(ensemble_error))/4

The constituent ANN models have been trained using the experimental values of mobile phone radiation impacted ECG and EEG values. A very small average ensemble error of 0.3298 has been achieved when the test data set has been applied to the trained ensemble. So, a new approach to predict the EEG parameters from the mobile phone radiations impacted ECG signals has been proposed. As comparable study could not be found in literature, so the results could not be compared with any other published results. These preliminary results show that it is possible to map the impact of mobile phone radiations on EEG parameters from the impact on parameters of ECG, and this approach will help in predicting the adverse effect of these radiations on the brain of the user without the need to measure the EEG. The ECG measurement is relatively easy as compared to EEG measurement, so the proposed method will help the researchers to easily predict the brain (EEG) signals from the heart (ECG) signals.

6 Conclusion

The experimental investigation of impact of radiations of mobile phones on ECG and EEG signals is an important research area, and the fact that the young generation uses mobile phones for longer durations makes this field more critical. A preliminary study to analyze the effects of radiations of mobile phones on ECG and EEG signals has been undertaken by comparing the variations in the values of various parameters with the ideal mode values of those parameters. With short duration (5 min) exposure, visible measurable changes have been noticed in all the ECG and EEG waves. Also, coherence between the ECG and EEG signals has been analyzed, and it has been found that there exist trends between the ECG and EEG signals with and without

mobile phone radiations impact. This coherence has been taken as the basis to design the ANN ensemble approach for mapping of ECG and EEG for impact analysis of mobile phone radiations. The results of this approach indicate that it is possible to map the impact of mobile phone radiations on EEG parameters from the impact on parameters of ECG, and this approach will help in predicting the adverse effect of these radiations on the brain of the user without the need to measure the EEG. Keeping in mind the fact that the people use mobile phones for longer durations, it is proposed that there is need of clinical supported research to quantify the impact.

References

1. Wessapan T, Rattanadecho P (2016) Flow and heat transfer in biological tissue due to electromagnetic near-field exposure effects. Int J Heat Mass Transfer 97:174–184
2. Rad SA, Jaleeli KA, Ahmad A (2015) Influence of electromagnetic radiation produced by mobile phone on viscosity of human blood. Int J Sci Environ Technol 4(6):1469–1475
3. Lu Y, Huang Y (2012) Biological effects of mobile phone radiations. In: International conference on microwave and millimeter wave technology (ICMMT), vol 2, pp 1–4
4. Lin JC (2010) Acquired cognitive behavior changes in children from cell phone use-health effects. IEEE Microw Mag 11(4):112–1144
5. Mat DAA, Kho F, Joseph A, Kipli K, Sahrani S, Lias K, Marzuki ASW (2010) Electromagnetic radiation from mobile phone near ear-skull region. In: Conference proceedings computer communication engineering, pp 1–5
6. Kaur B, Singh S, Kumar J (2015) A study of SAR pattern in biological tissues due to RF exposure. In: Conference proceedings on recent advances in engineering & computational sciences, pp 1–5
7. Repacholi MH (2001) Health risks from the use of mobile phones. Toxicology Lett 120:323–331
8. Guillet A, Arneodo A (2021) Tracking rhythms coherence from polysomnographic records: a time-frequency approach. Front Appl Math Stat. 7:624456
9. Pattnaik S, Dhaliwal BS, Pattnaik SS (2019) Impact analysis of mobile phone electromagnetic radiations on human electroencephalogram. Sadhana 44(134):1–12
10. Komeili G, Sarabandi SN (2012) Studying the effects of mobile phone waves on electro cardiogram parameters of students in Zahedan university of medical sciences. Int J High Risk Behav Addict. 1(2):75–78
11. Golinska AK (2011) Coherence function in biomedical signal processing: a short review of applications in neurology, cardiology and gynecology. Stud Log Grammar Rhetoric. 25(38):73–81
12. Yang J, Zeng X, Xhong S, Wu S (2013) Effective neural network ensemble approach for improving generalization performance. IEEE Trans Neural Netw Learn Syst 24(6):878–887
13. Zhang QJ, Gupta KC, Devabhaktuni VK (2003) Artificial neural networks for and microwave design - from theory to practice. IEEE Trans Microw Theory Tech 51(4):1339–1350
14. Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001
15. Yang S, Browne A (2004) Neural network ensembles: combining multiple models for enhanced performance using a multistage approach. Expert Syst 21(5):279–288

Chapter 33
Sharma-Mittal Entropy and Whale Optimization Algorithm Based Multilevel Thresholding Approach for Image Segmentation

Lovepreet Kaur, Baljit Singh Khehra, and Arjan Singh

1 Introduction

Segmentation has a significant role in field of image processing. It is one of the major preprocessing steps to enhance the quality of an image. Thresholding is one of the most important techniques for image segmentation due to its simplicity, accuracy, robustness, and effectiveness [1]. Thresholding has been widely applied for image segmentation in the field of satellite image processing [2, 3], agricultural image processing [4, 5], and medical image processing [6, 7]. The main target of image segmentation is to divide an image into a set of non-overlapping regions which are meaningful for a particular task. Many image processing applications required homogeneous characteristics of various regions of it to find out whether these regions are part of same object in interest.

There are two types of image thresholding: bi-level and multilevel. In bi-level, the image of interest is separated into two dissimilar regions. Pixels having gray values larger than a certain threshold value (T) are classified as object region, and rest of all gray values are classified as background region. On the other hand, in multilevel thresholding, image is divided into more than two separated regions. The idea is to find two or more threshold values for an image of interest so that it can be divided into various regions of brightness. From these various regions of brightness, one represents the background, and others are used to represent different objects in the image. Multilevel thresholding technique is quite useful in case of objects having complex backgrounds [8].

L. Kaur · A. Singh (✉)
Punjabi University, Patiala-147002, Punjab, India
e-mail: arjanpu@gmail.com

B. Singh Khehra
BAM KhalsaCollege, Garhshankar-144527, Hoshiarpur, Punjab, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_34

2 Related Work

Over the years, a lot of work has been done by different researchers in the field of image segmentation for automatic selection of suitable threshold value in order to segment an image. In the past forty years, entropy-based thresholding techniques have been explored by many scholars and researcher for the selection of suitable threshold value to segment different category of images. Otsu between class variance, Kapur's entropy, and minimum cross entropy are the three most popular techniques for automatic selection of threshold values to segment an image. In 1979, Ostu [9] proposed an image thresholding approach based on gray level intensities variance of regions of image to obtain suitable threshold value. In 1985, Kapur et al. [10] proposed an image thresholding approach based on the histogram of gray level values of image to segment the image into different segments by obtaining suitable threshold value. In 1985, Tsai [11] proposed a new approach to select automatic threshold. This approach is based on the moment-preserving principle. In 1988, Sahoo et al. [12] attempted to evaluate the performance of global thresholding methods like point dependent methods, histogram concavity analysis technique, entropic methods, etc. Originally, these techniques are developed for bi-level image thresholding, but afterward, these techniques have also been extended for multilevel image thresholding. But all these techniques are suffering from a major drawback, i.e., computational complexity. The computation complexity of these multilevel techniques increases sharply as the level of thresholding increases. The selection of optimal thresholds in multilevel thresholding is NP-hard problem. To overcome this drawback, number of meta-heuristic techniques has been applied by various researchers for multilevel image thresholding during last two decades.

Some of these meta-heuristic techniques are genetic algorithm (GA) [13], particle swarm optimization (PSO) [13, 17, 32], honey bee mating optimization [33], ant colony optimization (ACO) [13], firefly optimization algorithm [23, 34], differential evolution (DE) [13, 35], bacterial foraging algorithm [14, 15], cuckoo search algorithm [16, 18, 36], artificial bee colony algorithm [17, 21, 22], wind driven optimization [18], kinetic theory optimization algorithm [19], quantum inspired meta-heuristics algorithm [20], gravitational search algorithm [24], social spider optimization [25], flower pollination algorithm [25], gray wolf optimizer [26–28], whale optimization algorithm [29], moth-flame optimization [29], adaptive wind driven optimization [30], moth swarm algorithm [31], teaching-learning-based optimization algorithm [37], and bat optimization [38].

In this paper, a new meta-heuristic technique, named WOA-based Sharma-Mittal entropy (WOA-based MSEMT), has been proposed for finding an optimal set of thresholds and utilizing such thresholds to effectively segment digital images. The proposed technique is the combination of whale optimization algorithm [39] and Sharma-Mittal entropy [40]. The contribution of this work is novel, as the concept of threshold selection has not been explored using Sharma-Mittal entropy and WOA. The methodology makes use of the concept of Sharma-Mittal entropy and WOA to

threshold digital images into multiple regions. The major contributions of this paper are summarized as follows:

- Sharma-Mittal entropy is used to design objective function for formulating multilevel thresholding problem.
- Whale optimization algorithm (WOA) is applied to minimize Sharma-Mittal entropy-based objective function for searching optimal threshold values up to 6^{th} level.

The performance of the proposed approach is evaluated qualitatively and quantitatively with the state-of-the-art multilevel thresholding approaches using benchmark images of standard datasets.

3 Problem Formulation Based on Sharma-Mittal Entropy

If $f(x, y)$ is an input image of size $M \times N$, then normalized histogram of the input image is defined as

$$p_i = \frac{\text{freq}_i}{M \times N}, i = 0, 1, 2.....L_{\max} \tag{1}$$

where freq_i is the frequency of occurrence of gray level value i in the input image; $L_{\max}$ is the maximum gray level value in the input image, and p_i is the probability of occurrence of gray level value i in the input image.

When image is segmented by k-level thresholding, image is segmented into $k + 1$ segments $(S_1, S_2, ..., S_j, ..., S_k, S_{k+1})$ through k threshold values denoted as $T_1, T_2, T_3, ..., T_j, ..., T_{k-1}, T_k \in [0, L_{\max}]$.

Sharma-Mittal entropy is defined as

$$\text{SME} = \frac{1}{1-\alpha}\left[\left(\sum_{i=1}^{n} p_i^{\beta}\right)^{\frac{1-\alpha}{1-\beta}} - 1\right] \tag{2}$$

$$\begin{aligned}\text{Of}_{\text{SME}}(T_1, T_2, T_3, ..., T_k) =& \text{SME}_{S_1} + \text{SME}_{S_2} + ... + \text{SME}_{S_j} \\ &+ .... + \text{SME}_{S_k} + \text{SME}_{S_{k+1}}\end{aligned} \tag{3}$$

$$\text{SME}_{S_j} = \frac{1}{1-\alpha}\left[\left\{\sum_{i=T_{j-1}+1}^{T_j}\left(\frac{p_i}{\sum_{i=T_{j-1}+1}^{T_j} p_i}\right)^{\beta}\right\}^{\frac{1-\alpha}{1-\beta}} - 1\right] \tag{4}$$

$$\mathrm{SME}_{S_j} = \frac{1}{1-\alpha}\left[\left\{\sum_{i=T_{j-1}+1}^{T_j}\left(\frac{p_i}{\psi_J}\right)^{\beta}\right\}^{\frac{1-\alpha}{1-\beta}} - 1\right] \tag{5}$$

$$\psi_j = \psi(T_{j-1}+1, T_j) = \sum_{i=T_{j-1}+1}^{T_j} p_i \tag{6}$$

$$j = 1, 2, ..., k, k+1$$

$$T_0 + 1 = 0 \text{ and } T_{k+1} = L_{\max}$$

$$\mathrm{Of}_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k) = \sum_{j=1}^{k+1} \mathrm{SME}_{S_j} \tag{7}$$

$$\mathrm{Of}_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k) = \frac{1}{1-\alpha}\sum_{j=1}^{k+1}\left[\left\{\sum_{i=T_{j-1}+1}^{T_j}\left(\frac{p_i}{\psi_j}\right)^{\beta}\right\}^{\frac{1-\alpha}{1-\beta}}\right] - \frac{k+1}{1-\alpha} \tag{8}$$

$$\mathrm{Of}_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k) = \frac{1}{1-\alpha}\sum_{j=1}^{k+1}\left[\left\{\frac{\sum_{i=T_{j-1}+1}^{T_j} p_i^{\beta}}{\psi_j^{\beta}}\right\}^{\frac{1-\alpha}{1-\beta}}\right] - \frac{k+1}{1-\alpha} \tag{9}$$

Major aim is to determine the values of $T_1, T_2, T_3, ..., T_k$ from 0 to $L_{\max}$ such that Eq. (9) can be maximized. These determined k-level thresholding values of $T_1, T_2, T_3, ..., T_k$ are called optimal threshold values denoted as $T_1^*, T_2^*, T_3^*, ..., T_{k-1}^*$.

$$T_1^*, T_2^*, T_3^*, ..., T_k^* = \underset{T_1, T_2, T_3, ..., T_k = 0}{\overset{L_{\max}}{\arg\max}} [\mathrm{Of}_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k)] \tag{10}$$

The above maximization optimization problem can be written as minimization optimization problem using the following equation:

$$T_1^*, T_2^*, T_3^*, ..., T_k^* = \underset{T_1, T_2, T_3, ..., T_k = 0}{\overset{L_{\max}}{\arg\min}}\left[\frac{1}{1 + \mathrm{Of}_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k)}\right] \tag{11}$$

$$\mathrm{Of}'_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k) = \underset{T_1, T_2, T_3, ..., T_k = 0}{\overset{L_{\max}}{\arg\min}}\left[\frac{1}{1 + \mathrm{Of}_{\mathrm{SME}}(T_1, T_2, T_3, ..., T_k)}\right] \tag{12}$$

Subject to constraint is

$$0 < T_1^* < T_2^* < T_3^* < \ldots\ldots\ldots < T_{k-1}^* < L_{\max}$$

4 Proposed Approach

4.1 WOA-Based Sharma-Mittal Entropy Multilevel Thresholding Algorithm

The paper uses the WOA-based Sharma-Mittal entropy multilevel thresholding approach for finding an optimal set of thresholds and utilizing such thresholds to effectively segment digital images. The contribution of this work is novel, as the concept of threshold selection has not been explored using Sharma-Mittal entropy and WOA. The methodology makes use of the concept of Sharma-Mittal entropy and WOA to threshold digital images into multiple regions. The proposed methodology is described as a set of steps that are given below:

Step 1: Initialization.

Step 1.1: Initialize WOA parameters.

- Whales population size (number of search agents): N_{sa}
- Maximum number of iterations: $I_{\max}$
- Number of design variables (T_1, T_2, ..., T_k); k
- Dimension of search space: 0 to $L_{\max}$
- Limits of design variables:$0 \leq T_1, T_2, \ldots\ldots, T_k \leq L_{\max}$, in case of gray level image $L_{\max} = 255$
- Iteration counter, $I_c = 0$

Step 1.2: Initialize Whales Population.

Population is a set of some candidate solutions. Generate random population according to the population size and the number of design variables.

$$\text{Pop} = X[N_{\text{sa}}, k]$$

Population is represented by a matrix of size $N_{\text{sa}} \times k$ as follows:

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \ldots\ldots X_{1k} \\ X_{21} & X_{22} & X_{23} \ldots\ldots X_{2k} \\ . & . & . \\ . & . & . \\ X_{N_{\text{sa}}k} & X_{N_{\text{sa}}k} & X_{N_{\text{sa}}k} \ldots\ldots X_{N_{\text{sa}}k} \end{bmatrix}$$

$$\{X_{i1}\} \in T_1, \{X_{i2}\} \in T_2, \{X_{i3}\} \in T_3, \ldots\ldots, \{X_{ik}\} \in T_k$$

$$i = 1, 2, 3, ..., N_{\text{sa}}$$

Population is a random set of individuals. Random set of individuals may or may not be feasible solutions. So, it is required that randomly generated population satisfies the constraint: $0 \leq X_{i1} \leq X_{i2} \leq X_{i3} \leq ..... \leq X_{ik} \leq L_{\max}$.

Step 2: Fitness Evaluation.

Evaluate fitness of feasible solutions in the population and arrange these solutions according to their fitness values. For evaluating the fitness of feasible solutions in the population, objective function (*) is used. The value of a particular objective function for each feasible solution candidate as its fitness is denoted as

$$F_i = Of'_{SME}(X_i)$$

$$i = 1, 2, 3, ..., N_{sa}$$

Arrange feasible solution candidates of the population in ascending order according to their respective fitness value.

Step 3: Select best candidate solution (search agent)

$$X_\alpha = X_1 = X_{Of'_{SME}(X_i)=\min}$$

Step 4: Calculate E_1 and E_2.

The values of E_1 are linearly decreasing from 2 to 0 over the course of iterations

$$E_1 = 2 - I_c * (2/I_{\max})$$

The values of E_2 are linearly decreasing from -1 to -2 over the course of iterations

$$E_2 = -1 + I_c * (-1/I_{\max})$$

Step 5: For each candidate solution (search or hunt agent), i.e., $for\ i = 1 : N_{sa}\ do.$

Step 6: Declare Random Variables.

$r_1 = rand\ ()$; where r_1 is a random number in [0, 1].
$r_2 = rand\ ()$; where r_2 is a random number in [0, 1].
$r_3 = rand\ ()$; where r_3 is a random number in [0, 1].
$p = rand\ ()$; where p is a random number in [0, 1].
$r_4 = randi\ (N_{sa})$; where r_4 is an integer random number in [1, 2, 3 ,……, N_{sa}].
$r_5 = randi\ (k)$; where $r_5$is an integer random number in [1, 2, 3 ,……., k].

Step 7: Update the position of each search agent according to the position of the prey.

For each design variable in candidate solution, i.e., $for\ \ j = 1 : k\, do$

$A = 2 * E_1 * r_1 - E_1$

$C = 2 * r_2$

$g = 1$

$l = (E_2 - 1) * r_3 + 1$

$if\ \ p < 0.5$

$if\ \ |A| < 1$

$D = \left|C * X_\alpha(j) - X(i, j)\right|$

$X^{New}(i, j) = X_\alpha(j) - A * D$

$else$

$D = |C * X(r_4, r_5) - X(i, j)|$

$X^{New}(i, j) = X(r_4, r_5) - A * D$

$endif$

$else$

$D = \left|X_\alpha(j) - X(i, j)\right|$

$X^{New}(i, j) = D * e^{gl} * \cos(2\pi l) + X_\alpha(j)$

$endif$

Step 8: e*nd for* of Step 7.

Step 9: New feasible solution inserted in new population

$if\ \ 0 \le X_{i1}^{New} \le X_{i2}^{New} \le X_{i3}^{New} \le \ldots\ldots\ldots \le X_{ij}^{New} \le L_{\max}$

$X_i = X_i^{New}$

$else$

$X_i = X_i$

$endif$

Step 10: e*nd for* of Step 5.

Step 11: Increment in iteration

$$I_c = I_c + 1$$

Step 12: Go to Step 2 until the stopping criteria (maximum iteration: I_{max}) are not met, i.e.,

$$Go\ to\ \ Step2\ \ until\ \ \ I_c < I_{\max}$$

Step 13: Optimal threshold Values

$$T_1 = X_{11}, T_2 = X_{12}, T_3 = X_{13}, \ldots\ldots\ldots, T_k = X_{1k}$$

Step 14: Stop

4.2 Procedure for Calculating Objective Function Based on Sharma-Mittal Entropy

The paper uses Sharma-Mittal entropy as objective function for evaluating the fitness values of feasible solutions in the population. The complete steps for evaluating the fitness value of each candidate solutions in the population using Sharma-Mittal entropy are given below:

Step 1: Read an image

$f(x, y)$; f is the gray level intensity at a particular coordinator point (x, y).

Step 2: Find size of image

$$[M \times N]$$

Step 3: Find the probability distribution of the gray level values of the image

$$p_i = \frac{\text{freq}_i}{M \times N}$$

where freq_i is the frequency of occurrence of a particular gray level value (intensity) i in the input image f of size $M \times N$; $i = 0, 1, 2, \ldots\ldots\ldots\ldots\ldots, L_{\max}$.

Step 4: Extract threshold values from feasible solution of the population

$$T_0 = 0; T_1 = X_{i1}; T_2 = X_{i2}; T_3 = X_{i3}; \ldots\ldots\ldots; T_{k-1} = X_{ik-1}; T_k = L_{\max}$$

Step 5: Declare alpha (α), beta (β) parameters and calculate another parameter from alpha (α) and beta (β)

$$\gamma = \frac{1-\alpha}{1-\beta}$$

Step 6: Evaluate the fitness of feasible solution

$sum = 0$
$forj = 1tok$
$fn\beta = call\ \psi\beta\,(T_{j-1}, T_j)$
$fn = call\ \psi\,(T_{j-1}, T_j)$

$sum = sum + \left(\frac{fn\beta}{fn}\right)^{\gamma}$
$endfor$
$Of_{SME} = \frac{1}{1-\alpha}[sum - k - 1]]$
$Of'_{SME} = \frac{1}{1+Of_{SME}}$
$F_i = Of'_{SME}$

Step 7: Procedure for evaluating $\psi\beta(a, b) and \psi(a, b)$

$sum0 = 0$
$sum1 = 0$
$for\ l = a\ to\ b$
$sum0 = sum0 + p_l$
$sum1 = sum1 + p_l^{\beta}$
$endfor$
$\psi = sum0$
$\psi\beta = sum1$
$return\ \psi\beta$
$return\ \psi$

Step 8: Stop.

5 Experimental Results

For implementing the proposed WOA-based SMEMT algorithm, MATLAB R2019 is used on Intel(R) Core(TM) i7-8550U CPU@1.80 GHz machine. Four standard test images are taken for evaluating the performance of the proposed approach. Figure 1 illustrates the original images and their respective histograms.

Images are segmented by the proposed approach at 3rd, 4th, 5th, and 6th level thresholding. Experiment results are compared with GWO, TLBO, and GA. The segmented images using the proposed approach, GWO, TLBO, and GA at 6-level thresholding are illustrated in Fig. 2.The performance characteristics of WOA-based SMEMT, GWO-based SMEMT, TLBO-based SMEMT, and GA-based SMEMT algorithms at 6-level thresholding for standard test images are shown in Fig. 3. From visual point of view, it is observed that the performance of the proposed approach for segmentation of images is satisfactory. Moreover, the convergence rate of WOA-based SMEMT approach is better than other approaches.

For quantitative analysis, PSNR, uniformity, SSIM, and MSSIM parameters are used to evaluate the performance of the proposed approach with other methods. Higher values of these parameters point toward better segmented image. Threshold values which are achieved through GWO-based MCET as well all other methods for all the test images at various levels of thresholding are shown in Table 1.

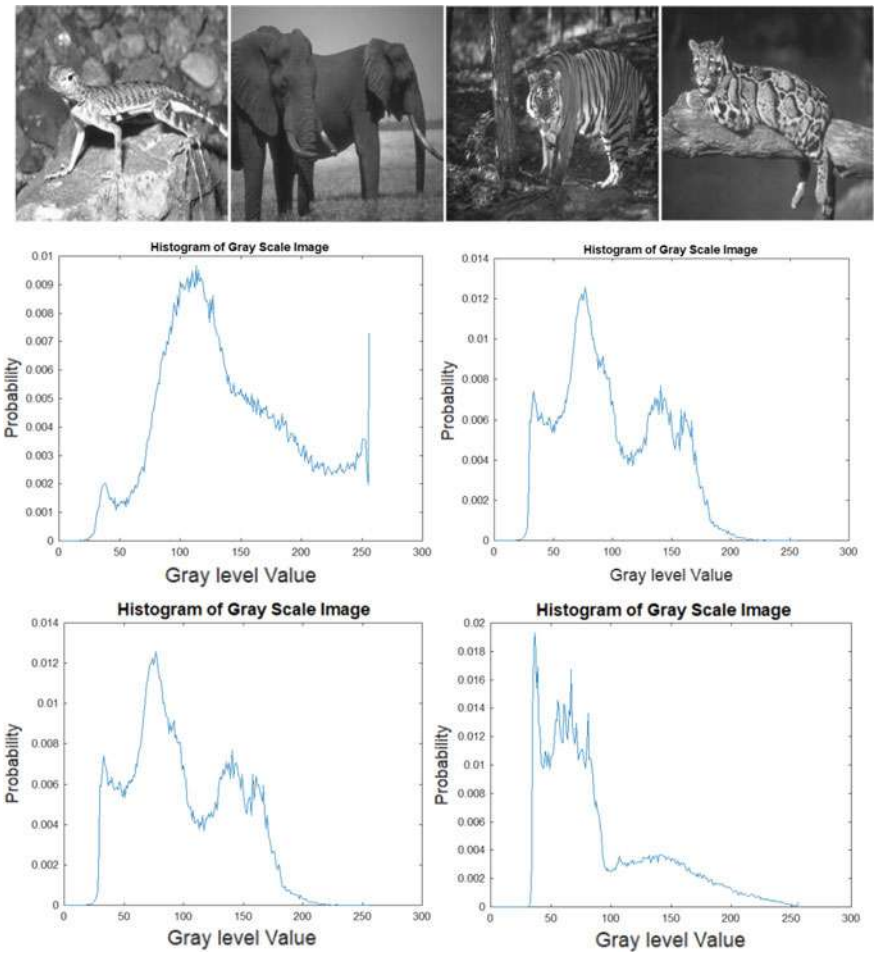


Fig. 1 Original images and their respective histograms

PSNR, uniformity, SSIM, and MSSIM of WOA-based SMEMT algorithm, GWO-based SMEMT algorithm, TLBO-based SMEMT algorithm, and GA-based SMEMT algorithm for standard test images are shown in Tables 2, 3, 4, and 5, respectively.

From Tables 2, 3, 4, and 5, it is clearly evident that the proposed WOA-based SMEMT approach is performing better than all other approaches in term of achieving higher PSNR, uniformity, SSIM, and MSSIM as level of threshold increases. Above mentioned results demonstrate that the proposed approach has the capability of finding multilevel optimal threshold values by which an image can segmented in an effective way for further analysis.

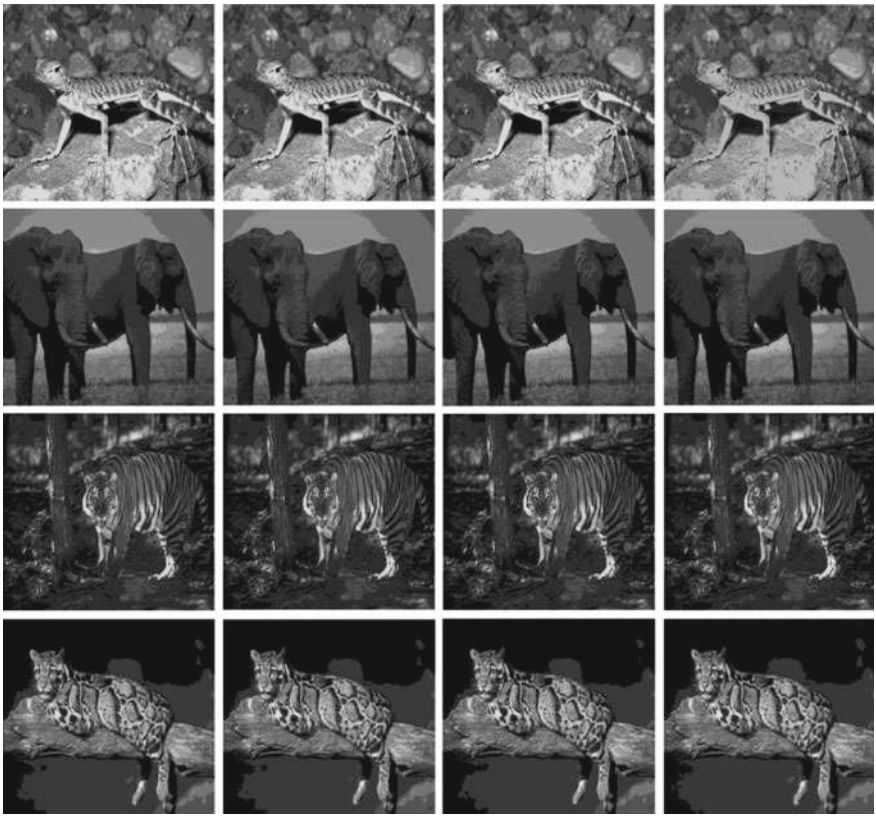


Fig. 2 Segmented images at $k = 6$ (thresholding at levels 6) by WOA-based SMEMT, GWO-based SMEMT, TLBO-based SMEMT, and GA-based SMEMT Methods

6 Conclusions and Future Scope

Whale optimization algorithm (WOA) and Sharma-Mittal entropy have been used in this paper to find best possible combination of threshold values for image segmentation at multiple levels. The new proposed approach is named as WOA-based SMEMT algorithm. To evaluate the performance of the proposed WOA-based SMEMT approach, the results are compared with GWO-based SMEMT, TLBO-based SMEMT, and GA-based SMEMT approaches. The performance is evaluated using a set of four different kinds of images that have variant characteristics. Threshold values are searched to segment test images at 3rd, 4th, 5th, and 6th levels. PSNR, uniformity, SSIM, and MSSIM are four measures that have been used to check the efficiency of the proposed approaches. From experimental results, it is concluded that the proposed WOA-based SMEMT approach has the more ability to produce quality segmented image as compared to GWO-based SMEMT, TLBO-based SMEMT, and GA-based SMEMT approaches. Therefore, WOA-based

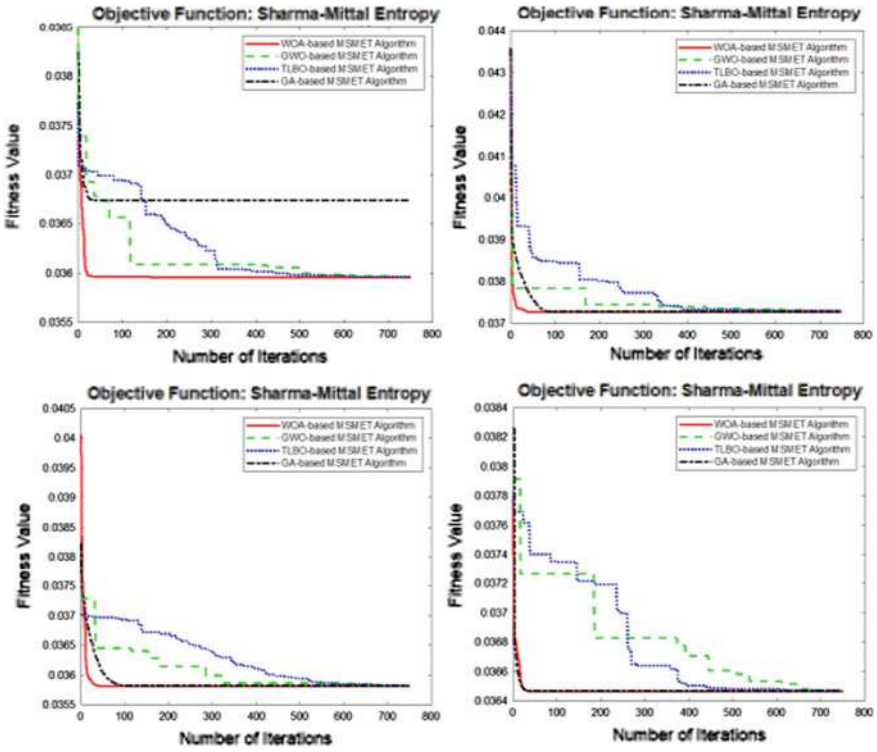


Fig. 3 Performance characteristics of WOA-based SMEMT, GWO-based SMEMT, TLBO-based SMEMT, and GA-based SMEMT algorithms at 6-level thresholding for standard test images

SMEMT approach has the capability to discover the best possible combination of threshold values for the segmentation of given images.

In future, the more advanced meta-heuristic techniques will be applied to further enhance performance of the proposed approach.

Table 1 Multilevel threshold (k) values of images obtained through proposed and other algorithms

Image # (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
1 (321 × 481)	3	75,135,195	75,135,195	82,141,194	75,135,195
	4	69,116,163,208	69,116,162,208	70,116,160,208	69,116,162,208
	5	61,99,138,177,216	61,99,138,177, 216	58,92,130,171, 216	35,74,115,155, 196
	6	56, 89, 122, 155, 188, 221	54, 87, 121, 154, 187, 221	54, 88, 121, 154, 187, 221	35, 66, 96, 126, 156, 189
2 (321 × 481)	3	74, 127, 181	74, 127, 181	68, 129, 180	72, 125, 181
	4	62,103,145,188	61,102,145,188	77,107,138,182	61,101,141,184
	5	59, 96, 132, 170, 202	59, 96, 132, 169, 202	60, 98, 134, 170, 203	57, 93, 128, 162, 194
	6	53, 85, 117, 148, 179, 208	54, 86, 118, 149, 180, 208	57, 90, 121, 148, 180, 208	54, 86, 118, 149, 180, 208
3 (321 × 481)	3	80, 139, 196	80, 139, 196	80, 139, 196	80, 139, 196
	4	65, 111, 157, 206	64,111,157,206	65,108,156,202	64,110,155,202
	5	54, 94, 134, 173, 213	55, 94, 134, 173, 213	58, 95, 133, 174, 214	43, 74, 119, 163, 209
	6	47, 81, 115, 150, 185, 219	47, 81, 116, 150, 185, 219	49, 83, 117, 151, 186, 219	47, 81, 116, 150, 185, 219
4 (321 × 481)	3	89, 144, 199	89, 144, 199	91, 150, 198	87, 138, 197
	4	82, 125, 168, 211	82,126,169,211	75,126,166,213	82,125,168,211
	5	67, 104, 142, 180, 218	68, 105, 143, 180, 218	67, 103, 141, 180, 218	67, 104, 142, 180, 218
	6	61, 91, 125, 158, 191, 223	61, 92, 125, 158, 191, 223	62, 92, 127, 160, 192, 222	60, 91, 125, 158, 191, 223

Table 2 PSNR values of images obtained through proposed and other algorithms after segmentation

Image (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
1 (321 × 481)	3	16.6401	16.6401	16.4496	16.6401
	4	18.6213	18.6392	18.6766	18.6392
	5	20.2867	20.2867	20.3611	19.7108
	6	21.5766	21.5584	21.5794	20.5864

(continued)

Table 2 (continued)

Image (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
2 (321 × 481)	3	16.2839	16.2839	17.0289	16.5784
	4	18.6957	18.7564	16.3486	18.8994
	5	19.4540	19.4790	19.3345	19.9593
	6	20.7752	20.6779	20.3837	20.6779
3 (321 × 481)	3	14.4154	14.4154	14.4154	14.4154
	4	16.5530	16.6875	16.6038	16.7122
	5	18.8621	18.7093	18.1880	20.7534
	6	20.6476	20.6288	20.2878	20.6288
4 (321 × 481)	3	13.6392	13.6392	13.4878	13.8176
	4	14.5028	14.4972	15.3875	14.5028
	5	16.8051	16.6504	16.8074	16.8051
	6	18.0830	18.0823	17.9016	18.2734

Table 3 Uniformity values of images obtained through proposed and other algorithms after segmentation

Image (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
1 (321 × 481)	3	0.9709	0.9709	0.9722	0.9709
	4	0.9759	0.9761	0.9764	0.9761
	5	0.9795	0.9795	0.9784	0.9715
	6	0.9815	0.9811	0.9814	0.9706
2 (321 × 481)	3	0.9782	0.9782	0.9767	0.9787
	4	0.9832	0.9831	0.9801	0.9836
	5	0.9818	0.9820	0.9817	0.9841
	6	0.9853	0.9852	0.9848	0.9852
3 (321 × 481)	3	0.9778	0.9778	0.9778	0.9778
	4	0.9807	0.9805	0.9815	0.9809
	5	0.9821	0.9824	0.9829	0.9826
	6	0.9836	0.9834	0.9836	0.9834
4 (321 × 481)	3	0.9716	0.9716	0.9709	0.9725
	4	0.9717	0.9715	0.9742	0.9717
	5	0.9803	0.9799	0.9804	0.9803
	6	0.9824	0.9824	0.9822	0.9825

Table 4 SSIM values of images obtained through proposed and other algorithms after segmentation

Image (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
1 (321 × 481)	3	0.9097	0.9097	0.9022	0.9097
	4	0.9419	0.9421	0.9419	0.9421
	5	0.9606	0.9606	0.9609	0.9505
	6	0.9703	0.9705	0.9706	0.9516
2 (321 × 481)	3	0.8301	0.8301	0.8534	0.8396
	4	0.9015	0.9033	0.8311	0.9060
	5	0.9160	0.9165	0.9138	0.9248
	6	0.9382	0.9365	0.9308	0.9365
3 (321 × 481)	3	0.6698	0.6698	0.6698	0.6698
	4	0.7968	0.8025	0.7988	0.8034
	5	0.8763	0.8721	0.8569	0.9170
	6	0.9172	0.9169	0.9099	0.9169
4 (321 × 481)	3	0.6936	0.6936	0.6820	0.7065
	4	0.7528	0.7525	0.7977	0.7528
	5	0.8526	0.8475	0.8526	0.8526
	6	0.8887	0.8887	0.8842	0.8933

Table 5 MSSIM values of images obtained through proposed and other algorithms after segmentation

Image (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
1 (321 × 481)	3	0.6421	0.6421	0.6389	0.6421
	4	0.7321	0.7329	0.7346	0.7329
	5	0.7998	0.7998	0.8067	0.7736
	6	0.8392	0.8368	0.8389	0.8210
2 (321 × 481)	3	0.3716	0.3716	0.3935	0.3816
	4	0.4692	0.4727	0.3899	0.4794
	5	0.5056	0.5059	0.4989	0.5228
	6	0.5616	0.5564	0.5428	0.5564
3 (321 × 481)	3	0.4483	0.4483	0.4483	0.4483
	4	0.5842	0.5907	0.5889	0.5927
	5	0.6908	0.6856	0.6659	0.7793
	6	0.7664	0.7656	0.7524	0.7656
4 (321 × 481)	3	0.3514	0.3514	0.3446	0.3586
	4	0.3889	0.3885	0.4136	0.7528

(continued)

Table 5 (continued)

Image (size)	k	WOA-based SMEMT algorithm	GWO-based SMEMT algorithm	TLBO-based SMEMT algorithm	GA-based SMEMT algorithm
	5	0.4573	0.4540	0.4573	0.4573
	6	0.4955	0.4950	0.4901	0.4992

References

1. Dougherty G (2010) Digital image processing for medical applications. Cambridge University Press, New Delhi, India
2. Suresh S (2017) ShyamLal: multilevel thresholding based on chaotic Darwinian particle swarm optimization for segmentation of satellite images. Appl Soft Comput 55:503–522
3. Bhandari AK, Kumar A, Singh GK (2015) Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's Otsu and Tsallis functions. Expert Syst Appl 42(3):1573–1601
4. Arroyo J, Guijarro M, Pajares G (2016) An instance-based learning approach for thresholding in crop images under different outdoor conditions. Comput Electron Agric 127(2016):669–679
5. Ji W, Qian Z, Xu B, Tao Y, Zhao D, Ding S (2016) Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot. Optik, 127(23):11173–11182
6. Li Y, Bai X, Jiao L, Xue Y (2017) Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Appl Soft Comput 56:345–356
7. Sarkar S, Das S, Chaudhuri SS (2017) Multi-level thresholding with a decomposition-based multi-objective evolutionary algorithm for segmenting natural and medical images. Appl Soft Comput 50:142–157
8. Horng MH (2011) Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Syst Appl 38(11):13785–13791
9. Otsu N (1979) A threshold selection from gray level histograms. IEEE Trans Syst, Man Cybern 9(1):62–66
10. Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray level picture thresholding using the entropy of the histogram. Comput Vision, Graphics Image Process 29(3):273–285
11. Tsai W-H (1985) Moment-preserving thresolding: a new approach. Comput Vision, Graphics Image Process 29(3):377–393
12. Sahoo PK, Soltani S, Wong AKC, Cheng YC (1988) A survey of thresholding techniques. Comput Vision, Graphics Image Process 41(2):233–260
13. Hammouche K, Diaf M, Siarry P (2010) A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. Eng Appl Artif Intell 23(5):676–688
14. Sathya PD, Kayalvizhi R (2011) Optimal multilevel thresholding using bacterial foraging algorithm. Expert Syst Appl 38:15549–15564
15. Sathya PD, Kayalvizhi R (2011) Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images. Measurement 44:1828–1848
16. Agrawal S, Panda R, Bhuyan S, Panigrahi BK (2013) Tsallis entropy based optimal multilevelthresholding using cuckoo search algorithm. Swarm Evol Comput 11:16–30
17. Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13(6):3066–3091
18. Bhandari AK, Singh VK, Kumar A, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst Appl 41(7):3538–3560
19. Fan C, Ouyang H, Zhang Y, Xiao L (2014) Optimal multilevelthresholding using molecular kinetic theory optimization algorithm. Appl Math Comput 239:391–408

20. Saha SI, Bhattacharyy S, Maulik U (2014) Multi-level thresholding using quantum inspired meta-heuristics. Knowl Based Syst 67:373–400
21. Bhandari AK, Kumar A, Singh GK (2015) Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's Otsu and Tsallis functions. Expert Syst Appl 42:1573–1601
22. Bouaziz A, Draa A, Chikhi S (2015) Artificial bees for multilevel thresholding of iris images. Swarm Evol Comput 21:32–40
23. Erdmann H, Wachs-Lopes G, Gallao C, Ribeiro M, Rodrigues P (2015) A study of a firefly meta-heuristics for multithreshold image segmentation. In: Developments in medical image processing and computational vision,lecture notes in computational vision and biomechanics (LNCVB), vol 19. Springer, pp 279–295
24. Sun G, Zhang A, Yao Y, Wang Z (2016) A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding. Appl Soft Comput 46:703–730
25. Ouadfel S, Taleb-Ahmed A (2016) Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study. Expert Syst Appl 55:566–584
26. Khairuzzaman AK, Chaudhury S (2017) Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Syst Appl 86:64–76
27. Li L, Sun L, Guo J, Qi J, Xu B, Li S (2017) Modified discrete grey wolf optimizer algorithm for multilevel image thresholding. Comput Intell Neurosci 1–16
28. Li MQ, Xu LP, Xu N, Huang T, Yan B (2018) SAR image segmentation based on improved grey wolf optimization algorithm and fuzzy C-means. Math Problems Eng 1–11
29. Abd El Aziz M, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevelthresholding image segmentation. Expert Syst Appl 83:242–256
30. Kotte S, Pullakura RK, Injeti SK (2018) Optimal multilevel thresholding selection for brain MRI image segmentation based on adaptive wind driven optimization. Measurement 130:340–361
31. Bhandari AK, Rahul K (2019) A context sensitive Masi entropy for multilevel image segmentation using moth swarm algorithm. Infrared Phys Technol 98:132–154
32. Yin P-Y (2007) Multilevel minimum cross entropy threshold selection based on particle swarm optimization. Appl Math Comput 184(2):503–513
33. Horng MH (2010) Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Syst Appl 37(6):4580–4592
34. Horng MH, Liou RJ (2011) Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Syst Appl 38(12):14805–14811
35. Sarkar S, Das S, Chaudhuri SS A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution. Pattern Recogn Lett 54(1):27–35
36. Pare S, Kumar A, Bajaj V, Singh GK (2017) An efficient method for multilevel color image thresholding using cuckoo search algorithm based on minimum cross entropy. Appl Soft Comput 61:570–592
37. Gill HS, Khehra BS, Singh A, Kaur L (2019) Teaching-learning-based optimization algorithm to minimize cross entropy for selecting multilevel threshold values. Egypt Inform J 20(1):11–25
38. Chen S, Peng GH (2019) Multilevel minimum cross entropy threshold selection based on the improved Bat optimization. In: Proceeding of international conference on intelligent and interactive systems and applications (IISA 2018), 29–30 June 2018, China. In: Xhafa F, Patnaik S, Tavana M (eds) Advances in intelligent systems and computing (AISC), vol 885. Springer, pp 312–320
39. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
40. Sharma BD, Mittal DP (1975) New non-additive measures of entropy for discrete probability distributions. J Math Sci 10:28–40

Chapter 34
COVID-19 Fake News Detection by Improved Ant Lion Optimizer Metaheuristics

Miodrag Zivkovic, Nebojsa Bacanin, Jelena Arandjelovic, Stefan Stanojlovic, Andjela Rakic, and K. Venkatachalam

1 Introduction

Because this pandemic of COVID-19 is the most catastrophic medical issue the world has ever seen, social media became a main source of diverse informations. Fake news around nowadays are something we encounter on a daily basis and what actually confuses nation and makes it more important to have credible information on the issue. The most popular social networking sites have made it easier for anyone all over the world to connect.

Huge range of news and informations and its quick dissemination result that some of it is true and some is false. People are influenced by fake news and many are pushed in the wrong way. As a result, there is a pressing need to develop a method for detecting false news concerning the COVID-19 epidemic. It is not simple to compare what is fake and what is real among news since fake news are designed to expose incorrect information. Various classification methods have been employed in false news

M. Zivkovic · N. Bacanin (✉) · J. Arandjelovic · S. Stanojlovic · A. Rakic
Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
e-mail: nbacanin@singidunum.ac.rs

M. Zivkovic
e-mail: mzivkovic@singidunum.ac.rs

J. Arandjelovic
e-mail: jelena.arandjelovic.16@singimail.rs

S. Stanojlovic
e-mail: stefan.stanojlovic.16@singimail.rs

A. Rakic
e-mail: andjela.rakic.16@singimail.rs

K. Venkatachalam
Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore, India
e-mail: venkatachalam.k@christuniversity.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_35

detection research. There news may have serious societal consequences. Social media informations are used as learning inputs in existing approaches for detecting false news. Fake news represents a significant position in deceiving the public and disseminating incorrect information. To identify COVID-19 tweets, BERT imbedding and a shallow neural network were used by the researchers. One other research [1] classified false news on COVID-19 using 10 ML algorithms and seven feature extraction techniques. Besides this, the study [2] utilized four ML classifiers for detection false news on social media:logistic regression, decision trees, gradient boost, and vector machine support. In order of detection false informations about COVID-19, researchers employed convolutional convolutional neural network (CNNs), long short-term memory (LSTM), and bidirectional encoder representations from transformers (BERT). With the introduction of DL, a significant advancement in the field of text categorization has been already there and therefore in the classification of false news. Furthermore, the study [3] established a method for detecting false news established on combination of three transformer types (ALBERT, BERT and XLNET). The Constraint AI 2021 Fake News Detection dataset was used to train and assess the model.

Purpose of this research was determination of the most essential characteristics related to false news content using approach for evolutionary fake news detection.

Identification of false news relied on two major sources of data: websites for news and social medias. Koirala dataset is used for tests executing, which is based on a fake news website, and six datasets were created with several tokenized shapes—bag-of-words, binary, TF-IDF, and TF. From the other side, there are stemming approaches—Data 1, Data 2, …, Data 6.

1.1 Research Goals and Contributions

Main research goal can be summed up in the following two aspects:

- Development of the modified ALO metaheuristics, that overcomes the recognized flaws of the original algorithm.
- Application of the devised modified ALO algorithm related to FS issue on the case study of fake news data.

1.2 Structure of the Paper

First section in this paper describes covid pandemic in the world as a painful health alarm, as well as false information and news today. In Sect. 2, the feature selection is presented, as well as its procedures, while in Sect. 3, the ALO algorithm and the operations related to it. Section 4 represents the utilized model, while Sect. 5 provides the results of ALO and tests for the k-NN-ALO technique.

2 Theoretical Background and Literature Survey

2.1 Feature Selection

Feature selection method represents a stage which must be completed earlier than data exploration activities (like arrangement and grouping) can be carried out. The fundamental purpose of feature selection is to plan information by decreasing evenness in a dataset's value. By removing balanced boisterous components and keeping uppermost relevant dataset's features, FS reduces the value of dataset [4].

Characteristics that are highly collerated with one another symmetric characteristics and those that are poorly connected to the (unrelated) target class are among the noisy features. Create a smaller version of the data set, reducing the expense of the learning system as far as time and equipment assets, is the primary advantage of utilizing FS. FS further develops the learning system's exhibition, since it might alleviate the issue of overfitting [1]. Searching and assessment are two of the most important FS procedures.

Temporal complexity of the various search methods varies. To determine the best feature subsets, brute-force approaches explore all of the created feature subsets in order to end the search process in exponential time. Random feature subsets are generated and examined by metaheuristic search methods until the near-optimal feature subset is found. By incorporating new operators, innovative update methods, new encoding plans, new fitness capacities and new instatement, just as equal and multi-unbiased algorithms, a considerable lot of these algorithms have been improved to further develop the FS cycle.

The FS assessment procedure is carried out using either dataset characteristics (filters) or a studying algorithm. Stainers do not require any learning and they are quick techniques. Wrappers, on the other hand, provide more precise findings since the categorizer employed inside the FS particular strategy is also used during the external testing phase.

In the past years, several metaheuristics and stochastic methods have already been proposed with the purpose of improving the FS procedure with encouraging results, such as the with a firefly algorithm an better approach for initialization that was proposed in [5], several particle swarm optimization-based versions [3, 6], gray wolf optimizer approach [7], genetic algorithm [8], the salp swarm algorithm method [9], etc.

2.2 Swarm Intelligence

Nature inspired methods, more precisely the swarm intelligence subfamily, are one of the most famous and popular family of the stochastic metaheuristics algorithms. Algorithms belonging to this family have been used in diverse range of applications. Famous and successful exemplars that established firm grounds for this kind of

metaheuristics include particle swarm optimization (PSO) [10], artificial bee colony (ABC) [11], and firefly algorithm (FA) [12]. Later, vast number of options emerged that were established themselves as very powerful and efficient optimizers, including gray wolf optimizer (GWO) [13], Harris Hawks optimization (HHO) [14], whale optimization algorithm (WOA) [15], and monarch butterfly algorithm (MBA) [16].

Recently, metaheuristics inspired by nature, and swarm intelligence algorithms in particular, have frequently been used in tackling different NP-hard tasks belonging to the information technologies domain, such as global numerical optimization problem [17], wireless sensor networks challenges including node localization problem and prolonging network lifetime [18–20], task scheduling challenge for cloud-based systems [21, 22], optimizing artificial neural networks [23–28], predicting COVID-19 cases [29, 30], optimizing classifiers for MRI or histological slides [31, 32].

3 Proposed Method

The ALO algorithm's concept is originally presented in this section. The mathematical model and ALO technique are then thoroughly discussed.

3.1 Inspiration

Antlions (doodlebugs) are members of the Neuroptera order's Myrmeleontidae family (net-winged insects). Antlions have two life stages: larvae and adults. Larvae can live for up to three years in their natural environment (adulthood takes just 3–5 weeks). Antlions develop into adults by metamorphosing in a cocoon. They mostly hunt as larvae, while the adult stage is devoted to breeding. They were named after different hunting technique and prey preferences. An antlion larvae plant out a cone-shaped trench in the sand by moving in a circular beat and flinging sand with its enormous mouth. A variety of cone-shaped pits of various diameters can be found. Following trap excavation, the larval lies under the cone's bottom and waits for doodlebug (most ants) to become caught in the hole.

3.2 The ALO Algorithm's Operators

Antlions and emmets interact in the pitfall using the ALO algorithm. Because of this, ants are forced to roam throughout the search field, while antlions are allowed to hunt them and gain fitness by using traps. In order to show how ants travel in the wild, a random walk is used:

$$X(t) = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \ldots, \text{cumsum}(2r(t_n) - 1)] \tag{1}$$

In which cumsum calculates total amount, n is the max number of iterations, and t is the number of steps in a a stroll at random:

$$r(t) = \begin{Bmatrix} 1 \text{ if rand} > 0.5 \\ 0 \text{ if rand} \leqslant 0.5 \end{Bmatrix} \tag{2}$$

t denotes the random walk step (in this case, iteration), rand is an accidental integer produced with uniform distribution which an be between 0 and 1.

3.3 Trapping in Antlion's Pits

Antlion traps influence random ant walks, as previously mentioned. The following equations are presented to illustrate this conjecture quantitatively:

$$c_i^t = \text{Antilion}_j^t + c^t \tag{3}$$

$$d_i^t = \text{Antilion}_j^t + d^t \tag{4}$$

where c^t is the vector containing the maximum of every variable at the t-th iteration, d^t is the vector containing the minimum of all variables at the t-th iteration, c_j^t is the minimum of every variable for the i-th ant, d_j^t is the maximum of every variable for the i-th ant, and Antliont j is the position of the chosen j-th ant.

3.4 Elitism

Evolutionary algorithms include an elitism feature which enables them to keep the finest solution that is found at any point throughout the process of optimization. The finest antlion acquired so far in each iteration is preserved and regarded an elite in this research. The elite should be able to affect the activities of all the ants during processes. As a result, it is believed that each ant travels around a randomly picked antlion by the roulette wheel and the elite at the same time:

$$\text{Ant}_i^t = \frac{R_A^t + R_E^t}{2} \tag{5}$$

R_A^t shows t-th iteration's random walk around the antlion, while R_E^t represents the t-th iteration's random walk around the elite, and AnttI represents the t-th iteration's position of the i-th ant.

3.5 ALO Algorithm

The ALO optimization method may now be defined using the suggested operators from the previous subsections. The ALO method is a triple function for solving optimization problems that approximates the global optimum using the following formula:

$$\mathrm{ALO}(A, B, C) \tag{6}$$

Function A creates a set of randomly generated beginning results, "B" changes the starting population of "A", and as soon as the end condition is met, C returns true. The definitions of the above functions are as follows:

$$\emptyset \xrightarrow{A} \left\{ M_{\mathrm{Ant}}, M_{\mathrm{OA}}, M_{\mathrm{Antilion}}, M_{\mathrm{OAL}} \right\} \tag{7}$$

$$\left\{ M_{\mathrm{Ant}}, M_{\mathrm{Antilion}} \right\} \xrightarrow{B} \left\{ M_{\mathrm{Ant}}, M_{\mathrm{Antilion}} \right\} \tag{8}$$

$$\left\{ M_{\mathrm{Ant}}, M_{\mathrm{Antilion}} \right\} \xrightarrow{C} \left\{ \mathrm{true}, \mathrm{false} \right\} \tag{9}$$

where M_{Ant} is the ant's position matrix, M_{Antlion} is the antlion's position matrix, M_{OA} is the ant's fitness matrix, and M_{OAL} is the antlion's fitness matrix [33]. The basic ALO pseudo-code is given by Algorithm 1.

Algorithm 1 Pseudo-code for the basic ALO metaheuristics

Randomly populate the initial colony of antilion and ants
Compute antlions' and ants' fitness levels
Discover the perfect antlions and consider yourself to be among the top
while the final criteria isn't met
for each ant
Using the Roulette wheel, choose an antlion
C and D should be updated
Create and normalize a random walk
Change the ant's positions
end for
Determine the fitness of each ant
If an antlion becomes fitter, replace it with its matching ant
If an antlion gets more fit than the elite, the elite should be updated
end while
Return elite

Because of the following arguments, suggested ALO approach could potentially solve optimization problems to their global optimums:

- The antlion's random selection and the ants' erratic movements around them ensure that the search space is explored.

- Random walks and the roulette wheel that are also used have a high likelihood of tackling the issue of lingering in the local optimum.
- The avoidance of local optima is included into ALO.
- The search domain is explored due to the random choosing of antlions and movement of ants around them.
- Over the course of rounds, intensity of ants' movements is adapted and reduced, ensuring that the ALO algorithm converges.
- During optimization, antlions migrate to the finest ant's position, saving encouraging portions of searching areas.
- Antlions point ants in the right direction in the search space.
- Increasing population diversity by calculating random paths for each ant and dimension.
- There are just a few settings to tweak using the ALO algorithm.
- Each iteration's finest antlion is kept in comparison to the finest antlion produced thus far.
- The antilion optimization method is a gradient-free method that treats the issue as if it were a black box.

3.6 Improved Antilion Optimizer

The basic ALO metaheuristics has established itself as a powerful optimizer, however, extensive simulations on the standard CEC benchmarks revealed that it can further be enhanced by improving both exploitation and exploration. The basic ALO can get stuck in sub-optimal areas of the search domain. Additionally, algorithm can suffer from slow convergence speed.

To handle the above mentioned drawbacks of the basic ALO, the proposed mALO method incorporates the chaotic initialization of the population, that was first proposed in [34]. The goal is to replace the random generation of population by chaotic sequences, that can make the search process more efficient, as shown in numerous practical implementations [35, 36]. A large number of chaotic maps exists, and after performing simulations with several chaotic maps, the best results were obtained by using the logistic map, that has been afterward selected to be implemented in ALO. The chaotic sequence that proposed modified ALO method utilizes is generated by logistic mapping, according to Eq. (10):

$$\mathrm{ch}_{i+1} = \mu \mathrm{ch}_{i+1} \times (1 - \mathrm{ch}_i), \quad i = 1, 2, \ldots, S - 1 \tag{10}$$

here μ and S are the control parameter and size of population, respectively. The μ parameter has the value up to 4, while $0 < \mathrm{ch}_1 < 1$, and $\mathrm{ch}_1 \neq 0.25, 0.5, 0.75, 1$. Solutions are being mapped to the generated chaotic sequences by using the following equation:

$$X_i^c = \mathrm{ch}_i X_i, \tag{11}$$

where X_i^c denotes the new position of solution i after chaotic perturbations. By performing the chaotic-based initialization, quality of the initial population is enhanced and the search process can use more rounds for exploitation. The implementation of chaotic population initialization is shown in Algorithm 2.

Algorithm 2 Pseudo-code for chaotic-based population initialization

Step 1: Generate standard random population P of N solutions with expression: $X_i = LB + (UB - LB) * \text{rand}(0, 1),\ i = 1, \ldots, N$, where rand$(0, 1)$ is pseudo-random number drawn from the interval [0, 1].
Step 2: Generate chaotic population P^c of N individuals by mapping solutions from P to chaotic sequences using expressions (10) and (11).
Step 3: Calculate fitness of all solutions from P and P^c.
Step 4: Sort all solutions from $P \cup P^c$ according to fitness.
Step 5: Select N best individuals from sorted set $P \cup P^c$ as initial population.

4 Experimental Setup

This section briefly represents the process related to fake news detection, divided in 5 phases, starting from the data collection, data preprocessing, over selection and extraction of features, execution of the model, and finally its evaluation. The environment for the experiments is equal to the referenced paper [9], as it was necessary to perform valid comparative analysis.

4.1 Dataset Collection

The chosen dataset was COVID-19 Fake News Dataset [37] that was originally created by Koirala [38]. This dataset carry true and false news (total over six thousands) related to the ongoing COVID-19 epidemic, collected from the December 2019 until July 2020. The news were collected by recognizing the keywords corona, coronavirus, and COVID-19, and labeled 0 for false and partly false, and 1 for completely true news.

4.2 Feature Extraction and Data Preprocessing

Extraction of features and preprocessing of data were performed in the same way as in the referenced paper [9]. Extraction of features was executed by tokenization of extracted tweets. Three different models were used for extraction of the word charac-

Table 1 Dataset representation

Dataset	Tokenization	Stemming	Number of features
Data 1	Binary	No	1231
Data 2	TF-IDF	No	1231
Data 3	TF	No	1231
Data 4	TF-IDF	Snowball stemming	1240
Data 5	TF-IDF	Lovins stemming	1223
Data 6	Bag-of-words	Dawson stemming	611

teristics: bag-of-words (BoW), term frequency and term frequency-inverse document frequency (TF and TF-IDf). To reduce the dimensionality of the problem, stemming was performed with three stemmers: Dawson, Lovins, and snowball. This process resulted in six combinations of the dataset based on various textual representations and stemming methods that were used, as given in Table 1.

4.3 Feature Selection

The proposed modified ALO was applied as the wrapper feature selection search method, together with k-nearest neighbors (k-NN) classifier that evaluated the selected features. The resulting subset was treated as a vector with length n (all features) and values 0 (specific feature not chosen) and 1 (specific feature chosen), as in [9]. The quality of selected features was verified when it comes to the obtained classifier accuracy and number of chosen features.

4.4 Model Implementation

The fitness function of the implemented model is defined by Eq. (12):

$$\text{Fitness} = \alpha \times (1 - \text{accuracy}) + \left(1 - \alpha * \frac{|S|}{|W|}\right) \tag{12}$$

here accuracy is resolved by the fraction of the correctly predicted true and fake news, in comparison to the total amount of classifications. Number $|S|$ represents the number of features that were chosen, while $|W|$ stands for the total amount of features, while $\alpha \in [0, 1]$. Dataset split to train and test parts was performed by fivefold cross-validation, as in [9].

4.5 Evaluation and Assessment

Model was evaluated by four standard metrics, namely precision, accuracy, g-mean, and recall. True positives (TP) are news that were forecasted as fake and confirmed to be fake. True negatives (TN) similarly as TP represent true news. False positives (FPs) mean the group of news forecasted as false but turned out to be true. False negatives (FN) on the other hand, are the news forecasted as true but were actually fake. With these terms defined, it is possible to compute the precision, accuracy, g-mean, and recall by the following equations.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{FP} + \text{TP} + \text{FN} + \text{TN}} \tag{14}$$

$$F\text{-Measure} = \frac{(2 \times \text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precission})} \tag{15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

5 Experimental Results

The simulations were carried out in a manner similar to that described in the referenced study [9]. We developed ALO and modified ALO and ran tests for the k-NN-ALO and k-NN-mALO techniques and compared the outcomes to other metaheuristics approaches derived from [9], namely k-NN-BSSA, k-NN-BPSO, k-NN-BGA, and basic k-NN. We have also implemented the arithmetic optimization algorithm (AOA), as it is one of the most popular approaches today, and included the k-NN-AOA results as well for reference.

The suggested k-NN-mALO had the best results on the dataset with BoW representation, with almost 0.75% accuracy and 644 features chosen from the original dataset. In terms of accuracy and precision, it beat other metaheuristic methods, and it was second best (behind k-NN-BPSO) for the F-measure and recall, as shown in Table 2, where bold marks the best result. The basic k-NN-ALO variant achieved average results.

Table 3 displays the dataset's outcomes using the TF-IDF format. In this experiment, the pure k-NN strategy achieved the highest accuracy of 0.70%, followed by the recommended k-NN-mALO approach, which achieved 0.69% and outperformed all other metaheuristics techniques. However, for several metrics, the suggested k-NN-mALO produced the best results (precision, recall, and F-measure).

Table 2 Classification outputs on BoW dataset for observed algorithms with FS and K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.7264	0.6045	0.5605	0.5817	790.4
k-NN-BPSO	0.7258	0.6194	**0.6030**	**0.6111**	619.8
k-NN-BGA	0.7348	0.6468	0.5142	0.5729	631.2
k-NN	0.7053	0.5628	0.5888	0.5755	All
k-NN-AOA	0.7401	0.6493	0.5928	0.6091	642.3
k-NN-ALO	0.7038	0.5902	0.5644	0.5925	648.5
k-NN-mALO	**0.7496**	**0.6541**	0.5936	0.6104	644.2

Table 3 Classification outputs on TF-IDF dataset for observed algorithms with FS and K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.6161	0.4587	0.7297	0.5633	753.6
k-NN-BPSO	0.6639	0.6194	0.6030	0.6111	613.8
k-NN-BGA	0.6764	0.6468	0.5142	0.5729	619.2
k-NN	**0.7053**	0.5628	0.5888	0.5755	All
k-NN-AOA	0.6814	0.6523	0.7314	0.6245	623.6
k-NN-ALO	0.6723	0.6388	0.5853	0.5936	649.2
k-NN-mALO	0.6949	**0.6558**	**0.7327**	**0.6311**	646.5

The findings for the dataset using TF representation are shown in Table 4. The suggested k-NN-mALO technique had the greatest accuracy of around 0.74%, just beating out k-NN-AOA and k-NN-BGA, which came in second and third, respectively, with 0.739 and 0.734%. Regarding to accuracy and recall metrics, the suggested k-NN-mALO approach outperformed the k-NN-AOA, k-NN-BSSA, and k-NN-BGA methods by a little margin. Regarding to the F-measure, the k-NN-mALO approach came in second place, just ahead of the k-NN-BSSA method.

The findings of the TF-IDF with snowball stemming on the dataset are shown in Table 5. The recommended k-NN-mALO technique produced the highest accuracy and precision metrics. When it comes to recall, the basic k-NN fared best, whereas the k-NN-BSSA performed best in terms of F-measure. The basic k-NN-ALO achieved very average performances.

Table 6 shows the findings for the dataset using TF-IDF and Lovins stemming. For accuracy and precision measures, the suggested k-NN-mALO approach outperformed the k-NN-AOA method, which was a close second. The best results were attained by k-NN-BSSA and k-NN-BGA for the recall and F-measure metrics, respectively. Again, the basic k-NN-ALO approach yielded average results.

The results of BoW and Dawson stemming on the dataset are shown in Table 7. Regarding to accuracy and precision, the recommended k-NN-mALO strategy once

Table 4 Classification outputs on TF dataset for observed algorithms with FS and regular K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.7332	0.6095	0.5945	**0.6019**	794.0
k-NN-BPSO	0.7348	0.6039	0.5577	0.5799	613.4
k-NN-BGA	0.7348	0.6442	0.4877	0.5551	606.4
k-NN	0.7053	0.5628	0.5888	0.5755	All
k-NN-AOA	0.7392	0.6483	0.6015	0.5844	611.6
k-NN-ALO	0.7385	0.6463	0.5997	0.5815	617.2
k-NN-mALO	**0.7405**	**0.6488**	**0.6026**	0.5993	612.6

* best results from each category are marked with bold style

Table 5 Classification outputs on Snowball dataset with TF-IDF for observed algorithms with FS and regular K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.7220	0.5782	0.6673	**0.6196**	808.6
k-NN-BPSO	0.7187	0.5810	0.6134	0.5968	625.8
k-NN-BGA	0.7251	0.5949	0.5955	0.5952	602.4
k-NN	0.6963	0.5408	**0.6957**	0.6085	All
k-NN-AOA	0.7328	0.6014	0.6377	0.6043	607.8
k-NN-ALO	0.7314	0.6022	0.6021	0.5995	614.3
k-NN-mALO	**0.7371**	**0.6127**	0.6412	0.6108	615.2

* best results from each category are marked with bold style

Table 6 Classification outputs on Lovins dataset with TF-IDF for observed algorithms with FS and regular K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.7261	0.5962	**0.5974**	0.5968	789.4
k-NN-BPSO	0.7312	0.6198	0.5378	0.5759	616.2
k-NN-BGA	0.7543	0.6622	0.5633	**0.6088**	605.4
k-NN	0.7065	0.5642	0.5936	0.5785	All
k-NN-AOA	0.7746	0.6836	0.5874	0.5963	611.5
k-NN-ALO	0.7391	0.6546	0.5741	0.5883	615.2
k-NN-mALO	**0.7792**	**0.6847**	0.5934	0.5982	618.6

* best results from each category are marked with bold style

Table 7 Classification outputs on Bow and Dawson dataset with for observed algorithms with FS and regular K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.7338	0.6111	0.5926	**0.6017**	400.0
k-NN-BPSO	0.7393	0.6319	0.5548	0.5908	301.6
k-NN-BGA	0.7402	0.6351	0.5510	0.5901	300.0
k-NN	0.6985	0.5499	**0.6144**	0.5804	All
k-NN-AOA	0.7524	0.6405	0.5654	0.6007	305.0
k-NN-ALO	0.7316	0.6382	0.5573	0.5922	302.0
k-NN-mALO	**0.7583**	**0.6412**	0.5667	0.6015	311.5

* best results from each category are marked with bold style

Table 8 Classification outputs on Lovins dataset with TF-IDF for typical classifiers and algorithms with FS and regular K-NN (no FS)

Approach	Accuracy	Precision	Recall	F-measure	Chosen features
k-NN-BSSA	0.7261	0.5962	**0.5974**	0.5968	789.4
k-NN-BPSO	0.7312	0.6198	0.5378	0.5759	616.2
k-NN-BGA	0.7543	0.6622	0.5633	**0.6088**	605.4
k-NN	0.7065	0.5642	0.5936	0.5785	All
J48	0.7229	0.5960	0.5690	0.5822	All
RF	0.7041	0.6342	0.3015	0.4087	All
SVM	0.7075	0.5665	0.5879	0.5770	All
k-NN-AOA	0.7746	0.6836	0.5874	0.5963	611.5
k-NN-ALO	0.7391	0.6546	0.5741	0.5883	615.2
k-NN-mALO	**0.7792**	**0.6847**	0.5934	0.5982	618.6

* best results from each category are marked with bold style

again outperformed other techniques, with the k-NN-AOA method coming in second and the k-NN-BGA method coming in third. The k-NN and k-NN-BSSA approaches received the greatest recall and F-measures scores, while the recommended approach finished in third and second place, respectively.

On the same lines as [9], the k-NNAOA method's results were compared to those produced by the most popular conventional techniques from the current literature: J48, Random Forest, and SVM, as shown in Table 8. TF-IDF and Lovins deriving results from [9] were merged with results from Table 6. On this specific dataset, the suggested k-NN-mALO technique beat the standard methods by a wide margin.

6 Conclusion

The ongoing COVID-19 pandemic is definitely the biggest health challenge that human kind encountered since the Spanish flu hundred years ago. The COVID-19

has affected lives of billions of people in the whole world. This unusual and delicate situation, unfortunately resulted in an enormous quantity of false news that can be related to COVID-19, disease spreading and transmission, possible medications and treatments, and finally vaccination, that was and still is spreading through the social media websites, resulting in uncertainty, ignorance, doubts, skepticism, dismay, and panic in the population.

This type of false information during the COVID-19 pandemic is very hazardous, and it can result in leveraged death toll. As a way to try to prevent it, the researches implemented various models that can detect fake news. The research presented in this manuscript proposes an efficient ALO enhanced k-NN model for detection of the COVID-19 misinformation. The modified ALO was employed as a wrapper to facilitate the feature selection and reduce the quantity of features, while enhancing the accuracy of the k-NN. The suggested model was named k-NN-mALO, developed and verified against the recently published Koirala dataset. The resulting performances of the suggested model have been compared to the basic k-NN and other novel swarm intelligence models, including basic k-NN-ALO, k-NN-AOA, k-NN-BGA, k-NN-BSSA, and k-NNB-PSO, and some traditional approaches such as random forest and SVM, and obtained experimental results clearly indicate that the k-NN-mALO achieves higher accuracy than other approaches observed in the research.

Future investigations will focus on application of the suggested model to different domains of interest, like detection of the false news in economy, military, etc.

References

1. Brezocnik L, Fister I, Podgorelec V (2018) Swarm intelligence algorithms for feature selection: a review. Appl Sci 8:1521
2. Nguyen BH, Xue B, Zhang M (2020) A survey on swarm intelligence approaches to feature selection in data mining
3. Unler A, Murat A, Chinnam RB (2011) mr2pso: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification
4. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40:16–28
5. Bezdan T, Cvetnic D, Gajic L, Zivkovic M, Strumberger I, Bacanin N (2021) Feature selection by firefly algorithm with improved initialization strategy. In: 7th Conference on the engineering of computer based systems, pp 1–8
6. Chuang LY, Chang HW, Tu CJ, Yang CH (2008) Improved binary PSO for feature selection using gene expression data. Comput Biol Chem 32(1):29–38
7. Al-Tashi Q, Kadir SJA, Rais HM, Mirjalili S, Alhussian H (2019) Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access 7:39496–39508
8. Stoean C (2016) In search of the optimal set of indicators when classifying histopathological images. In: 2016 18th International symposium on symbolic and numeric algorithms for scientific computing (SYNASC), pp 449–455. https://doi.org/10.1109/SYNASC.2016.074
9. Al-Ahmad B, Al-Zoubi A, Abu Khurma R, Aljarah I (2021) An evolutionary fake news detection method for covid-19 pandemic information. Symmetry 13:1091
10. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4. IEEE, pp 1942–1948

11. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 39(3):459–471
12. Yang XS (2009) Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. Springer, Berlin, pp 169–178
13. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
14. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872
15. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
16. Wang GG, Deb S, Cui Z (2019) Monarch butterfly optimization. Neural Comput Appl 31(7):1995–2014
17. Bezdan T, Petrovic A, Zivkovic M, Strumberger I, Devi VK, Bacanin N (2021) Current best opposition-based learning salp swarm algorithm for global numerical optimization. In: 2021 Zooming innovation in consumer technologies conference (ZINC). IEEE, pp 5–10
18. Bacanin N, Tuba E, Zivkovic M, Strumberger I, Tuba M (2019) Whale optimization algorithm with exploratory move for wireless sensor networks localization. In: International conference on hybrid intelligent systems. Springer, Berlin, pp 328–338
19. Zivkovic M, Bacanin N, Tuba E, Strumberger I, Bezdan T, Tuba M (2020) Wireless sensor networks life time optimization based on the improved firefly algorithm. In: 2020 International wireless communications and mobile computing (IWCMC). IEEE, pp 1176–1181
20. Zivkovic M, Bacanin N, Zivkovic T, Strumberger I, Tuba E, Tuba M (2020) Enhanced grey wolf algorithm for energy efficient wireless sensor networks. In: 2020 Zooming innovation in consumer technologies conference (ZINC). IEEE, pp 87–92
21. Bacanin N, Bezdan T, Tuba E, Strumberger I, Tuba M, Zivkovic M (2019) Task scheduling in cloud computing environment by grey wolf optimizer. In: 2019 27th Telecommunications forum (TELFOR). IEEE, pp 1–4
22. Bezdan T, Zivkovic M, Tuba E, Strumberger I, Bacanin N, Tuba M (2020) Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. In: International conference on intelligent and fuzzy systems. Springer, Berlin, pp 718–725
23. Bacanin N, Bezdan T, Zivkovic M, Chhabra A (2022) Weight optimization in artificial neural network training by improved monarch butterfly algorithm. In: Mobile computing and sustainable informatics. Springer, Berlin, pp 397–409
24. Bezdan T, Stoean C, Naamany AA, Bacanin N, Rashid TA, Zivkovic M, Venkatachalam K (2021) Hybrid fruit-fly optimization algorithm with k-means for text document clustering. Mathematics 9(16):1929
25. Cuk A, Bezdan T, Bacanin N, Zivkovic M, Venkatachalam K, Rashid TA, Devi VK (2021) Feedforward multi-layer perceptron training by hybridized method between genetic algorithm and artificial bee colony. In: Data science and data analytics: opportunities and challenges, p 279
26. Gajic L, Cvetnic D, Zivkovic M, Bezdan T, Bacanin N, Milosevic S (2021) Multi-layer perceptron training using hybridized bat algorithm. In: Computational vision and bio-inspired computing. Springer, Berlin, pp 689–705
27. Milosevic S, Bezdan T, Zivkovic M, Bacanin N, Strumberger I, Tuba M (2021) Feed-forward neural network training by hybrid bat algorithm. In: Modelling and development of intelligent systems: 7th international conference, MDIS 2020, Sibiu, Romania, 22–24 Oct 2020. Revised selected papers 7. Springer International Publishing, pp 52–66
28. Strumberger I, Tuba E, Bacanin N, Zivkovic M, Beko M, Tuba M (2019) Designing convolutional neural network architecture by the firefly algorithm. In: 2019 International young engineers forum (YEF-ECE). IEEE, pp 59–65
29. Zivkovic M, Bacanin N, Venkatachalam K, Nayyar A, Djordjevic A, Strumberger I, Al-Turjman F (2021) Covid-19 cases prediction by using hybrid machine learning and beetle antennae search approach. Sustain Cities Soc 66:102669
30. Zivkovic M, Venkatachalam K, Bacanin N, Djordjevic A, Antonijevic M, Strumberger I, Rashid TA (2021) Hybrid genetic algorithm and machine learning method for covid-19 cases prediction. In: Proceedings of international conference on sustainable expert systems: ICSES 2020, vol 176. Springer Nature, Berlin, p 169

31. Bezdan T, Milosevic S, Venkatachalam K, Zivkovic M, Bacanin N, Strumberger I (2021) Optimizing convolutional neural network by hybridized elephant herding optimization algorithm for magnetic resonance image classification of glioma brain tumor grade. In: 2021 Zooming innovation in consumer technologies conference (ZINC). IEEE, pp 171–176
32. Bezdan T, Zivkovic M, Tuba E, Strumberger I, Bacanin N, Tuba M (2020) Glioma brain tumor grade classification from MRI using convolutional neural networks designed by modified FA. In: International conference on intelligent and fuzzy systems. Springer, Berlin, pp 955–963
33. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 80–98
34. Caponetto R, Fortuna L, Fazzino S, Xibilia MG (2003) Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans Evol Comput 7(3):289–304
35. Jin Q, Lin N, Zhang Y (2021) K-means clustering algorithm based on chaotic adaptive artificial bee colony. Algorithms 14(2):53
36. Wang M, Chen H (2020) Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. Appl Soft Comput 88:105946
37. Mendeley data—covid-19 fake news dataset. https://data.mendeley.com/datasets/zwfdmp5syg/1. Accessed on 27 Aug 2021
38. Koirala A (2020) Covid-19 fake news classification with deep learning. Preprint

Chapter 35
Improving Nepali News Classification Using Bidirectional Encoder Representation from Transformers

Prakash Kafle, Roshan Chaitrakar, and Kriti Nemkul

1 Introduction

Due to the increment in digitalization technology, online news readers are increasing rapidly. Because of increasing online news portals, different types of news are generated on various topics. Content analysis to recommendation systems all is beneficial from the automatic categorization of news and social media posts. Deep learning is becoming popular for solving machine learning problems rather than the traditional machine learning approaches. The transformer is widely used for translating languages for a large number of languages and has become very popular. The result of the transformer model is accurate in different tasks. The architecture of the transformer is used by the different algorithms, and they perform well for the different tasks in NLP. The news generated by those news portals is not classified into different categories. So, when users want to find some specific category of news, they need to search it manually. Sponsored content is displayed in social media rather than the user's choice. So, there is a need for a proper news classification system that helps users to find the specific category of news. Proper classification of news helps users to display the recommended news as well. In this research, the news is classified into ten categories by taking five different news portals.

Models built with the transformer network, and its variants have lately demonstrated success in a large number of different natural language processing tasks, especially for languages from a lot of resources. Deep learning frameworks have

P. Kafle (✉) · R. Chaitrakar
Nepal College of Information Technology, Pokhara University, Lalitpur, Nepal
e-mail: kaflep26@ncit.edu.np

R. Chaitrakar
e-mail: roshanchi@gmail.com

K. Nemkul
Ratna Rajyalaxmi Campus, Tribhuvan University, Kathmandu, Nepal
e-mail: kriti.nemkul@rrlc.tu.edu.np

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_36

shown more accuracy in recent years than previously developed machine learning techniques. The results of deep learning networks are showing signs of being vastly superior to those techniques. Transformers contain two parts, encoder and decoder. BERT uses only the encoder part of the transformers and can achieve state of the art in different tasks like text classification, next sentence prediction, fill in the blanks, question answering task, and named entity recognition.

2 Literature Review

Various efforts have been undertaken in recent years to categorize Nepali news into various categories. Most of the studies have been conducted by taking raw data from the various news portals to classify the news in different classes. Some researchers have used the different available datasets and some have used their own datasets. Munikar et al. [1], in their research work "Fine-grained sentiment classification using BERT," Complex designs are outperformed by the BERT model in which a fine-grained pretrain model was applied on the Stanford Sentiment Treebank (SST) dataset and there were five different classes for classification of movie reviews in sentiment analysis. González-Carvajal et al. on their research work "Comparing BERT against traditional machine learning text classification," Chinese Hotel reviews were used for classifying positive and negative classes. Chinese language reviews were translated by using a Google translator. BERT produces 93.81% accuracy, whereas auto ml accuracy was 73.99% [2]. Kaur et al. on research work "Online news classification using Deep Learning Technique," the neural network classifier was used to classify four different types of news articles. Results obtained from the neural network were compared with random forest. The accuracy of neural networks was much higher than the traditional methods [3]. Zhou et al. [4] worked on "A C-LSTM Neural Network for Text Classification" and found that C-LSTM was able to learn phrase-level features through a convolutional layer; sequences of such higher-level representations were then fed into the LSTM to learn long-term dependencies. Kafle et al. successfully tested the use of neural networks in the word2vec for word embedding to improve the vector representation of the text. The result obtained from TF-IDF was compared with the result obtained from word2vec. There was an improvement in the overall results by 1.6%. SVM was able to classify the 12-news site using the word2vec model [5]. Basnet et al. on research work "Improving Nepali news recommendation using a classification based on LSTM recurrent neural networks," argued LSTM using word2vec was able to outperform the results obtained from SVM. There was an increment of accuracy and precision around 3 and 5% on Nepali news classification [6]. Sun et al. on their research work "How to fine-tune BERT for text classification?" to provide the simple fine-tuning BERT model they had experimented large number of experiments [6]. On eight highly investigated text categorization datasets, the suggested approach obtains unbelievable results [7]. de Vries et al. on their research work "BERTje: A Dutch BERT Model" created and tested BERTje, a monolingual Dutch BERT model based on the BERT architecture and parameters. Large number

of tokens were used in BERTje so it was able to outperform multilingual BERT model on different NLP related tasks [8]. Victor Sanh et al. on research work "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter." DistilBERT was developed from the scratch which was small model as compared to another BERT model. The model was faster and can be used for language understanding task with high accuracy [9]. Jain et al. on research work "Indic-transformers: An analysis of Transformer language models for Indian languages," the performance of the fine-tuning model and the results achieved after various parameter changes were compared to a model that was built from the scratch [5]. The languages Hindi, Bengali, and Telugu were employed for the question answering, and POS tagging in Bengali, Telugu, and Hindi were done using transformers-based models. BERT, ROBERTA, DistilBERT were used for different tasks on different languages [10].

3 Data Collection

For this research, Nepali news raw data have been collected from five different news portals. Python language was used to extract news with beautiful Soup library. All the raw texts of desired categories were scraped from the news website and stored in text file. Only ten categories of data are used in this research. News category and news text have been collected from below news sites. All the scrapped data were first stored in text file, and after that, the text has been stored in csv file; in one column, it contains news text; and in another column, it contains respective label. The following websites have been used to obtain data (Table 1).

The news which are collected from different sites are further classified into following categories. News from different sites are classified into ten different categories as given in Table 2.

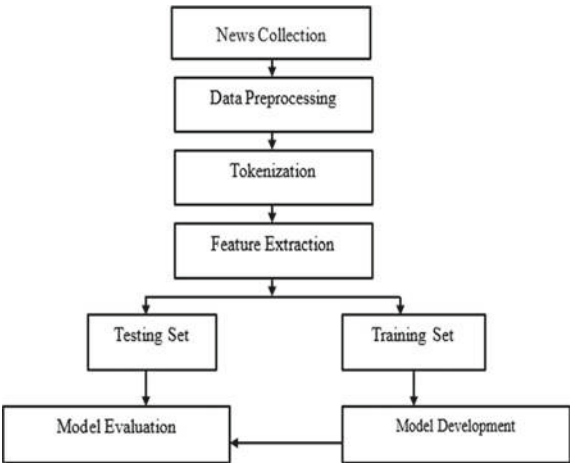
Table 1 News portal names and URLs

S.No.	Name of the website	URL	Article numbers
1	DC Nepal	https://www.dcnepal.com	12,842
2	Imagekhabar	https://www.imagekhabar.com	19,296
3	Onlinekhabar	https://www.onlinekhabar.com	20,504
4	Ratopati	https://ratopati.com	7855
5	Ujayaloonline	https://ujyaaloonline.com	11,473
		Total	**71,970**

Table 2 News categories and description

S.No.	Category name	Number of article
1	Diaspora	6224
2	Economy	14,069
3	Entertainment	7588
4	Health	3122
5	International	9879
6	Opinion	2675
7	Politics	8352
8	Society	4979
9	Sports	13,018
10	Technology	2064
	Total	**71,970**

Fig. 1 Process flow



4 Proposed Model

See Fig. 1.

5 Data Preprocessing

In this phase, noisy unwanted, and uninformative data were removed that do not make sense in the classification process. After data preprocessing, the data contain only the specific information from which we can extract the desired information. Keeping all the raw text may be difficult to process and that does not give meaningful result

so data cleaning is required. Corpus consists of two columns in the first column. It contains the text, and in another column, it contains the category of the news. Data preprocessing is performed by removing HTML tags, white space, special symbols, and stop words. Label encoding of the data is done into ten categories. BERT uses its own tokenizer which is used for tokenization. BERT-based multilingual tokenizer from pretrained model is used for the tokenization process from the transformer library. 80% of data are used as training set and remaining 20% of data was used as a test set.

6 Model Building

To train the model, pretrain model from the BERT multilingual cased version is used in accordance with the training set created during the data section phase. The accuracy and other model evaluation criteria are evaluated, and different parameters are changed to improve the accuracy and other model evaluation criteria.

The BERT model has the following configuration:

1. Sequence length 512
2. Number of encoder layer 12, i.e., default by the BERT base
3. 4 epochs
4. Batch size 5
5. Optimizer AdamW with learning rate 1e−5, i.e., 0.0001
6. 768 input vectors
7. 12 Attention Heads

The LSTM, GRU, BiLSTM, BiGRU algorithm are used to train the model. Those data which are split for train set are used to train the model. The accuracy and other model evaluation criteria have been evaluated, and different parameters are changed to improve the accuracy and other model evaluation criteria.

LSTM, GRU, BiLSTM, BiGRU model has the following configuration:

1. Sentence length 200
2. Embedding vector feature 40
3. Vocabulary size 15000
4. 100 hidden units of first layer for LSTM
5. 50 hidden unit for dense layer
6. 15 epochs
7. Dropout 0.2
8. As a loss function, sparse categorical cross-entropy
9. Softmax activation as output
10. Adam as optimizer

7 Transformer

The transformer is a deep learning model introduced in 2017 that utilizes the mechanism of attention, weighing the influence of different parts of the input data [11]. Without using sequence-aligned RNNs or convolution, the transformer is the first transduction model to compute representations of its input and output using only self-attention. The transformer is used to solve the different NLP tasks. Encoder and decoder are the two parts of a transformer. Figure 2 shows the architecture of transformers.

Transformer consists of two different components, i.e., encoder and decoder. From the encoder, data is sent to the model, and finally, output is obtained after the processing of data in the encoding part. The data which is processed in encoder layer is sent to the decoder. The encoding and decoding consist of six layers which are stacked with each other. Inside the encoder layer, there are two sub-components.

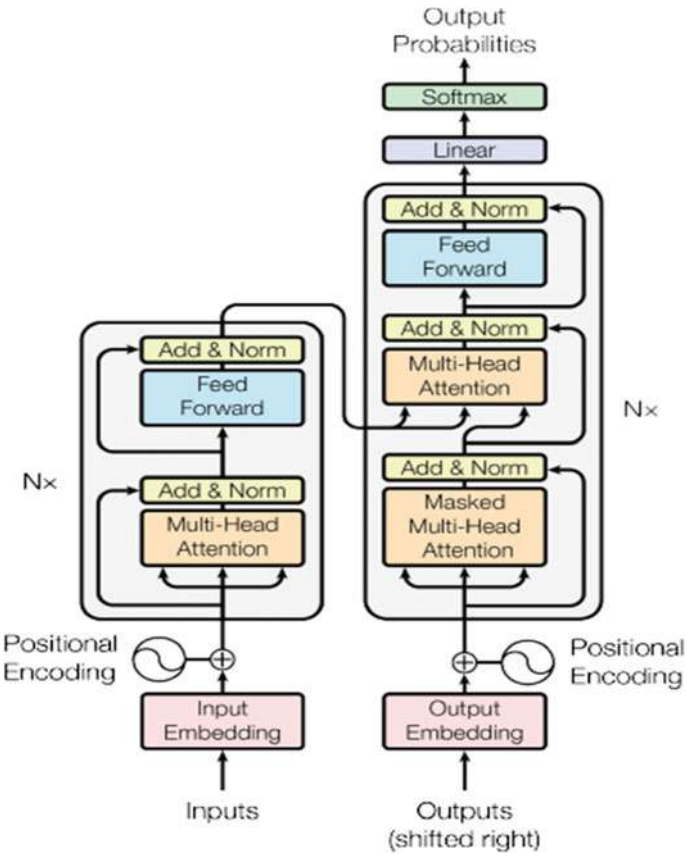


Fig. 2 Transformer [11]

They are self-attention and feed-forward neural networks. The first sequence is passed through the attention layer which is used for calculating the attention, and finally, the output of the attention is passed into the feed-forward neural network transformer that uses the maximum sequence length of 512 and 8 attention head. After the data is passed into the encoder layer, there is a complex mathematical operation to calculate the query key and value to provide the attention.

Six identical layers stacked with each other are used in encoder, in which two sub-layers, multi-head self-attention and simple position-wise fully connected feedforward network, are employed by residual connection around both sub-layers followed by layer normalization [12, 13].

Self-attention can be defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d_k}}\right)V \tag{1}$$

where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimension of the Q and K matrices multi-head attention can now be defined as

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^0 \tag{2}$$

where $\text{head}_i = \text{Attention}\ (QW_i{}^Q,\ KW_i{}^K,\ VW_i{}^V)$. Multi-head attention consists on projecting the queries, keys, and values h times with different, learned linear projections to d_k, $d_{k,}$ and d_v (dimension of the values matrix), respectively.

In transformer model, attention is calculated by using key, queries, and value parameter. In first phase, when word embedding vector of length 512 is multiplied with the weight matrix W^Q, then we get the matrix for query. When weight matrix K is multiplied with the input embedding, then we get the key matrix. When value matrix W^V is multiplied with the input embedding, we get the value matrix. After key matrix, value matrix, and query matrix are obtained in second phase dot product of the query vector, the key vector is used to calculate the score. After the score is calculated, then it is dividing by the square root of the dimension of the key vector, and then softmax is calculated. The value of softmax obtained is now multiplied with the value vector, and then by summing up those value, we get the new value which is sent to the feed-forward network.

Mathematically,

$$X \times W^Q = Q \tag{3}$$

$$X \times W^K = K \tag{4}$$

$$X \times W^V = V \tag{5}$$

Finally, attention is calculated as:

$$\text{Attention} = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d_k}}\right)V \tag{6}$$

where X denotes the input vector, W^Q denotes the weight matrix, Q denotes the query vector, W^K denotes the key matrix, K denotes the key vector, W^V denotes the value matrix, V denotes the value vector, and d_k denotes the dimension of the key vector.

8 Experiment Results and Analysis

The experiment has been carried out in Google Colab. The data was trained and tested using BERT, LSTM, GRU, BiGRU, and BiLSTM models. From the comparison, we found that the BERT performed well in Nepali news classification with 94% precision, 93% recall, 94% F_1-score, accuracy 95%, and micro-AUC 0.99. BERT is a transformer-based machine learning technique with two major steps: pretraining and fine-tuning. BERT-based model contains 768 hidden units and 12 attention heads. Whereas BERT large consists of 24 layers, 1024 hidden units, 16 heads, and total 340M parameters. 80% of the data is used for training, and 20% of data is for testing. Sequence length 512 is used for this training and testing purpose. The comparative chart of the models is shown in Fig. 3 (Table 3).

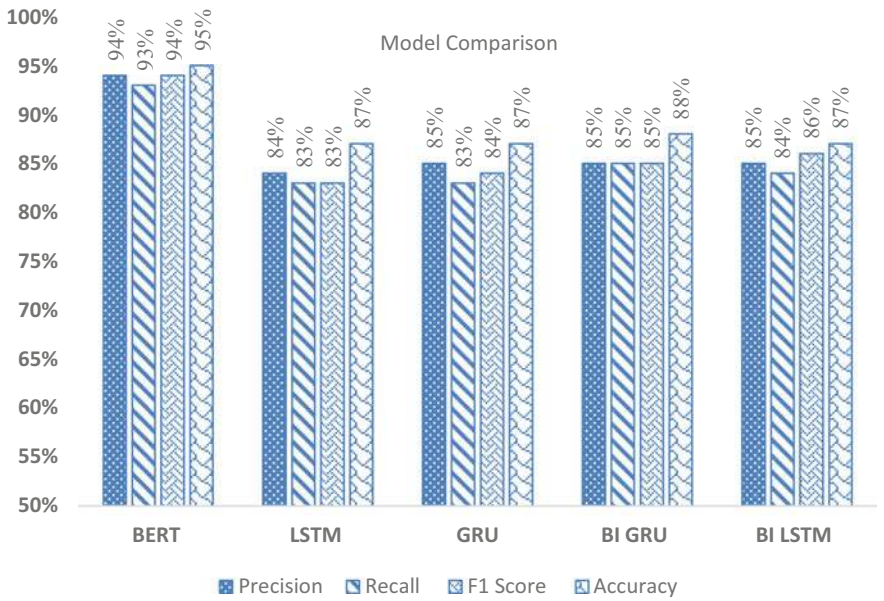


Fig. 3 Model comparison chart

Table 3 Result comparison between models

Model	Precision (%)	Recall (%)	F_1-score (%)	Accuracy (%)	Micro-AUC
BERT	94	93	94	95	0.99
LSTM	84	83	83	87	0.93
GRU	85	83	84	87	0.93
BiGRU	85	85	85	88	0.93
BiLSTM	85	84	86	87	0.93

9 Conclusion

Use of transformer-based neural network shows the improvement on news classification over LSTM, GRU, BiLSTM, BiGRU. In all models, sports category is classified with high accuracy, precision, recall, and micro-AUC. Due to the use of multi-headed attention mechanism, BERT model produces good result as compared to other models. In case of other models, the sequence of sentence is passed sequentially. So, it is unable to store the information of every sequence which are linked with each other but BERT passes the sequence at once parallelly, and there exists a relationship between different sequences. This is the main reason why BERT is powerful than other models. Even a bidirectional LTSM and GRU cannot see the information truly in both directions but BERT can. The pretrain model of BERT is trained on large corpus and available on different languages, but all other models must be trained from scratch by selecting different parameters depending on use case. Large number of encoder layer and parameter are used in BERT as compared to other models which results in the low accurate result for different datasets.

References

1. Munikar M, Shakya S, Shrestha A (2019) Fine-grained sentiment classification using BERT. In: Artificial intelligence for transforming business and society (AITB)
2. González-Carvajal S, Garrido-Merchán EC (2020) Comparing BERT against traditional machine learning text classification. arXiv [cs.CL]
3. Kaur S, Khiva NK (2016) Online news classification using deep learning technique. Int Res J Eng Technol (IRJET) cilt 3(10)
4. Zhou C, Sun C, Liu Z, Lau FCM (2015) A C-LSTM neural network for text classification. arXiv [cs.CL]
5. Kafle K, Sharma D, Subedi A, Timalsina AK (2016) Improving Nepali document classification by neural network. In: IOE graduate conference, pp 317–322
6. Basnet A, Timalsina AK (2018) Improving Nepali news recommendation using classification based on LSTM recurrent neural networks. In: IEEE 3rd International conference on computing, communication and security (ICCCS)
7. Sun C, Qiu X, Xu Y, Huang X (2019) How to fine-tune BERT for text classification? arXiv: 1905.05583v3 [cs.CL]
8. de Vries W, van Cranenburgh A, Bisazza A, Caselli T, van Noord G, Nissim M (2019) BERTje: a Dutch BERT model. ReserchGate

9. Sanh V, Debut L, Chaumond J, Wolf T, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL]
10. Jain K, Deshpande A, Shridhar K, Laumann F, Dash A (2020) Indic-transformers: an analysis of transformer language models for Indian languages. arXiv:2011.02323 [cs.CL]
11. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser LU, Polosukhin I (2017) Attention is all you need. In: Advances in neural ınformation processing systems
12. Devlin J, Chang MW, Toutanova K, BERT: pre-training of deep bidirectional transformers for language understanding. arXiv [cs.CL]
13. Martin L et al (2020) CamemBert: a tasty French language model. In: Proceedings of 58th annual meeting of the association for computationsl linguistics

Chapter 36
Extended TOPSIS and VIKOR Methods Based on a New Distance Measure of Intuitionistic Fuzzy Number

Prakash N. Kamble and Naziya Parveen

1 Introduction

MCDM is the process of selecting the best answer from the feasible options when there are numerous, typically conflicting criteria. Decision makers struggle to make decisions when faced with MCDM challenges with unknown data. Zadeh [1] created the concept of fuzzy sets, which encompasses uncertainty and the degree of truth fullness of values. It consists of membership functions ranging from 0 to 1. In some cases, fuzzy set theory makes it difficult to explain membership functions. To overcome this issue, Atanasov [2] developed the IFS (intuitionistic fuzzy sets), which consists of membership and non-membership degrees, to reduce misunderstanding and ambiguity. A fuzzy set extension is the IFS. Atanasov developed the notion of Atanasov's interval-valued intuitionistic fuzzy sets as a development of IFS. Boran [3] used the TOPSIS approach to offer an intuitionistic fuzzy multi-criteria decision-making method for facility location selection. In an intuitionistic fuzzy environment, Parveen and Kamble [4] enhanced TOPSIS for decision-making. Using interval-valued intuitionistic fuzzy numbers, Izadikhah [5] proposed a TOPSIS technique for rational decision-making for supplier selection. The fuzzy TOPSIS approach using a hexagonal fuzzy number was proposed by Parveen and Kamble [6] for a decision-making problem. Opricovic and Teeng [7] proposed an enhanced VIKOR method in comparison with outranking alternatives. Luo and Wang [8] proposed an enhanced VIKOR approach for intuitionistic fuzzy sets based on a unique distance measure.

Kamble and Parveen [9] combined the FAHP and FTOPSIS methods.

TOPSIS and VIKOR are two popular fuzzy MCDM methods. Both methods rely on an aggregating function to determine how near a solution is to being perfect. Hwang and Yoon [10] invented TOPSIS, which is one of the classic MCDM approaches. The TOPSIS method produces the shortest distance between the positive

P. N. Kamble · N. Parveen (✉)
Dr. Babasaheb Ambedker Marathwada University, Aurangabad, Maharashtra 431001, India
e-mail: naazparveen96@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_37

ideal solution and the biggest distance between the negative ideal solution. Opricovic [11] was the first to propose the VIKOR for multi-criteria optimization of complex systems. The VIKOR method can be used to arrive at a compromise solution that benefits the majority the most and the opponents the least.

In this paper, we offer the TOPSIS and VIKOR methods for ranking sets of alternatives using intuitionistic fuzzy numbers. The rankings of each option in relation to each parameter are taken as IFNs throughout the evaluation process. A new IFN distance metric is proposed in this technique.

2 Preliminaries

We have explored some basic concepts and theories related to IFSs in this part.

2.1 Fuzzy Number (FN)

If the following properties are met, a fuzzy set A on U is termed a fuzzy number.

1. $h(A) = 1$, hence A must be a typical fuzzy set.
2. For each $\alpha \in (0, 1]$, ${}^{\alpha}A$ must be a closed interval.
3. The support of A, ${}^{0+}A$, must be constrained.

2.2 Intuitionistic Fuzzy Set (IFS)

Intuitionistic fuzzy sets (IFS) [12] are the extension of fuzzy sets. Membership function, non-membership function, and hesitation margin are the three parameters that define it. Let X be a set that is not empty. An IFS A defined on X has the form, $A = \{(y, \mu_A(y), \nu_A(y), \pi_A(y)) : y \in X\}$, where the functions $\mu_A(y) : X \to [0, 1]$, $\nu_A(y) : X \to [0, 1]$ and $\pi_A = 1 - \mu_A(y) - \nu_A(y)$ represent the degree of membership, degree of non-membership, and hesitation margin of an element $y \in A \subset X$, respectively.

2.3 Intuitionistic Fuzzy Number (IFN)

An IFN is defined as an IFS [12] $A = (y, \mu_A(y), \nu_A(y))$ of the real line if, An IFS [12] $A = (y, \mu_A(y), \nu_A(y))$ of the real line is called an IFN if,

1. The value A is IF-normal.
2. A is an IF-convex variable.
3. Upper semi continuous $\mu_A(y)$ and lower semi continuous $\mu_A(y)$.
4. The function $A = \{y \in X | \nu_A(y) < 1\}$ is bounded.

2.4 Pentagonal Intuitionistic Fuzzy Number (PIFN)

PIFN [13] of an intuitionistic fuzzy set is denoted as $P = \langle (p_1, p_2, p_3, p_4, p_5); (p_1', p_2', p_3, p_4', p_5') \rangle$ where all $p_1' \leq p_1 \leq p_2' \leq p_2 \leq p_3 \leq p_4 \leq p_4' \leq p_5 \leq p_5'$ are real numbers and its membership function $\mu_P(y)$, non-membership function $\nu_P(y)$ are given by,

$$\mu_P(y) = \begin{cases} 0, & y < p_1 \\ \frac{y-p_1}{p_2-p_1}, & p_1 \leq y \leq p_2 \\ \frac{y-p_2}{p_3-p_2}, & p_2 \leq y \leq p_3 \\ 1, & y = p_3 \\ \frac{p_4-y}{p_4-p_3}, & p_3 \leq y \leq p_4 \\ \frac{p_5-y}{p_5-p_4}, & p_4 \leq y \leq p_5 \\ 0, & y > p_5, \end{cases}$$

$$\nu_P(y) = \begin{cases} 1, & y < p_1' \\ \frac{p_2'-y}{p_2'-p_1'}, & p_1' \leq y \leq p_2' \\ \frac{p_3-y}{p_3-p_2'}, & p_2' \leq y \leq p_3 \\ 0, & y = p_3 \\ \frac{y-p_3}{p_4'-p_3}, & p_3 \leq y \leq p_4' \\ \frac{y-p_4'}{p_5'-p_4'}, & p_4 \leq y \leq p_5' \\ 1, & y > p_5'. \end{cases}$$

2.5 Accuracy Function of PIFN

Accuracy function of a PIFN $P = (p_1, p_2, p_3, p_4, p_5); (p_1', p_2', p_3, p_4', p_5')$ is defined as

$$H(P) = \frac{(p_1 + p_1' + p_2 + p_2' + 2p_3 + p_4 + p_4' + p_5 + p_5')}{5}.$$

3 New Distance Measure of PIFNs

A novel formula for calculating the distance between two PIFNs is proposed. Let $P = \{(p_1, p_2, p_3, p_4, p_5); (p_1', p_2', p_3', p_4', p_5')\}$ and $Q = \{(q_1, q_2, q_3, q_4, q_5); (q_1', q_2', q_3', q_4', q_5')\}$ are two PIFNs, then the new distance measure between P and Q is denoted by $d_n(P, Q)$ and defined by,

$$d_n(P, Q) = \sqrt{\frac{1}{20}\left\{\begin{array}{l}\sum_{i=1}^{5}(q_i - p_i)^2 + \sum_{i=1}^{5}(q_i' - p_i')^2 + \sum_{i\in\{1,2,3,4\}}(q_i - p_i)(q_j - p_j) \\ + \sum_{i\in\{1,2,3,4\}}(q_i' - p_i')(q_j' - p_j') + (q_3 - p_3)^2 + (q_3' - p_3')^2\end{array}\right\}},$$

where $j = i + 1$.

Then, we have the following properties.

Property 1 *If P and Q are two PIFNs, then the distance measure $d_n(P, Q)$ is same as the Euclidean distance.*

Proof Let $P = \{(p_1, p_2, p_3, p_4, p_5); (p_1', p_2', p_3', p_4', p_5')\}$ and $Q = \{(q_1, q_2, q_3, q_4, q_5); (q_1', q_2', q_3', q_4', q_5')\}$ are two PIFNs, then let, $p_1 = p_2 = p_3 = p_4 = p_5 = p_1' = p_2' = p_3' = p_4' = p_5' = P$ and $q_1 = q_2 = q_3 = q_4 = q_5 = q_1' = q_2' = q_3' = q_4' = q_5' = Q$ using above formula of distance measure $d_n(P, Q)$ can be calculated as,

$$\begin{aligned} d_n(P, Q) &= \sqrt{\frac{1}{20}\left\{\begin{array}{l}5(q_1 - p_1)^2 + 5(q_1' - p_1')^2 + 4(q_1 - p_1)(q_2 - p_2) \\ +4(q_1' - p_1')(q_2' - p_2') + (q_3 - p_3)^2 + (q_3' - p_3')^2\end{array}\right\}} \\ &= \sqrt{\frac{1}{20}\left\{\begin{array}{l}5(Q - P)^2 + 5(Q - P)^2 + 4(Q - P)(Q - P) \\ +4(Q - P)(Q - P) + (Q - P)^2 + (Q - P)^2\end{array}\right\}} \\ &= \sqrt{\frac{1}{20}\Big(10(Q - P)^2 + 10(Q - P)^2\Big)} \\ &= \sqrt{(Q - P)^2} \\ d_n(Q, P) &= |Q - P| \end{aligned}$$

Property 2 *Two PIFNs P and Q are identical if and only if $d_n(P, Q) = 0$.*

Proof Let $P = \{(p_1, p_2, p_3, p_4, p_5); (p_1', p_2', p_3', p_4', p_5')\}$ and $Q = \{(q_1, q_2, q_3, q_4, q_5); (q_1', q_2', q_3', q_4', q_5')\}$ are two PIFNs, If P and Q are identical, then $p_1 = q_1$, $p_2 = q_2$, $p_3 = q_3$, $p_4 = q_4$, $p_5 = q_5$, $p_1' = q_1'$, $p_2' = q_2'$, $p_3' = q_3'$, $p_4' = q_4'$, and $p_5' = q_5'$. Using proposed distance formula, the distance between P and Q is,

$$d_n(P, Q) = \sqrt{\frac{1}{20}\left\{\begin{array}{l}(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2 + (q_4 - p_4)^2 \\ +(q_5 - p_5)^2 + (q_1' - p_1')^2 + (q_2' - p_2')^2 + (q_3' - p_3')^2 \\ +(q_4' - p_4')^2 + (q_5' - p_5')^2 + (q_1 - p_1)(q_2 - p_2) \\ +(q_2 - p_2)(q_3 - p_3) + (q_3 - p_3)(q_4 - p_4) \\ +(q_4 - p_4)(q_5 - p_5) + (q_1' - p_1')(q_2' - p_2') \\ +(q_2' - p_2')(q_3' - p_3') + (q_3' - p_3')(q_4' - p_4') \\ +(q_4' - p_4')(q_5' - p_5') + (q_3 - p_3)^2 + (q_3' - p_3')^2\end{array}\right\}}$$

$$= \sqrt{\frac{1}{20}\left\{\begin{array}{l}(p_1 - p_1)^2 + (p_2 - p_2)^2 + (p_3 - p_3)^2 + (p_4 - p_4)^2 \\ +(p_5 - p_5)^2 + (p_1' - p_1')^2 + (p_2' - p_2')^2 + (p_3' - p_3')^2 \\ +(p_4' - p_4')^2 + (p_5' - p_5')^2 + (p_1 - p_1)(p_2 - p_2) \\ +(p_2 - p_2)(p_3 - p_3) + (p_3 - p_3)(p_4 - p_4) \\ +(p_4 - p_4)(p_5 - p_5) + (p_1' - p_1')(p_2' - p_2') \\ +(p_2' - p_2')(p_3' - p_3') + (p_3' - p_3')(p_4' - p_4') \\ +(p_4' - p_4')(p_5' - p_5') + (p_3 - p_3)^2 + (p_3' - p_3')^2\end{array}\right\}}$$

$$= \sqrt{\frac{1}{20}\{0\}}$$

$$= 0$$

Conversely, if $d_n(P, Q) = 0$, then from new distance formula we have,

$$d_n(P, Q) = \sqrt{\frac{1}{20}\left\{\begin{array}{l}(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2 + (q_4 - p_4)^2 \\ +(q_5 - p_5)^2 + (q_1' - p_1')^2 + (q_2' - p_2')^2 + (q_3' - p_3')^2 \\ +(q_4' - p_4')^2 + (q_5' - p_5')^2 + (q_1 - p_1)(q_2 - p_2) \\ +(q_2 - p_2)(q_3 - p_3) + (q_3 - p_3)(q_4 - p_4) \\ +(q_4 - p_4)(q_5 - p_5) + (q_1' - p_1')(q_2' - p_2') \\ +(q_2' - p_2')(q_3' - p_3') + (q_3' - p_3')(q_4' - p_4') \\ +(q_4' - p_4')(q_5' - p_5') + (q_3 - p_3)^2 + (q_3' - p_3')^2\end{array}\right\}}$$

$$= 0$$

Implies $p_1 = q_1$, $p_2 = q_2$, $p_3 = q_3$, $p_4 = q_4$, $p_5 = q_5$, $p_1' = q_1'$, $p_2' = q_2'$, $p_3' = q_3'$, $p_4' = q_4'$, and $p_5' = q_5'$. Therefore, two PIFNs P and Q are identical and the property has been proved.

Property 3 *If P, Q and R are three PIFNs, then intuitionistic fuzzy number Q is closer to P than the other intuitionistic fuzzy number R if and only if $d_n(P, Q) < d_n(P, R)$.*

Proof This property is trivial. For example, three PIFNs are,

$P = (1, 2, 3, 4, 5); (0, 1, 3, 5, 6)$
$Q = (2, 3, 4, 5, 6); (1, 2, 4, 6, 7)$

$R = (3, 4, 5, 6, 7); (2, 3, 5, 7, 8)$

using proposed distance formula we calculate,

$d_n(P, Q) = 1$ and,
$d_n(P, R) = 2$.

According to the distance measure, we can conclude that the PIFN Q is closer to P than R.

4 Proposed Methods

The TOPSIS and VIKOR algorithms are enhanced with a new distance measure in this section. Both approaches have the same initial five steps, which are listed below.

1. Creating a decision-making committee and then describing a finite set of criteria and alternative.
2. Defining the linguistic variables that will be used to rate alternatives. Intuitionistic fuzzy numbers are used to represent these linguistic variables.
3. Calculate the weight of each criterion using the formula below:

$$w_j = \frac{1}{N}[w_j^1 + w_j^2 + \cdots + w_j^n], \quad j = 1, 2, \ldots, m. \tag{1}$$

4. Calculate the fuzzy ratings of each alternative with following formula,

$$x_{ij} = \frac{1}{N}[x_j^1 + x_j^2 + \cdots + x_j^n], \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m, \tag{2}$$

   where rating of each alternative is represented by IFN.
5. Make an intuitionistic fuzzy decision matrix. Each x_{ij} is assumed to be a pentagonal intuitionistic fuzzy number.
   i.e., $x_{ij} = (p_{ij}, q_{ij}, r_{ij}, s_{ij}, t_{ij}); (p_{ij}^{'}, q_{ij}^{'}, r_{ij}^{'}, s_{ij}^{'}, t_{ij}^{'})$

$$D = \begin{pmatrix} x_{11} & \ldots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nm} \end{pmatrix}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m. \tag{3}$$

4.1 TOPSIS Method

The proposed TOPSIS approach is decomposed into the steps below.

1. Calculate the intuitionistic fuzzy decision matrix that has been normalized. The normalized intuitionistic fuzzy decision value is computed using the following relation and operating the arithmetic operation of IFN on it.

$$n_{ij} = (r_{ij}; s_{ij}), \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m, \tag{4}$$

where

$$r_{ij} = \frac{(p_{ij}, q_{ij}, r_{ij}, s_{ij}, t_{ij})}{\sqrt{\sum_{j=1}^{m} (p_{ij}, q_{ij}, r_{ij}, s_{ij}, t_{ij})^2}}, \quad i = 1, 2, \ldots, n.$$

$$s_{ij} = \frac{(p_{ij}^{'}, q_{ij}^{'}, r_{ij}^{'}, s_{ij}^{'}, t_{ij}^{'})}{\sqrt{\sum_{j=1}^{m} (p_{ij}^{'}, q_{ij}^{'}, r_{ij}^{'}, s_{ij}^{'}, t_{ij}^{'})^2}}, \quad i = 1, 2, \ldots, n.$$

2. Construct the weighted normalized intuitionistic fuzzy decision matrix. The weighted normalized value v_{ij} is calculated as:

$$v_{ij} = w_j \times n_{ij}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m, \tag{5}$$

where w_j is the weight of the criterion, which is calculated as in Eq. 1.

3. Determine the intuitionistic fuzzy positive ideal solution and negative ideal solution. They are generated using IFN's accuracy function.

$$A^+ = \{v_1^+, v_2^+, \ldots, v_m^+\}, \tag{6}$$

$$A^- = \{v_1^-, v_2^-, \ldots, v_m^-\}, \tag{7}$$

where each criterion has the greatest number of values in v_j^+, the least values of each criterion item are found in v_j^-.

4. Determine how far apart the two points are. The new distance measure is used to determine the separation measure of each alternative from the intuitionistic fuzzy positive ideal solution as:

$$m_i^+ = \sqrt{\sum_{j=1}^{m} d_n(v_j^+, v_{ij})^2}, \quad i = 1, 2, 3, \ldots, n. \tag{8}$$

Similarly, the distance between the intuitionistic fuzzy negative ideal solution is:

$$m_i^- = \sqrt{\sum_{j=1}^{m} d_n(v_j^-, v_{ij})^2}, \quad i = 1, 2, 3, \ldots, n. \tag{9}$$

5. Determine how the solution is near to the ideal. The relative distance between each alternative and the positive ideal is computed as follows:

$$\mathrm{cl}_i^+ = \frac{(m_i^+)}{(m_i^+ + m_i^-)}. \tag{10}$$

6. Sort your preferences in ascending order. A proximity coefficient of cl_i^+ with the lowest value suggests that the alternatives are performing well. The best option is the one that is the furthest away from the ideal answer.

4.2 VIKOR Method

The steps in the proposed VIKOR technique are as follows:

1. Compute the fuzzy best x_j^+ and the fuzzy worst x_j^- values of all criteria ratings from the decision matrix 3, $j = 1, 2, \ldots, m$.

$$x_j^+ = \begin{cases} \max_i(x_{ij}), & \text{for benifit criteria;} \\ \min_i(x_{ij}), & \text{for cost criteria;} \end{cases} \tag{11}$$

$$x_j^- = \begin{cases} \min_i(x_{ij}), & \text{for benifit criteria;} \\ \max_i(x_{ij}), & \text{for cost criteria;} \end{cases} \tag{12}$$

2. Calculate the individual regret value and the collective utility value. These are the key values of the VIKOR technique. The new distance measure is used to generate this key values, the group utility value S_i and the individual regret value R_i are given as:

$$S_i = \sum_{j=1}^{m} w_j \left(\frac{d(x_j^+, x_{ij})}{d(x_j^+, x_j^-)} \right) \tag{13}$$

$$R_i = \max_j \quad w_j \left(\frac{d(x_j^+, x_{ij})}{d(x_j^+, x_j^-)} \right) \tag{14}$$

3. Compute the compromise value Q_i, which is calculated as follows:

$$Q_i = \nu \left(\frac{S_i + S^+}{S^- - S^+} \right) + (1 - \nu) \left(\frac{R_i + R^+}{R^- - R^+} \right), \tag{15}$$

where w_j is obtained by Eq. 1.

$$S^- = \max_i(S_i),$$
$$S^+ = \min_i(S_i),$$
$$R^- = \max_i(R_i),$$
$$R^+ = \min_i(R_i).$$

ν is a weight for the highest group utility approach in Eq. 15, whereas $1 - \nu$ is the weight of individual regret. ν is commonly considered to have a value of 0.5.

4. Sort the options in order of preference. Create three ranking lists by sorting S_i, R_i, and Q_i in ascending order.
5. Propose a compromise solution, the alternative $\text{Ar}^{(1)}$ that ranks highest in Q_i (least value) and meets both of the following two conditions at the same time.
   Condition 1: If,

$$Q(\text{Ar}^{(2)}) - Q(\text{Ar}^{(1)}) \geq \frac{1}{m-1}, \tag{16}$$

   where $\text{Ar}^{(1)}$ and $\text{Ar}^{(2)}$ are the top two options in Q_i, and m is the total number of options.
   Condition 2: The alternative $\text{Ar}^{(1)}$ should be ranked first by S_i as well as R_i.
   If all of the above conditions cannot be met at the same time, there are several compromise options.

   a. If only requirement 2 is not met, there are two alternatives: $\text{Ar}^{(1)}$ and $\text{Ar}^{(2)}$.
   b. If condition 1 is not met, $\text{Ar}^{(1)}, \text{Ar}^{(2)}, \ldots, \text{Ar}^{(n)}$, here $\text{Ar}^{(n)}$ is calculated by the equation $Q(\text{Ar}^{(n)}) - Q(\text{Ar}^{(1)}) \leq \frac{1}{m-1}$ for the maximum.

5 Numerical Example

A software firm is looking for a software engineer. In this problem, there are five candidates: (Ar_1), (Ar_2), (Ar_3), (Ar_4) and (Ar_5) and four criteria: Oral communication (Ct_1), personality (Ct_2), experience (Ct_3), and self-assurance (Ct_4) are all taken into account. For the selection of the most suitable engineer, a two-person panel of decision makers. It is made up of (Dk_1) and (Dk_2). The MCGDM problem is solved using the proposed fuzzy TOPSIS and fuzzy VIKOR methods using PIFNs. Tables 1 and 2 show the linguistic factors and their accompanying PIFNs for evaluating the alternative and weight of criterion, respectively. Tables 3 and 4 show the ratings of alternatives and the weights of criteria assigned by two decision makers. The fuzzy weights of all criteria are given in Table 5, and aggregated weights are calculated

Table 1 Linguistic terms and their corresponding rating of alternatives PIFN'S

Linguistic terms (LT)	Ratings of alternatives
Poor (P)	⟨(1, 2, 3, 4, 5); (0, 1, 3, 5, 6)⟩
Fair (F)	⟨(2, 3, 4, 5, 6); (1, 2, 4, 6, 7)⟩
good (G)	⟨(3, 4, 5, 6, 7); (2, 3, 5, 7, 8)⟩
Very good (VG)	⟨(4, 5, 6, 7, 8); (3, 4, 6, 8, 9)⟩

Table 2 Linguistic terms and their corresponding weights of criteria

Linguistic terms (LT)	Weights of criteria
Low (L)	0.1
Medium (M)	0.2
High (H)	0.3
Very high (VH)	0.4

Table 3 Rating of alternative by decision makers in LT

Criteria	Ar_1		Ar_2		Ar_3		Ar_4		Ar_5	
	Dk_1	Dk_2	Dk_1	Dk_2	Dk_1	Dk_2	Dk_1	Dk_2	Dk_1	Dk_2
Ct_1	F	F	G	G	F	G	G	VG	VG	VG
Ct_2	VG	VG	G	VG	G	G	F	G	F	F
Ct_3	P	F	P	P	P	F	G	G	P	G
Ct_4	P	P	F	F	VG	G	P	P	G	P

Table 4 Rating of weights of the criteria by decision makers in LT

Criteria	Dk_1	Dk_2
Ct_1	M	H
Ct_2	VH	M
Ct_3	H	VH
Ct_4	L	L

using Eq. 1, shown in Table 6. Equations 2 and 3 are used to create a pentagonal intuitionistic fuzzy decision matrix, as shown in Table 7.

Using Eqs. 4 and 5, pentagonal intuitionistic fuzzy normalized and weighted normalized decision matrix are generated, shown in Tables 8 and 9. Using Eqs. 6 and 7 the pentagonal intuitionistic fuzzy positive ideal solution (PIS) and negative ideal solution (NIS) are determined. Using Eqs. 8 and 9, the separation measure from PIS and NIS are determined. Using Eq. 10, relative closeness is calculated. Finally, ranks are given in ascending order, and the evaluation table for TOPSIS is given in Table 10.

Table 5 Rating of weights of the criteria by decision makers

Criteria	Dk_1	Dk_2
Ct_1	0.2	0.3
Ct_2	0.4	0.2
Ct_3	0.3	0.4
Ct_4	0.1	0.1

Table 6 Aggregated weights of criteria

Ct_1	Ct_2	Ct_3	Ct_4
0.25	0.3	0.35	0.1

Table 7 Intuitionistic fuzzy decision matrix

	Ct_1	Ct_2	Ct_3	Ct_4
Ar_1	⟨(2, 3, 4, 5, 6); (1, 2, 4, 6, 7)⟩	⟨(4, 5, 6, 7, 8); (3, 4, 5, 6, 8, 9)⟩	⟨(1.5, 2.5, 3.5, 4.5, 5.5); (0.5, 1.5, 3.5, 5.5, 6.5)⟩	⟨(1, 2, 3, 4, 5); (0, 1, 3, 5, 6)⟩
Ar_2	⟨(3, 4, 5, 6, 7); (2, 3, 5, 7, 8)⟩	⟨(3.5, 4.5, 5.5, 6.5, 7.5); (2.5, 3.5, 5.5, 7.5, 8.5)⟩	⟨(1, 2, 3, 4, 5); (0, 1, 3, 5, 6)⟩	⟨(2, 3, 4, 5, 6); (1, 2, 4, 6, 7)⟩
Ar_3	⟨(2.5, 3.5, 4.5, 5.5, 6.5); (1.5, 2.5, 4.5, 6.5, 7.5)⟩	⟨(3, 4, 5, 6, 7); (2, 3, 5, 7, 8)⟩	⟨(1.5, 2.5, 3.5, 4.5, 5.5); (0.5, 1.5, 3.5, 5.5, 6.5)⟩	⟨(3.5, 4.5, 5.5, 6.5, 7.5); (2.5, 3.5, 5.5, 7.5, 8.5)⟩
Ar_4	⟨(3.5, 4.5, 5.5, 6.5, 7.5); (2.5, 3.5, 5.5, 7.5, 8.5)⟩	⟨(2.5, 3.5, 4.5, 5.5, 6.5); (1.5, 2.5, 4.5, 6.5, 7.5)⟩	⟨(3, 4, 5, 6, 7); (2, 3, 5, 7, 8)⟩	⟨(1, 2, 3, 4, 5); (0, 1, 3, 5, 6)⟩
Ar_5	⟨(4, 5, 6, 7, 8); (3, 4, 6, 8, 9)⟩	⟨(2, 3, 4, 5, 6); (1, 2, 4, 6, 7)⟩	⟨(2, 3, 4, 5, 6); (1, 2, 4, 6, 7)⟩	⟨(2, 3, 4, 5, 6); (1, 2, 4, 6, 7)⟩

By using Eqs. 11 and 12, Using the decision matrix 7, find the fuzzy good and worst values.

$$x_1^+ = \langle (4, 5, 6, 7, 8); (3, 4, 6, 8, 9) \rangle$$

$$x_2^+ = \langle (4, 5, 6, 7, 8); (3, 4, 6, 8, 9) \rangle$$

$$x_3^+ = \langle (3, 4, 5, 6, 7); (2, 3, 5, 7, 8) \rangle$$

$$x_4^+ = \langle (3.5, 4.5, 5.5, 6.5, 7.5); (2.5, 3.5, 5.5, 7.5, 8.5) \rangle$$

$$x_1^- = \langle (2, 3, 4, 5, 6); (1, 2, 4, 6, 7) \rangle$$

$$x_2^- = \langle (2, 3, 4, 5, 6); (1, 2, 4, 6, 7) \rangle$$

$$x_3^- = \langle (1, 2, 3, 4, 5); (0, 1, 3, 5, 6) \rangle$$

$$x_4^- = \langle (1, 2, 3, 4, 5); (0, 1, 3, 5, 6) \rangle$$

Table 8 Intuitionistic fuzzy normalized decision matrix

	Ct_1	Ct_2	Ct_3	Ct_4
Ar_1	⟨(0.13, 0.22, 0.35, 0.55, 0.87); (0.06, 0.13, 0.35, 0.87, 1.47)⟩	⟨(0.25, 0.37, 0.53, 0.71, 1.61); (0.17, 0.25, 0.53, 1.61, 1.90)⟩	⟨(0.11, 0.23, 0.40, 0.70, 1.28); (0.03, 0.11, 0.40, 1.28, 2.78)⟩	⟨(0.07, 0.18, 0.33, 0.59, 1.06); (0, 0.07, 0.33, 1.06, 2.09)⟩
Ar_2	⟨(0.19, 0.29, 0.44, 0.66, 1.01); (0.11, 0.19, 0.44, 1.01, 1.68)⟩	⟨(0.22, 0.33, 0.49, 0.71, 1.08); (0.14, 0.22, 0.49, 1.08, 1.79)⟩	⟨(0.07, 0.18, 0.35, 0.62, 1.16); (0, 0.07, 0.35, 1.16, 2.56)⟩	⟨(0.15, 0.27, 0.44, 0.73, 1.27); (0.06, 0.15, 0.44, 1.27, 2.44)⟩
Ar_3	⟨(0.16, 0.26, 0.40, 0.60, 0.94); (0.08, 0.16, 0.40, 0.94, 1.58)⟩	⟨(0.19, 0.29, 0.44, 0.66, 1.01); (0.11, 0.19, 0.44, 1.01, 1.68)⟩	⟨(0.11, 0.23, 0.40, 0.70, 1.28); (0.03, 0.11, 0.40, 1.28, 2.78)⟩	⟨(0.26, 0.40, 0.61, 0.95, 1.59); (0.16, 0.26, 0.61, 1.59, 2.96)⟩
Ar_4	⟨(0.22, 0.33, 0.49, 0.71, 1.08); (0.14, 0.22, 0.49, 1.08, 1.79)⟩	⟨(0.16, 0.26, 0.40, 0.60, 0.94); (0.08, 0.16, 0.40, 0.94, 1.58)⟩	⟨(0.23, 0.37, 0.58, 0.93, 1.63); (0.13, 0.23, 0.58, 1.63, 3.42)⟩	⟨(0.07, 0.18, 0.33, 0.59, 1.06); (0, 0.07, 0.33, 1.06, 2.09)⟩
Ar_5	⟨(0.25, 0.37, 0.53, 0.71, 1.61); (0.17, 0.25, 0.53, 1.61, 1.90)⟩	⟨(0.13, 0.22, 0.35, 0.55, 0.87); (0.06, 0.13, 0.35, 0.87, 1.47)⟩	⟨(0.15, 0.27, 0.46, 0.77, 1.39); (0.06, 0.15, 0.46, 1.39, 2.99)⟩	⟨(0.15, 0.27, 0.44, 0.73, 1.27); (0.06, 0.15, 0.44, 1.27, 2.44)⟩

Table 9 Intuitionistic fuzzy weighted normalized decision matrix

	Ct_1	Ct_2	Ct_3	Ct_4
Ar_1	⟨(0.03, 0.05, 0.08, 0.14, 0.22); (0.02, 0.03, 0.08, 0.22, 0.36)⟩	⟨(0.07, 0.11, 0.16, 0.21, 0.48); (0.05, 0.07, 0.16, 0.48, 0.57)⟩	⟨(0.04, 0.08, 0.14, 0.44, 0.44); (0.01, 0.04, 0.14, 0.44, 0.97)⟩	⟨(0.007, 0.01, 0.03, 0.06, 0.10); (0, 0.007, 0.03, 0.10, 0.20)⟩
Ar_2	⟨(0.04, 0.07, 0.11, 0.16, 0.25); (0.02, 0.04, 0.11, 0.25, 0.42)⟩	⟨(0.06, 0.09, 0.14, 0.21, 0.32); (0.04, 0.06, 0.14, 0.32, 0.53)⟩	⟨(0.02, 0.06, 0.12, 0.21, 0.40); (0, 0.02, 0.12, 0.40, 0.89)⟩	⟨(0.01, 0.02, 0.04, 0.07, 0.12); (0.006, 0.01, 0.04, 0.12, 0.24)⟩
Ar_3	⟨(0.04, 0.06, 0.1, 0.15, 0.23); (0.02, 0.04, 0.1, 0.23, 0.39)⟩	⟨(0.005, 0.08, 0.13, 0.19, 0.30); (0.03, 0.05, 0.13, 0.30, 0.50)⟩	⟨(0.04, 0.08, 0.14, 0.24, 0.44); (0.01, 0.04, 0.14, 0.44, 0.97)⟩	⟨(0.02, 0.04, 0.06, 0.09, 0.15); (0.01, 0.02, 0.06, 0.15, 0.29)⟩
Ar_4	⟨(0.05, 0.08, 0.12, 0.17, 0.27); (0.03, 0.05, 0.12, 0.27, 0.44)⟩	⟨(0.04, 0.07, 0.12, 0.18, 0.28); (0.02, 0.04, 0.12, 0.w, 0.47)⟩	⟨(0.08, 0.13, 0.20, 0.32, 0.57); (0.04, 0.08, 0.20, 0.57, 1.19)⟩	⟨(0.007, 0.01, 0.03, 0.06, 0.10); (0, 0.007, 0.03, 0.10, 0.20)⟩
Ar_5	⟨(0.06, 0.09, 0.13, 0.17, 0.40); (0.04, 0.06, 0.13, 0.40, 0.47)⟩	⟨(0.04, 0.06, 0.10, 0.16, 0.26); (0.02, 0.04, 0.10, 0.26, 0.44)⟩	⟨(0.05, 0.09, 0.16, 0.27, 0.48); (0.02, 0.05, 0.16, 0.48, 1.04)⟩	⟨(0.01, 0.02, 0.04, 0.07, 0.12); (0.006, 0.01, 0.04, 0.12, 0.24)⟩

Table 10 Evaluation table for TOPSIS

	m^+	m^-	Cl	Rank
Ar_1	0.1285	0.1047	0.5510	3
Ar_2	0.1350	0.0564	0.7053	5
Ar_3	0.1349	0.0622	0.6844	4
Ar_4	0.1026	0.1151	0.4712	1
Ar_5	0.1205	0.1034	0.5381	2

Table 11 Evaluation table for VIKOR

	S_i	Rank	R_i	Rank	Q_i	Rank
Ar_1	0.5468	5	0.2500	2	0.4579	2
Ar_2	0.6100	4	0.3500	5	1.0000	5
Ar_3	0.6000	3	0.2625	3	0.6222	3
Ar_4	0.3875	1	0.2250	1	0.0000	1
Ar_5	0.5350	2	0.3000	4	0.6278	4

The value of S_i, R_i and Q_i is calculated using Eqs. 13, 14 and 15. Sort S_i, R_i and Q_i and arrange in ascending order and generate three ranking list. Here, ν is taken as 0.5. The evaluation table for VIKOR is given in Table 11.

The following calculation shows that condition 1 (acceptable advantage) is satisfied.

$$Q(\mathrm{Ar}_1^{(2)}) - Q(\mathrm{Ar}_4^{(1)}) \geq \frac{1}{m-1}$$

$$0.4579 - 0 \geq \frac{1}{5-1}$$

$$0.4579 > 0.25$$

Condition 2 (acceptable stability) has been met as well. By S_i and, R_i the alternative $\mathrm{Ar}_4^{(1)}$ is also the first rated.

6 Conclusion

Using a new distance measure of IFN, this paper expanded two MCDM algorithms, TOPSIS and VIKOR. A new pentagonal intuitionistic fuzzy number distance metric has been presented. This distance metric is used in TOPSIS and VIKOR. When the findings were compared, it was found that both systems rated the identical alternative first. Because it is the top ranked alternative according to the TOPSIS technique, it

is the best option in terms of the ranking index. The fact that the identical alternative is ranked highest by the VIKOR technique indicates that it is the most similar to the optimal answer. However, because to its elegance and computational simplicity, we have determined that the VIKOR technique is the best.

References

1. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353
2. Atanassov KT (1999) Intuitionistic fuzzy sets. Physica-Verlag, Heidelberg, New York
3. Boran FE (2011) An integrated intuitionistic fuzzy multi criteria decision making method for facility location selection. Math Comput Appl 16:487–496
4. Parveen N, Kamble PN (2021) An extension of TOPSIS for group decision making in intuitionistic fuzzy environment. Math Found Comput 4:61–71 (American Institute of Mathematical Sciences). https://doi.org/10.3934/mfc.2021002
5. Izadikhah M (2012) Group decision making process for supplier selection with TOPSIS method under interval-valued intuitionistic fuzzy numbers. Adv Fuzzy Syst 1–14
6. Parveen N, Kamble PN (2020) Decision making problem using fuzzy TOPSIS method with hexagonal fuzzy number. In: Advances in intelligent systems and computing, vol 1025. Springer Nature, Singapore, pp 421–430
7. Opricovic S, Teeng GH (2007) Extended VIKOR method in comparison with outranking methods. Eur J Oper Res 178:514–529
8. Luo X, Wang X (2017) Extended VIKOR method for intuitionistic fuzzy multiattribute decision making based on a new distance measure. Math Probl Eng 1–16
9. Kamble PN, Parveen N (2018) An application of integrated fuzzy AHP and fuzzy TOPSIS method for staff selection. J Comput Math Sci 9(9):1161–1169
10. Hwang CL, Yoon K (1981) Multiple attribute decision making method. Springer, Berlin, pp 115–140
11. Opricovic S (1998) Multi-criteria optimization of civil engineering systems. Faculty of Civil Engineering, Belgrade
12. Mahapatra GS, Roy TK (2013) Intuitionistic fuzzy number and its arithmetic operation with application on system failure. J Uncertain Syst 7:92–107
13. Ponnivalavan K, Pathinathan T (2015) Intuitionistic pentagonal fuzzy number. ARPN J Eng Appl Sci 10:5446–5450

Chapter 37
Object Recognition Using Semantic Segmentation

Rosepreet Kaur Bhogal and V. Devendran

1 Introduction

Computer vision is a multi-disciplinary domain. Computer vision provide us with the structure, for automatic extraction, analysis, and comprehension from a single image or image sequence. Recognizing scenes from images or videos is one of the most real-world challenging application of computer vision in various spheres like industry, academia, security agencies, consumer agencies, and general-purpose, etc. Segmentation is one of the challenging tasks and it can be the first step in recognition of any task under preprocessing technique. The benefits of visual or group perception have been one of the most important computer vision problems used in various applications such as automatic driving [1] human machine communication, computer graphics [2], age-old search engines, and augmented to new reality.

The semantic segmentation can handle all deep learning architecture that has been used for classification. Finding objects from the image or videos can be the first step for classification. This not only provide us with the classes but also other information which may be useful for the recognition of objects with spatial information. Its goal is to label every pixel from the images or videos [3]. Per-pixel classification will be an effective and useful technique for various applications [1–3].

Segmentation is one of the most important techniques for image or video processing. Semantic segmentation served the front-end processing for computer vision applications. There is a different existing technique that relies on a database, a simple approach based on oriented pattern besides the use of color of texture [4]. Some methods, which are trying to train from end-to-end CNN networks, for multiple groups where co-segmentation can be possible to obtain the foreground of

R. K. Bhogal (✉)
School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, India
e-mail: rosepreetkaur12@gmail.com

V. Devendran
School of Computer Science and Engineering, Lovely Professional University, Phagwara, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_38

each group. It can be used as two subnetworks for region extraction and segmentation [5]. Segmentation can also use in hashing-based multimedia method retrieval due to the advantage of less cost and more efficiency [6]. The majority of current state of art methods are having high computation resources, so can be used for industry-oriented applications for scene understanding, etc. [7]. Semantic segmentation can be use on urban scenes also. It is giving suffice results, if used with insufficient labeled data [8]. Automatic digitizing the maps for both academia and industry instead of manual interpretation of scene using convolutional neural network [9]. There is a need to improve the existing method techniques, so graphical approach-based model has been proposed that applies the information of several functions and it also have done the comparison based on execution time with other methods [10]. Segmentation involves the object as well as pixel segmentation. To do pixel-wise classification, it is required to concentrate on high and low-level features. To overcome this, Seed Picking Crossover Optimization (SPCO) is proposed to use the training Conditional Random Field (CRF) by selecting the appropriate features to classify the object with high accuracy [11]. For building a neural deep network with the newly designed Spatio-temporal continuous (STC) semantic segmentation of video has done by Chen et al. [12]. This proposed method is for video-based segmentation. In the practical scenario like the robotic field, semantic segmentation is playing an important role. In the case of robots, if they have to make decisions based on complex images then we can improve system using semantic segmentation which can help for taking complex visual decisions [13]. To perform the image classification function, sequential maps from FCN are compiled to enhance segmentation effect [14]. To design a system that can easily identify the prohibited items given as [15]. The network can be design by combining two subnetworks which can focus on attention of task [16].

There is always requirement for the objects analysis either in video or images. This paper has been given the network which used to identify the objects from video frames. The object detection can be done using deep learning networks and that can be important steps under preprocessing of videos or images. This object recognition helps to understand the required object is present in the video or not. The paper divided into various Sects. 2 is detail explanation of used dataset for training and validation. In Sect. 3 includes the research methodology, further Sects. 4 and 5 having experimental results and conclusion, respectively.

2 Dataset

The video-based object analysis database which is named as Cambridge driving Labeled Video Database (CamVid) database has been used in this paper. The database videos have been recorded with a camera mounted inside the car. Using a high-resolution digital camera 3CCD Panasonic HVX200, it captures 960×720 pixels at 30 frames per second (fps). The CamVid website contains four HD video streams. Their names and times (in minutes: seconds) are: 0001TP (8:16), 0006R0 (3:59), 0016E5 (6:19), and Seq05VD (3:40). Full time then 22:14. Sequences with a variety

of class situations such as cars, pedestrians, and cyclists. It has been identified 32 classes from video sequences [17]. In this, the different R, G, B color values have been used to labeled spatial information as given in Fig. 1.

The pixel-wise labeling with RGB values allows to accurately learn about the appearance and shape. The sample one frame of video with the corresponding pixel-wise labeled image as shown in Fig. 2a, b, respectively. Figure 3 indicates the video sequences duration. The number of labeled frames has been mentioned corresponding to each video with the duration of each video. If we sum up the duration of all videos, it comes out to be 10 min. If we see each frame in videos, then it has several objects that can be labeled as per Fig. 1 [17].

CamVid database can be used for various applications. The researcher can design various algorithms or network by using this database which can be used analysis of objects using semantic segmentation. The algorithm can be addressed for object recognition, pedestrian detection, and object label propagation [17].

Fig. 1 List of the 32 object class names and their corresponding colors used for labeling [17]

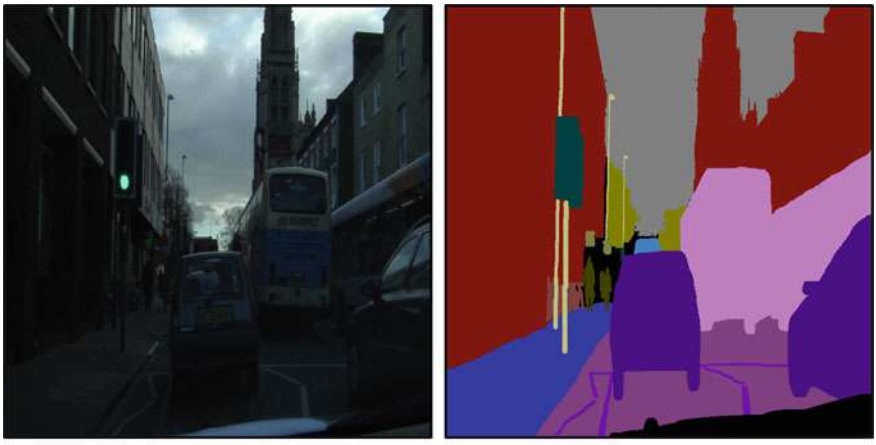


Fig. 2 **a** Sample of the captured frame from CamVid, **b** sample of pixel-wise labeled frame CamVid

labeled sequence name	original sequence name	frame rate in fps	number of labeled frames	corresponding duration
0001TP_L	0001TP	1	124	2:04
0016E5_1Hz_L	0016E5	1	204	3:24
0016E5_15Hz_L	0016E5	15	101	0:06
0006R0_L	0006R0	1	101	1:41
Seq05VD_L	Seq05VD	1	171	2:51
	Total		701	10:06

Fig. 3 Labeled videos with name, frame per second, number of frames and duration [17]

3 Research Methodology

3.1 Image Preprocessing

Every deep neural network required a designated size of image according to the input layer of the network. In this work, the input size of the image is $720 \times 960 \times 3$.

3.2 Convolutional Neural Network for Segmentation

Image segmentation is a problem for remote computer viewing. A few algorithms are designed to solve this task, such as Watershed algorithm, Image thresholding, *K*-means clustering, Graph partitioning ways. Many methods have been developed for semantic segmentation, but the deep neural network had given the best result among all. The deep neural network architecture has various layers such as convolutional layers, activation layers, batch normalization layer, max or min pooling layer. To generate desired output, we are required to have information about the spatial component of the input images. To accomplish this task, we required a fully convolutional network. Convolutional layers combined with sample layers produce an optimal weight that contains high-level of details. This is called the encoder-decoder structure. Where the layers below the input are part of the encoder and the layers that make up the sample are part of the decoder [18]. In this work, the DeepLab v3+ network was restored as a convolutional neural network for semantic image separation. The network uses encoder-decoder construction, integrated integration, and skips connections to image segments. The example of encoder and decoder is given as follows in Fig. 4.

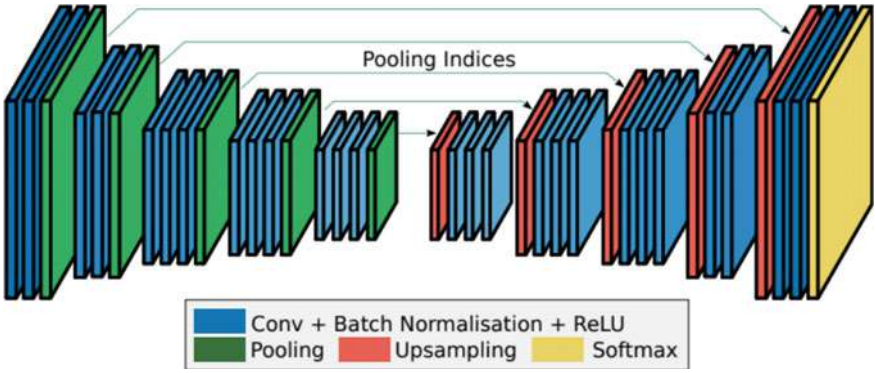


Fig. 4 Encoder-decoder architecture [18]

When a model is trained for semantic segmentation function, the encoder sets a tensor that contains details about objects, their shape and size. The decoder captures this information and generates segmentation maps [19].

4 Experimental Work

The semantic segmentation network is classifying every pixel from an image/video. In this work, the deep learning network has been created which is called as Deeplabv3+ network with weights initialized from a pre-trained Resnet-18 network. Resnet-18 is one of the convolutional networks and is also called a deep architecture network [20]. It is being trained by millions of images from the ImageNet database. It can classify more than 1000 categories or have classes. We can use deep architecture for the classification of any new classes with help of the transfer learning concept. To do the training, train Deeplabv3+ which is one type of convolutional neural network (CNN) designed for semantic image segmentation.

There are various steps involves to carry out the object recognition using semantic segmentation. The first most important step is to finalize the neural network which is going to be used for this task. Resnet18 has been finalized and install pertained network in programming software. It is suited for the application where required limited processing resources. After loading the dataset of CamVid images and CamVid pixel labeled image, then partition for training, testing, and validation data is done. Having pre-processed dataset is a very important with the tuned deep neural network. For recognition, create a DeepLabv3+ network based on Resnet18. Choosing between a networks for your application requires empirical analysis and tuning of hyper parameter [21]. The network has always been analyzed after training, validation, and testing the data many times. In any network, how many deep layers or overall are using that will make network worth to give more recognition rate. Various hyper parameters are playing a role called training options.

Figure 6, is about the training progress plot which is showing the progress that how accuracy is varying with each epoch. The blue-colored solid line is showing that how training data is trained per epoch. And, a black-colored dotted line is showing that how validation testing being processed. The algorithm has been used for training the stochastic gradient descent with momentum rate. The various hyper parameters have been used in this for tuning the network. The learning rate has been used at 0.0003, and it is being reduced after every epoch. This help network to train and find the solution to be optimum. The network has been tested with validation data as well as tested separated with test data. If we see Fig. 5, in the starting the accuracy is 10% during the training within the first epoch only, it accuracy reached 80%. In the progress plot of training the final validation accuracy slowing increasing and end at 90.2808% after epoch 10. Generally, this type of network can prefer because when network is trained, the training stops early when validation accuracy converged. This can be considering as advantage for any network and help it from overfitting problems during the training progress.

There are three type of situations, which can arise while using deep neural network. It may be possible that network can over fit, under fit or perfect fit on the training dataset and validation dataset. To check the network is working perfectly, it is needed to analyze the network based on training loss and validation loss per iteration. Figure 7, is giving information on how is the loss is varying on subsequent iterations. If training loss is higher than the validation loss, then the network is facing underfitting.

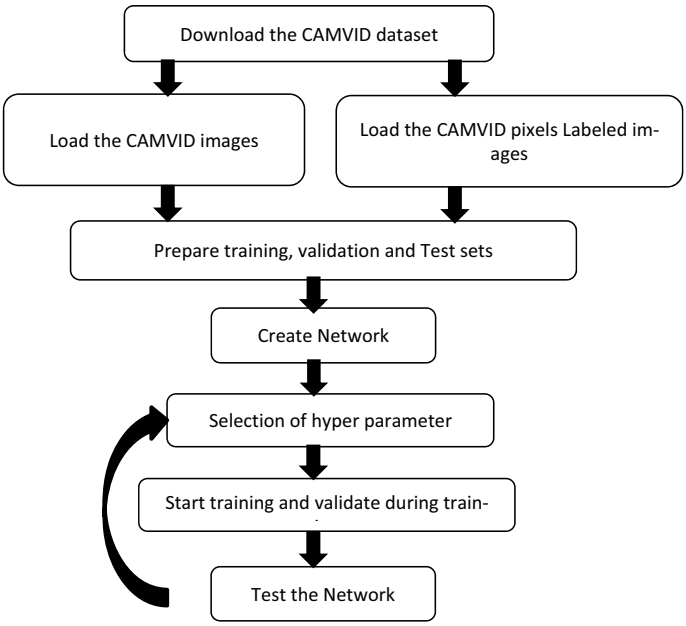


Fig. 5 Research methodology used in the method

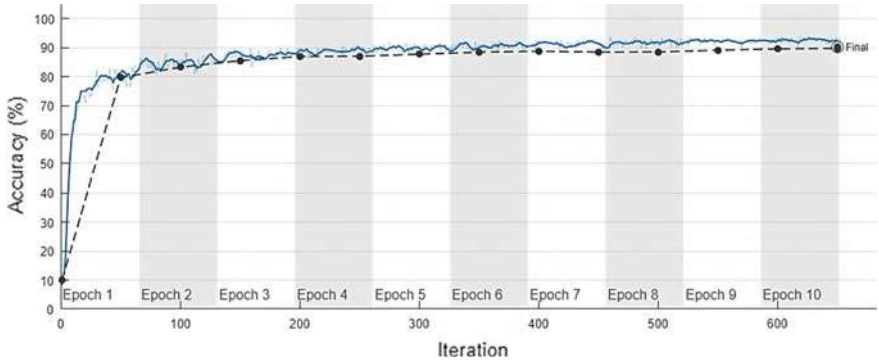


Fig. 6 Training progress plot (accuracy vs. iterations)

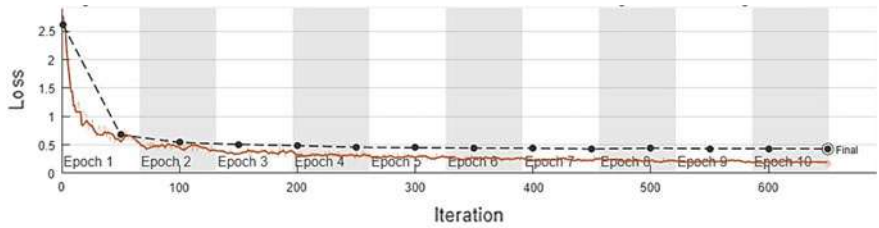


Fig. 7 Training progress plot (loss vs. iterations)

If validation loss is more than the training loss, then the network is having overfitting problem. And, third if validation loss and training loss are almost equal then the chances are high that the network worked well. In Fig. 7, the training loss (shown as a red solid line) is nearly 2.5 on initial iterations, after that it reduces gradually within the first epoch only. Similarly, validation loss (shown as a black dotted line) is roughly the same as training loss. It is observed that loss is reducing after epoch 2, and the final validation loss achieved in this is 0.4224 after epoch 10 which is close to 600th iteration.

The recognition rate is a very important factor to understand that how much our system can verify the test data correctly. Table 1 shows the recognition rate of 11 classes which are separately evaluated for test data only. There are 140 frames for RGB and pixel label data each. In Table 1, the left first column is called as predicted labels, and the first row is called as the target name of the classes. If we see the class name "sky", 95.05% of frames from test data have been correctly identified as "sky". Similarly, 84.3% as "building", 71.71% as "pole", 95.32% as "road", 88.1% as "pavement", 86.19% as "tree", 75.36% as "signsymbol", 76.44% as "fence", 89.89% as "car", 84.45% as "pedestrian", 86.5% as "bicyclist" (green colored box text shows the maximum accuracy for each target class). In Table 1, the accuracy variation of 5 Color space has been shown from 0 to 1.

Table 1 Recognition rate per classes with testing dataset

Observation class versus target class	Sky	Building	Pole	Road	Pavement	Tree	SignSymbol	Fence	Car	Pedestrian	Bicyclist
Sky	0.9505	0.0043	0.0085	0	0	0.0352	0.0011	0	0.0004	0	0
Building	0.0086	0.843	0.0477	0.0002	0.0125	0.0219	0.0355	0.0047	0.01	0.0137	0.0021
Pole	0.0191	0.0891	0.7171	0.0017	0.0166	0.0542	0.0538	0.0086	0.0137	0.024	0.0023
Road	0	0.0002	0.0015	0.9532	0.0319	0.0003	0.0001	0	0.0093	0.001	0.0025
Pavement	0	0.0063	0.015	0.076	0.881	0.0014	0.0005	0.0019	0.0053	0.0093	0.0033
Tree	0.0435	0.0295	0.0328	0.0007	0.0017	0.8619	0.0101	0.0111	0.0045	0.0035	0.0007
SignSymbol	0.0008	0.1035	0.0433	0.001	0.0042	0.0483	0.7536	0.0038	0.0312	0.0097	0.0005
Fence	0	0.0588	0.0676	0.0005	0.0176	0.0412	0.0037	0.7644	0.0171	0.0277	0.0013
Car	0.0012	0.0254	0.0172	0.0098	0.0053	0.0044	0.0074	0.0054	0.8989	0.02	0.0049
Pedestrian	0	0.0455	0.0506	0.0016	0.0104	0.0041	0.0045	0.0053	0.0186	0.8445	0.0151
Bicyclist	0	0.0131	0.0048	0.008	0.0059	0.0078	0.0031	0.0002	0.0129	0.0792	0.865

The output of the network has been shown in Fig. 8. Where the image frame is shown in Fig. 8a, c are RGB frames from the CamVid dataset and corresponding semantic segmented frames are given in Fig. 8b, d. If we observe the semantic segmented frame, the semantic segmentation has been done well on the roads, sky, and building as compared to pedestrians and other objects. Although, the semantic segmentation is going well on the part of the frame, where area coverage is more and less complex. Otherwise, where an object is small and more complex. The semantic segmentation results are not that much good.

Creating a best model for any data and weights that contain any function can be difficult. The important thing is that we can compare our model with different parameters. This overall model can be concluded using three evaluation metrics. First, accuracy is ratio to the predicted and target class. Second, Intersection of Union (IoU) is an important measure of accuracy to be followed when collecting human annotations. It is used to give a number between 0 and 1, which specifies the value of the interval between the predicted box and the world that binds the truth. If it is zero, then there is no excess space between the boxes. Also, if you are closer to 1 it means that the junction of the boxes is similar to their overlap which indicates

Fig. 8 The output of test data, where **a**, **c** are RGB frame from the CamVid dataset, **b**, **d** are corresponding segmented images

Table 2 Class wise accuracy, IoU and mean scores values

Class | Parameters	Accuracy	IoU	Mean scores
Sky	0.9505	0.9088	0.9044
Building	0.843	0.8118	0.6711
Pole	0.7171	0.2629	0.6034
Road	0.9532	0.9337	0.8293
Pavement	0.881	0.7411	0.7551
Tree	0.8619	0.773	0.7234
SignSymbol	0.7536	0.4112	0.5286
Fence	0.7644	0.5932	0.5398
Car	0.8989	0.791	0.7543
Pedestrian	0.8445	0.4288	0.596
Bicyclist	0.865	0.6222	0.5634

that they are completely spaced. Third, scores are kinds of values which are giving information to decide on the class to which the observation belongs. The maximum score per class has been given in Table 2. The more the value of the score, the more chances are there that it belongs to a class. All metrics are given in Table 2. The maximum accuracy is of class name "sky" and minimum is of class name as "pole" and same is the case when looking the result of IoU.

5 Conclusion

This paper presented object recognition using semantic segmentation. Based on scene classification into 11 classes, we have used a deep neural network as a model to do the segmentation of each frame of the video. The CamVid dataset has been used to train and test the DeepLab v3+ network with resnet18 [20]. There are various challenges during this research. This model is not working well in smaller objects in the frame. However, it will be an ideal model for bigger objects in the frame. The methodology of work has been given in the section under research methodology and shown in Fig. 4. The work has been concluded based on three parameters which are accuracy, IoU, and scores. The maximum accuracy is given per class has explained in the section under experimental result.

Semantic segmentation can be used for various applications like autonomous cars, action recognition, medical field, etc. To detect the object is available is one of the most important initial steps in any recognition or detection. In the future scope, this model can use by focusing on smaller objects in image/videos. If smaller objects are also segmented properly, then this model will be preference for semantic segmentation.

References

1. Ess A, Müller T, Grabner H, Van Gool L (2009) Segmentation-based urban traffic scene understanding. In: British machine vision conference BMVC 2009, proceedings, pp 1–11
2. Yoon Y, Jeon HG, Yoo D, Lee JY, Kweon IS (2015) Learning a deep convolutional network for light-field image super-resolution. In: Proceedings of the IEEE international conference on computer vision, vol 2015 Feb, pp 57–65
3. Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, Garcia-Rodriguez J (2018) A survey on deep learning techniques for image and video semantic segmentation. Appl Soft Comput J 70:41–65
4. Ong HT, Ma KK (2011) Semantic image segmentation using oriented pattern analysis. In: ICICS 2011—8th International conference on information, communications & signal processing, pp 1–4
5. Luo K, Meng F, Wu Q, Li H (2018) Weakly supervised semantic segmentation by multiple group cosegmentation. In: VCIP 2018—IEEE International conference on visual communications and image processing, pp 1–4
6. Zhang M, Liu S, Zhu Y, Bai Z, Lin J (2020) Seg-Hashnet: semantic segmentation based unsupervised hashing. Institute of Big Data Technology, Shenzhen Graduate School, Peking University School of Software & Microelectronics, Peking University, pp 12–16
7. Lyu H, Fu H, Hu X, Liu L (2019) ESNet: edge-based segmentation network for real-time semantic segmentation in traffic scenes. In: Proceedings—International conference on image processing ICIP, vol 2019 Sept, pp 1855–1859
8. Zheng Q, Chen J, Huang P, Hu R (2019) Urban scene semantic segmentation with insufficient labeled data. China Commun. 16(11):212–221
9. Guo Z et al (2018) Semantic segmentation for urban planning maps based on U-net. In: International geoscience and remote sensing symposium, vol 2018 July, pp 6187–6190
10. Noormohamadi N, Adibi P, Ehsani SMS (2018) Semantic image segmentation using an improved hierarchical graphical model. IET Image Process 12(11):1943–1950
11. (2020) Seed picking crossover optimisation algorithm for semantic segmentation from images. IET Image Process 14(11):2503–2511
12. Chen X, Wu A, Han Y (2019) Capturing the spatio-temporal continuity for video semantic segmentation. IET Image Process 13(14):2813–2820
13. Huang L, He M, Tan C, Du J, Li G, Yu H (2020) Jointly network image processing: multi-task image semantic segmentation of indoor scene based on CNN. IET Image Process 14(15):3689–3697
14. Li H, Qian X, Li W (2017) Image semantic segmentation based on fully convolutional neural network and CRF. In: Yuan H, Geng J, Bian F (eds) Communications in computer and information science, vol 698. Springer Singapore, Singapore, pp 245–250
15. An J, Zhang H, Zhu Y, Yang J (2019) Semantic segmentation for prohibited items in baggage inspection, vol 2, pp 495–505
16. Li G, Li L, Zhang J, BiAttnNet: bilateral attention for improving real-time semantic segmentation. IEEE Signal Process Lett. https://doi.org/10.1109/LSP.2021.3124186
17. Brostow GJ, Fauqueur J, Cipolla R (2009) Semantic object classes in video: a high-definition ground truth database. Pattern Recognit Lett 30(2):88–97
18. Syed A, Morris BT (2019) SSeg-LSTM: semantic scene segmentation for trajectory prediction. In: 2019 IEEE Intelligent vehicles symposium, no. IV. IEEE Conference Publication, pp 2504–2509
19. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics), vol 11211 LNCS, pp 833–851

20. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE computer society conference on computer vision pattern recognition, vol 2016 Dec, pp 770–778
21. Ahmed A, Yu K, Xu W, Gong Y, Xing E (2008) Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. In: Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics), vol 5304 LNCS, no Part 3, pp 69–82

Chapter 38
Smart Grid Communication Network Reliability Assessment Using Graphical Computational Model

Swati Sharda, Kapil Sharma, and Mukhtiar Singh

1 Introduction

The power system is one of the most important infrastructures or critical infrastructure (CI), as it is reliant on almost all other infrastructures [1]. The reliability of information and communication technology (ICT) plays major role in the smooth working of smart grid. Smart grid is the smarter version of power grid which involves the combination of traditional power grid and new ICT. Hence, to construct a cyber-physical system (CPS), i.e. smart grid, ICT is a crucial addition to the conventional power system modules, allowing for improved flexibility and functionality. Although ICT systems improve functionality, they also lead to failures, such as unseen failures in protection systems, as seen in recent power outages. It also brings with it new challenges, including cyber-attacks.

The functionalities of cyber-components have been assumed perfectly reliable in the conventional reliability assessment phase. For conventional power grids where there was only minimal cyber-physical interdependence, this assumption might be rational. However, modern power grid reliability is more reliant on the availability of cyber-enabled functions with the growing deployment of cyber infrastructure. To achieve more practical results, it is important to include cyber-malfunctions in the assessment of power system reliability. The impact of cyber-malfunctions on the reliability of composite power systems is investigated in this subject, which is referred to as 'cyber-physical reliability' [2].

The efficiency of the power distribution system is calculated by indices such as the average system interruption period index (SAIDI) and the average system interruption frequency index (SAIFI) [3]. Using analytical or modelling methods, these indicators may be evaluated.

S. Sharda (✉) · K. Sharma · M. Singh
Delhi Technological University, New Delhi, Delhi 110042, India
e-mail: swatisharda2807@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_39

Analytical methods use mathematical models that are typically large and inefficient, but they are used to analyse large-scale systems of complex schemes. Simulation methods based on the Monte Carlo method, on the other hand, take into account the actual procedure and random actions of network components. It takes into account component failure history as well as maintenance time data [4]. Embedded or non-component dependent faults, such as cold load pickup (CLPU), are not typically included in the estimates of these indices. As a consequence, these measurements cannot accurately represent real device behaviour.

The Markovian model [5], reliability block diagram (RBD) [6, 7], fault-tree analysis [8], failure propagation studies [9], and state-mapping techniques [10] are some of the other works relevant to evaluating the direct cyber-to-power effect on system reliability. The majority of previous works necessitate a large amount of computational effort, and the computational time increases exponentially with the scale of the system under investigation.

The graph-theoretical approach to the smart grid is a solution to the above challenges. The aim of graphical approaches is to classify the nodes and edges that are most vulnerable to attacks and failures due to human error. The graphical reliability model takes into account both electrical (transmission line impedance) and reliability indices (probability of failure in network components). Reliability graphs were encoded to binary decision diagrams (BDD) to reduce storage and computation complexity [11]. Another valuable method for reliability analysis is Bayesian networks.

There are several other facets and factors that affect the overall network's reliability. Many techniques for determining the importance of a part in a network have been developed. An analytical method based on state matrix which including different uncertainties of PHEV has been introduced in [12]. The criticality reliability importance to quantify the degree of impact of key component failures on system failure, mainly for fault diagnosis, has been suggested in [13]. The redundancy value metric seeks to determine the incremental reliability growth of a system when an additional redundant variable is introduced. This metric is very useful in deciding the best degree of redundancy in reliability design [14]. The improvement potential (IP) metric is introduced to estimate the change in system efficiency when a part of a system is perfectly restored after failure [15]. Risk mitigation and risk achievement are the most widely used risk value indicators, which imply the contribution of a particular component to the overall risk [16]. The Bayesian reliability value method was proposed to determine the weakest component that is most likely to be the cause of system failure [17].

However, since these approaches rely heavily on cause-and-effect relationships, they are unsuccessful in assessing the overall reliability of large interconnected ICT-EP systems and in examining the role of cyber infrastructure. As a consequence, a scalable but realistic tool for assessing the risk of interconnected system failure due to ICT system reliability and identifying essential cyber-components is needed.

This paper provides a threefold contribution. To determine the structural significance of the ICT components within a Supervisory Control and Data Acquisition (SCADA) system, we first propose a graphical model of our given system. Second, a new metric named network accuracy indicator (NAI) is developed to quantify the sys-

tem's reliability. The network reliability deterioration worth (NRDW) index is used to rank the criticality of each node in the corresponding graphical model providing system reliability. Finally, simulations are performed on an IEEE 14 bus test case to prove that the proposed approach correctly tests the system's reliability parameters.

The rest of the paper is set out as follows. The reliability of each path within a network is quantified in Sect. 2 using the system's graphical model, and the method for calculating the NAI index is presented. Mathematical indices to measure relative structural significance are suggested in Sect. 3. In Sect. 4, case studies illustrate the successful implementation of the proposed method, the case study findings are discussed in depth, and the proposed method's application is illustrated. Section 5 concludes the paper.

2 Proposed Method

2.1 Graphical Modelling of Reliability Graph

To evaluate the hazard of network failures, the network's configuration is first plotted into a reliability graph, which is a pair of sets (V, E) where $V = V_1, V_2, V_3, \ldots, V_m$ is the set of all vertices and $E = E_1, E_2, E_3, \ldots, E_n$ is the set of all edges, with m and n being the number of nodes and edges, respectively. Paths in the graph can be broken out to understand the time constraints of a power system control action. Edge weight (E_w) has been assigned to each edge of the graph, and vertex weight (V_w) has been assigned to each vertex, indicating the latency at each transmission path and component, respectively. The power system's latency threshold can be used to determine the possible transmission paths between a source and a target node.

The reliability of route from source node to target node with N number of nodes is the product of the each node's reliability along the path, where each node fails independently with failure rate f_n. The reliability of path is given by following equation:

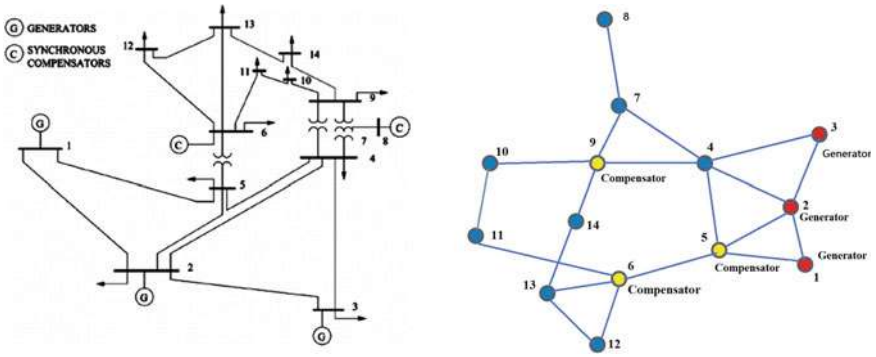
$$R_{\text{path}} = \prod_{n=1}^{N} r_n = \prod_{n=1}^{n} 1 - f_n \tag{1}$$

where r_n and f_n determines the reliability and failure rate of node n, respectively.

Figure 1 shows the IEEE 14 bus system and its equivalent graphical representation.

2.2 Reliability Analysis

Power system reliability is an important concern in the planning, construction, and operation of power distribution systems. Electric power utilities are expected to



(a) IEEE-14 bus system (b) Graphical model of IEEE-14 bus system

Fig. 1 IEEE 14 bus system and its equivalent graph

provide their customers with uninterrupted electrical service at the lowest possible cost while ensuring a reasonable level of service quality.

The value of reliability is demonstrated by the fact that it can be used to express the cost of service outages. The quality of service rendered by a distribution system can be calculated by its reliability indices, which can be enhanced by automating its feeder and related components, resulting in the desired reduction in power interruptions. Power delivery networks that are reliable are those that achieve a high degree of reliability.

A variety of studies have taken a systemic approach to smart grid stability, taking into account, to varying degrees, the effects of cyber network disturbances. The Dijkstra's algorithm has been used in graphical model to find reliability of each path [18]. However, the 'shortest paths' may not be adequate to quantify the level of reliability of the system.

2.3 Network Accuracy Indicator (NAI)

To fix this problem, the network accuracy indicator (NAI) has been introduced, which is based on complex network theory (CNT), and an algorithm for evaluating it is shown in Fig. 4. For a complete power system monitoring and control action, the depth first search (DFS) has been used to explore all possible pathways between source node and target node. Then, the routes denoted by p_c that fulfil the latency requirements for a given power source have been filtered out. Following that, for all $p_x \in p_c$, the sum of the natural logarithm of the reliability value of each cut c_{hv} within the path p_x is determined. Finally, the network's NAI is measured Eq. 2.

$$\mathrm{NAI} = \frac{1}{N(N-1)} \prod_{x=1}^{R} \frac{1}{\prod_{h,v \in V_R} - \log c_{hv}} \tag{2}$$

where R is the total number of routes that fulfil latency criteria, and N is the total number of ICT components.

3 Important Measures

For quantifying overall reliability of power system, there is utmost need to discover metrics that quantify the relative importance of each node in the network. The two metrics namely, structural importance (SI) and total system efficiency (TSE), have been used for this purpose. These are described in detail in following subsection.

3.1 Structural Importance

A component's structural importance (SI) is determined by the system's dependence on that component as well as the value it brings to the network's total efficiency (TSE). The structural value is calculated by the ratio of the difference between TSE (N) and TSE ($N - I$), which are the total system's efficiency with and without node I to TSE (N).

$$\mathrm{TSE} = \frac{1}{N(N-1)} \prod_{i,j \in V_R} \frac{1}{d_{ij}} \tag{3}$$

$$\mathrm{SI} = \frac{\mathrm{TSE}(N) - \mathrm{TSE}(N-1)}{\mathrm{TSE}(N)} \tag{4}$$

where N denotes the total number of nodes in the network, and d_{ij} is the shortest path distance between node i and node j.

3.2 Network Reliability Deterioration Worth

Risk achievement worth (RAW), which calculates the refinement of a component's reliability to one to the overall device reliability, is one of the most commonly utilized punitive measure in nuclear power plant factories. Here, in power system network, it is important to deduce network reliability in terms of risk introduced by each component. Therefore, network reliability deterioration worth (NRDW) is defined by following equation

$$\mathrm{NRDW} = \mathrm{NAI}(1 f)/\mathrm{NAI} \tag{5}$$

Table 1 All available paths from source to target

Path No.	Path	Reliability	Latency (ms)
1	1-2	0.81	3
2	1-5-2	0.729	4
3	1-5-4-2	0.6561	5
4	1-5-4-3-2	0.59049	3
5	1-5-6-11-10-9-7-4-2	0.387420	4
6	1-5-6-11-10-9-7-4-3-2	0.3486784	5
7	1-5-6-12-13-14-9-7-4-2	0.3486784	3
8	1-5-6-12-13-14-9-7-4-3-2	0.3138105	4
9	1-5-6-13-14-9-7-4-2	0.3874204	5
10	1-5-6-13-14-9-7-4-3-2	0.3486784	3

where NAI stands for network accuracy indicator in a normal state, and NRI $(1\,f)$ stands for network misfiring rate indicator when the component's misfiring rate is one (i.e. the component is failed). The component's castigation is then measured as shown in Eq. 5, where NAI $(1\,f)$ is when the component's failure rate is one. When the NRDW is compared to the structural importance (SI) of a component, it can be deduced that the NRDW offers much better results than the pure structural index SI for locating the prime component in a network.

4 Simulation Setup

Simulation is used in reliability modelling to classify nodes where the grid has the highest probability of system-level failure and to determine the system's overall criticality. The custom simulator based on PSAT [19], an open-source power system simulation environment, has been utilized. A data file describing the power grid topology and cyber-infrastructure requirements is fed into the simulator. The IEEE 14 bus system is used as a case study as it is a compact system with a small number of buses and transmission lines. It was important to begin with a small system in order to obtain a better understanding of the system's operation dynamics when line contingencies and distributed generation (DG) sources are implemented. The proposed method does not require a lot of computing power. The experiment we conducted can easily be applied to larger systems, such as the IEEE 57 or IEEE 118 bus systems, giving us a greater understanding of the dynamics of larger systems.

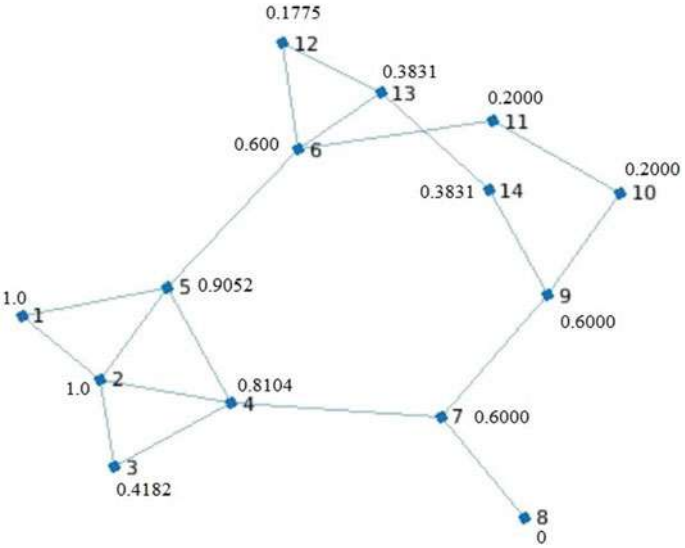


Fig. 2 NRDW index of each node

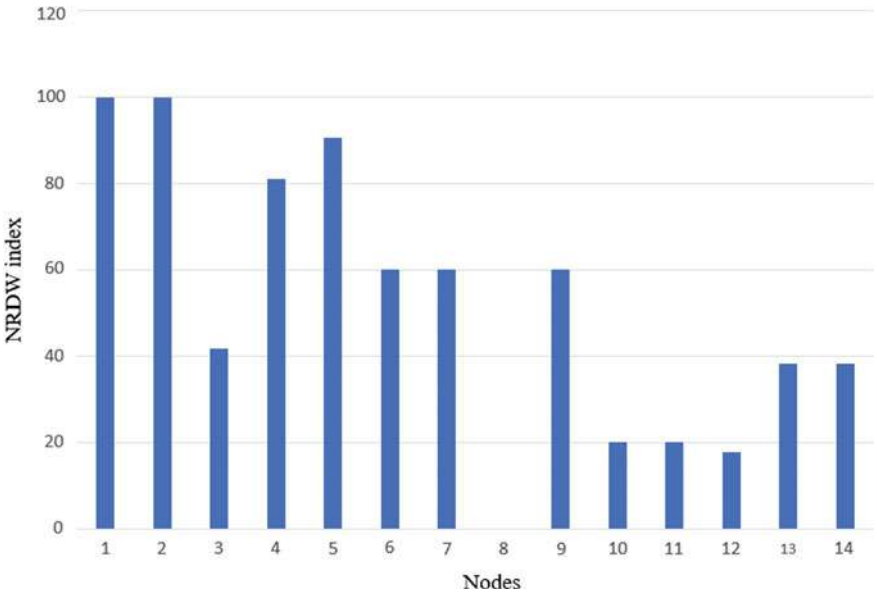


Fig. 3 Comparing NRDW index of each node

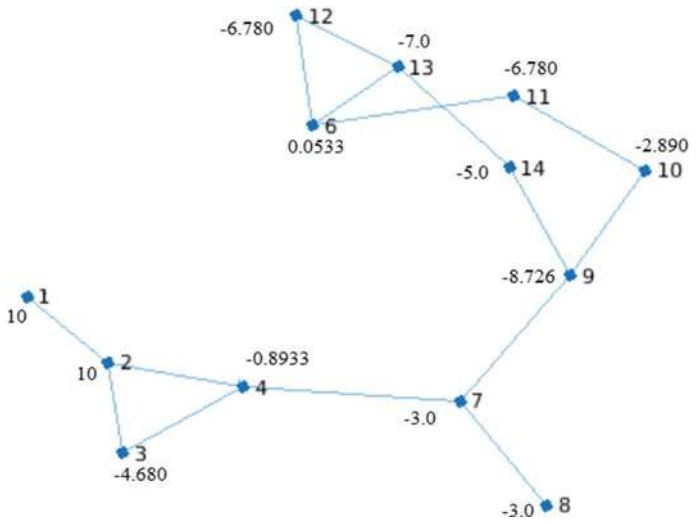


Fig. 4 NRDW values after removal of critical node 5

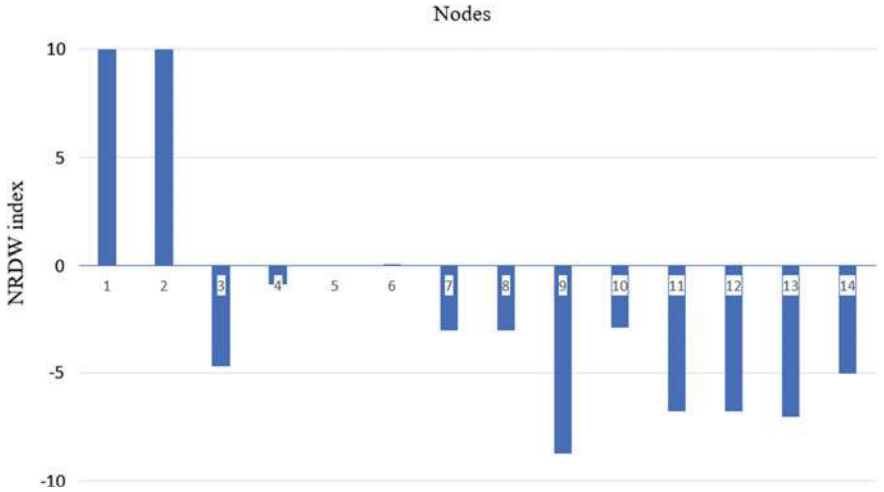


Fig. 5 Comparing NRDW index after removal of critical node 5

4.1 Case Study

The IEEE 14 bus test case is a simplified representation of the American Electric Power grid in February 1962. It has 14 buses, 5 engines, and 11 loads in total. The system's comprehensive data can be found in [20].

4.2 Reliability Analysis

The failure rate of each node and the latency of each communication channel are initialized after the IEEE 14 bus system is modelled into its subsequent graph. The reliability of all N-node routes from the source to the target node is determined after this initialization. The node 1 is used as the source and node 2 as the target node in the case study.

The routing strategies and their total latencies are presented in Table 1, taking into account that the source node has a processing time of 1 s, the target node's data collection time and command execution time are also 1 s, and other nodes manage signals instantly. Table 1 signifies that there are ten different available paths from source node 1 to destination node 2. All the paths have different reliability value c_{hv}. Also, all displayed paths fulfil the latency criteria as its value has been provided with corresponding paths.

The NAI of the IEEE 14 bus system is determined using Eq. 2; it comes out to be 0.11.

4.3 NRDW Analysis

After that, each component's NRDW index is determined using Eq. 5. The NRDW index of each node in the system graph is shown in Fig. 2.

When the findings in Fig. 3 are contrasted, the structural significance of node 5 and its effect on the system becomes apparent. Node 8 on the other hand is not included in any of the routes and thus has no effect on the system's reliability.

Figure 4 depicts the network effect of eliminating the most important node 5. The system's reliability index NAI decreases to 0.0102 from 0.1000 after node 5 is withdrawn, suggesting a 90% reduction in reliability. As a consequence, the criticality of node 5 can be deduced from the preceding observations. Node 9 is now the system's most important node as seen from Fig. 5. After removal of critical node 5, we can see there is a only a single path from source, i.e. node 1 to the target node 2, so all the nodes will have a negligible impact on the criticality of the system. This is can be observed through the below graph. And since the critical node is also the node after removal of which, the system becomes the most fragmented, we observe the node 9 is the new critical node of the system.

5 Conclusion

The graphical computational model has been developed to signify the structural importance of ICT components. The model is computationally inexpensive and easy to deploy and efficient in visualizing the contribution of each node in network relia-

bility. The contributions are twofold: first, the overall network's reliability has been calculated using network accuracy indicator (NAI) metric. Second, relative importance of each node in the network has been analysed using structural importance (SI) and network reliability deterioration worth (NRDW). Simulations on standard IEEE 14 bus system demonstrate the most and least important node in contribution to the network reliability according to the NAI calculations. Then, the results of network overall reliability after removal of the most important node have been discussed. It has been shown that removing the node with maximum NRDW can greatly affect the overall reliability of network. The graphical model and the metrics introduced in the manuscript greatly help in the power network reliability assessment. Future research will concentrate on larger systems, such as IEEE 118 bus systems, as well as the effects of PV sources on the system.

References

1. Dalamagkas C, Sarigiannidis P, Ioannidis D, Iturbe E, Nikolis O, Ramos F, Rios E, Sarigiannidis A, Tzovaras D (2019) A survey on honeypots, honeynets and their applications on smart grid. In: 2019 IEEE conference on network softwarization (NetSoft), pp 93–100
2. Marashi K, Sarvestani SS, Hurson AR (2018) Consideration of cyber-physical interdependencies in reliability modeling of smart grids. IEEE Trans Sustain Comput 3(2):73–83
3. Jamshid S, Abrar M (2020) Evaluation of load point and customer point indices of a distribution generation system. In: 2020 International youth conference on radio electronics, electrical and power engineering (REEPE), pp 1–5
4. Praks P, Kopustinskas V, Masera M (2017) Monte-Carlo-based reliability and vulnerability assessment of a natural gas transmission system due to random network component failures. Sustain Resilient Infrastruct 2(3):97–107
5. Wu L, Chenglin W, Haipeng R (2017) Reliability evaluation of the solar power system based on the Markov chain method. Int J Energy Res 41(15):2509–2516
6. Jakkula B, Mandela GR, Murthy ChSN (2020) Reliability block diagram (RBD) and fault tree analysis (FTA) approaches for estimation of system reliability and availability—a case study. Int J Qual Reliab Manag
7. Kemikem D, Boudour M, Benabid R (2018) Reliability modeling and evaluation of reparable electrical power supply systems using reliability block diagram. In: 2018 International conference on electrical sciences and technologies in Maghreb (CISTEM), pp 1–6
8. Chybowski L (2020) Importance analysis of components of a multi-operational-state power system using fault tree models. Information 11(1)
9. Nakarmi U, Rahnamay-Naeini M (2018) Analyzing power grids' cascading failures and critical components using interaction graphs. In: 2018 IEEE Power energy society general meeting (PESGM), pp 1–5
10. Falahati B, Kahrobaee S, Ziaee O, Gharghabi P (2017) Evaluating the differences between direct and indirect interdependencies and their impact on reliability in cyber-power networks. In: 2017 IEEE Conference on technologies for sustainability (SusTech), pp 1–6
11. Zheng J, Okamura H, Pang T, Dohi T (2021) Availability importance measures of components in smart electric power grid systems. Reliab Eng Syst Saf 205:107164
12. Hariri A-M, Hashemi-Dezaki H, Hejazi MA (2020) A novel generalized analytical reliability assessment method of smart grids including renewable and non-renewable distributed generations and plug-in hybrid electric vehicles. Reliab Eng Syst Saf 196:106746
13. Suwanasri C, Saribut S, Suwanasri T, Phadungthin R (2021) Risk analysis using failure modes, effects, and criticality analysis for transmission network assets. Energies 14(4)

14. Duman O, Zhang M, Wang L, Debbabi M, Atallah R, Lebel B (2020) Factor of security (FoS): quantifying the security effectiveness of redundant smart grid subsystems. IEEE Trans Dependable Secure Comput 1–1
15. Rivas AEL, Abrão T (2020) Faults in smart grid systems: monitoring, detection and classification. Electr Power Syst Res 189:106602
16. Jin Z, Yu P, Guo SY, Feng L, Zhou F, Tao M, Li W, Qiu X-S, Shi L (2020) Cyber-physical risk driven routing planning with deep reinforcement-learning in smart grid communication networks. In: 2020 International wireless communications and mobile computing (IWCMC), pp 1278–1283
17. Hossain NUI, Jaradat R, Hosseini S, Marufuzzaman M, Buchanan RK (2019) A framework for modeling and assessing system resilience using a bayesian network: a case study of an interdependent electrical infrastructure system. Int J Crit Infrastruct Prot 25:62–83
18. Panic B, Kontrec N, Vujosevic M, Panic S (2020) A novel approach for determination of reliability of covering a node from k nodes. Symmetry 12:1461
19. Milano F (2005) An open source power system analysis toolbox. IEEE Trans Power Syst 20(3):1199–1206
20. Ordoudis C, Pinson P, Morales JM, Zugno M (2016) An updated version of the IEEE RTS 24-bus system for electricity market and power system operation studies, vol 13. Technical University of Denmark

Chapter 39
Safety Rings Principle and Energy and Information Networks Coupling

Bianca Tonino-Heiden, Bernhard Heiden, and Volodymyr Alieksieiev

1 Introduction

Energy and information can be regarded as a back-coupling system, which is formulated as the fourth and fifth main sentences of thermodynamics, leading to a direction in entropy and hence direction in energy order [18]. An energy-information-material system or a cybernetic system can, concerning the information, be regarded as a self-ordering back-coupling system, in non-equilibrium thermodynamics, which can (1) be put in a neat set of balancing equations extending the knot matrix equation-based material energy-solving system [42] and by (2) the *informational knot matrix*, balancing around information networks [19]. The ordering of informational structures is closely connected to algorithms [5] that implement in a higher-order cybernetic system, a computer programming environment, the higher-order back-coupling in the system with the system, which then results in increasingly higher automated or artificial intelligence (AI) systems, pointing to the order increase utilising implicit informational density. In systems theory, the graph theory, which is the mathematical generalisation of networks consisting of vertices and edges, plays an essential role [5, 34, 35], which we will discuss in this article concerning energy systems to some aspects, the Safety Rings Principle (SRP).

Goal and Method The goal of this paper is to introduce a new system theoretic tool and to illuminate the mechanisms of systems with high survivability. We apply then this principle as an example to a network of energy and information and investigate

B. Tonino-Heiden · B. Heiden (✉)
University of Graz, 8010 Graz, Austria
e-mail: b.heiden@cuas.at
URL: http://www.cuas.at

B. Heiden
Carinthia University of Applied Sciences, 9524 Villach, Austria

V. Alieksieiev
Leibniz University Hannover, 30823 Garbsen, Germany

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_40

some general properties here. The method used is system theoretic motivated by using axioms in form of natural language arguing.
Content In this paper, we first formulate and then investigate in Sect. 2 the Safety Rings Principle (SRP) and give the related axioms. In Sect. 3, we give an example of coupling of energy and information networks and show how this relates to the new tool. Finally in Sect. 4, we conclude what we have achieved and give an outlook for further developments in this field of research.

2 Safety Rings Principle (SRP)

Descartes has originated fundamentally the thinking in the Western philosophical tradition, by means of a decisive epistemology. The ancient Greece scientific philosophical tradition is mainly beginning with Socrates way of dialogical thinking (*dialeghestai*—to talk so. through sth.) [1], which is systemically the beginning of information flow in two directions, which can be regarded as a bidirectional graph, according to graph theory (see also [5]), Aristotle's logic calculus [44] and Euklid's [8] axiomatic geometry. Descartes has built his foundation of science, an increasingly self-ordered system, (1) on these geometric principles, and (2) on the relation of geometry and algebra (compare [53, §14], where he relates a *unit measure* to a *general tool*, which can be identified as Descartes's coordinate system tool). He argues that more generally applicable methods are of higher value [40, p. 570]. In [6], Descartes describes how the human epistemological cycle works, which gives birth to modern science. But is it sufficient to know *how to* for science? Allowing for agent-based rationality, leads to shifting towards a *what for* paradigm.

When shifting from *how to* to *what for* (compare also Frankl [11, 12] and Nietzsche's 'reason why' [39, p. 135]) the human impact gets greater in the world. Descartes's epistemology contains the *agent-based form* implicitly. When deciding not to publish his *Regulae ad directionem ingenii* [6], this was also due to the fact, that it was highly probable, that he would have suffered from religious prosecution, if he had revealed the *scientific method* publicly in his lifetime. This means, he showed his insights epistemologically, *by acting* in an *explicitly* non-acting way, which he documented in his writing. So the Safety Ring (SR) according to the following Axiom 1–3 can be interpreted here as the time buffer—in general as the room-time buffer—as Descartes used his own knowledge for his own thinking and the time in which he did not publish it, knowing the impact of publishing and the accompanying back-coupling effect. The axioms can now be formulated for the SRP:

Axiom 1 *A Safety Ring (SR) can be regarded as a system that encloses a system symbiotically, hedging and hence giving survivability.*

Axiom 1 denotes a process of growing rings that can be found in nature in trees or tulips structurally.

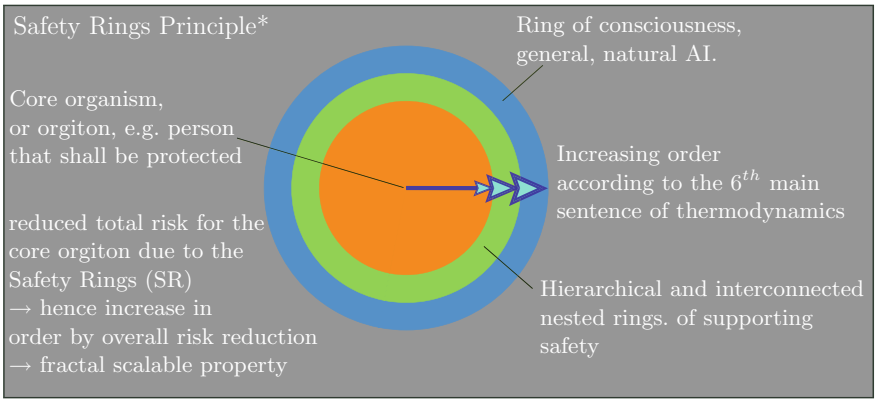


Fig. 1 Rings of safety—*cf. also [16, 19, 21]

Axiom 2 *The order increase is double or multi-folded: By a distance measured from the inner core to the outer environment, and by a hierarchic self-functional distance to the inner–outer border.*

The schematic according to Axiom 2 is depicted in three colours in Fig. 1. The growing core can be interpreted as an increase in *space-time* order. The space is indicated by the core and the time by the digits. And from the physics standpoint, they can be regarded as a nested or interconnected variable, space-time (cf. also [32]).

Axiom 3 *The SR comprises a multi-level border membrane structure from the inner to the outer system.*

Axiom 3 can also be interpreted from the osmotic paradigm [17, 26, 45, 46], which means that, there is a flow of information and material through the system, by this increasing the diversity of the regarded flow paths. Here flow is understood very general. Examples are energy or information flow.

Furthermore, we can connect a discovery from [50], that small and sparsely connected systems can be regarded similar like a fully and dense connected system. That means that the Safety Ring Separation according to Axiom 1–3 and Fig. 1, leads to one system, although, as long as there is an even sparse connection, this can be regarded in general as an osmotic flow field. A prominent example is the human with his three brains (cf., e.g., Koestler [30]), having one personality.

Another aspect can be followed from the SRP: That each SR core constitutes an emergent order an intermission at the system environment border (c.f. [24, 33]).

This guideline [6] is now the foundation of science epistemology of all kinds. According to Wittgenstein, facts about what a person/things in the world is/are doing compared to what he/she is saying [52, §4.122, §6.36] shows by itself. The method of Descartes divides matter and mind sharply. Both are acting according to this

framework independently but connected like two synchronised clocks according to Ceulinx, Descartes' pupil [40, p. 571]. This allows for the mind-performed analysis method.

Contrary to this, according to Descartes' interpretation, matter and mind can be regarded as interconnected. This becomes increasingly important with human–nature order evolution. The mind–matter connection begins with this to make a difference, in the thinking mode. That means it is nowadays, not possible to design merely *a how* to think, but it becomes increasingly necessary to have a *what for* to think of. This seems to be a little difference, and it is one. But a decisive and meaningful. This interpretation can be given with a second recourse to his method and the SR here.

In [6], Descartes like the growing rings of a tree or tulip structurally gave rules R_i $(i = 1 \ldots 21)$, which describe the *scientific method* in general, with an application to the geometric algebra. This describes a closed *cybernetic-cycle* of epistemic knowledge generation. The rules can be classified as neutral ones $(R_{1,2,3,4,5,6,7,8,13,17,18,19,20,21})$, sensoric ones $(R_{9,10,11,12})$ and actoric ones $(R_{14,15,16})$ with regard to a general cybernetic-cycle, here connected with material or knowledge production (compare also [41, 48, 49]). The sensoric input is related to the subject that is *perceiving* (compare also the English empiricists, Berkeley [3, 4] and Hume [28, 29, 31, p. 30]) the world and hence the *mental epistemological side*, whereas the actoric output is related to the *agent-epistemological side* of the cycle—the material side. The neutral layer of rules is related to ordering tasks that seem neutral to the subject, as intermediary helpful steps in the *cybernetic epistemological cycle*.

Generalisation of Descartes method, which introduces first the geometric algebra and can be later on used for the general science epistemological process, allows for all sorts of human knowledge generation and above. The actions of humanity are expanding ever further into the cosmos and are thus increasingly returning to themselves. Therefore, increasing SRs becomes more necessary, increasing both humanity and rising posthumanity (cf. [2, 10]), when shifting the epistemological cycle from the mind to the material balance-point starting point, thereby moving from independent *how to* to interconnected *what for* epistemological ethics.

3 Coupling of Energy and Information Networks

In this section, we will try to give an application of the SRP.

3.1 Key Aspects

The essential part of increasing energy efficiency is to provide an algorithm-like system, an energy material information system. The informational aspect is an ever meta-aspect of a system, framing it thoroughly. To increase order in living systems, the order has to build upon order, as Schrödinger stated [43] and also Prigogine

according to [47]. Concerning energy nets, they can be divided into elementals, and these can then be designed as a network with a material information system, which can be solved sequentially or matrix oriented, e.g. with the informational knot matrix method [19]. For dense networks and, in general, back-coupled networks, which can be regarded as strongly coupled according to system or graph theory, the equation-based methods seem mostly to be more suitable for solving the accompanying non-linearities. Anyhow when the complexity or network connectivity is increasing, the system changes fundamentally. This appears then as emergence of the principle of diversity or complexity order increase through variegated networks of different sorts of, e.g. energy forms. The main sustainable driver for energy systems is syntropy triggered. Hence, material energy-transportation lengths have to be short, leading to necessarily some sort of decentralised systems for energy optimisation (cf. also [22]). Humans themselves, concerning existence and organisation, are operating decentral in nature. This is shortly related to their decisions. Hence, the goal of energy systems is related to humans, as is their nature. Therefore, energy systems can be regarded as a system enhancing humans decentralised, together with a materialised component, that is, physically, materially and informationally communicating, leading to potentially optimal decentralised order increase.

The nesting of communication and information directions leads then to a potential higher-order concerning cooperation. And hence there is for each a specific combination of cooperation and concurrence system optimal. This also corresponds well to Grossberg's statement that both are important in human neuronal networks [13].

3.2 Coupling of Energy, Information from a Material Logistics Networks View

The coupling of energy and information can also be applied in material logistic networks. Material and information flows, which are the main elements of logistic processes, have to be powered by energy to be in motion. The form of energy can be different and depends on a rule and on many different factors (size of networks, automation grade, etc.). In [16], the scalable logistic cell is defined and considered as a scalable system, which has functional autonomy as well as potential interaction with other cells in material, energy and information levels. To demonstrate the coupling of matter, energy and information this cell with a storage option was simulated in WITNESS (see Fig. 2).

In this model, internal and external processes of such a logistic cell were simulated to demonstrate a potential interaction between different cells (see Fig. 2). The cell with a storage option (cell 2) has an external communication with a supplier (cell 1) and client (cell 3). Order in the form of information (paper-based or digital) has to be received by cell 2 from a client and according to this, a consignment has to be ordered to and delivered from a supplier. The delivery process has to be done with a truck (material object), driven by a driver (energy producer) and powered by driven

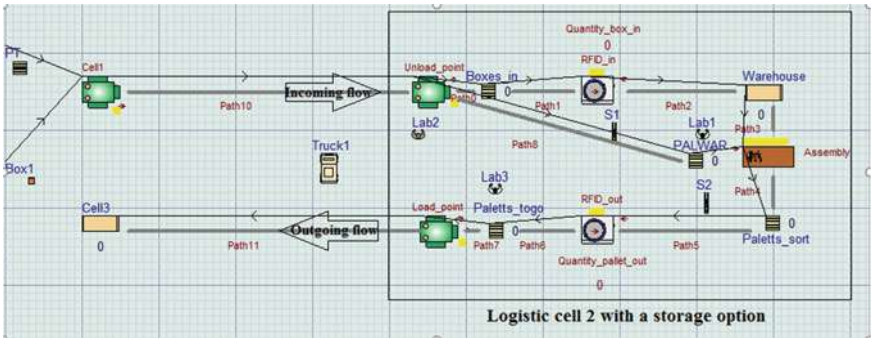


Fig. 2 Model of scalable logistic cell with storage option in WITNESS [16]

energy, according to the order (a form of information). The delivery process from the warehouse to a customer is equal.

In internal processes, an important role plays the identification procedure. As an item's identification means the Radio Frequency IDentification (RFID) technology was chosen, which has many advantages compared to the second most popular identification technology—barcode (overcoming a bigger distance, scanning up to 1000 items per second, sight contact is not needed, etc.). In addition to this, the RFID technology enables us to show the coupling of matter, energy and information immediately. In the first and final stages, where pallets with goods enter and leave the cell, they have to be transported through the RFID frame. This technology yields immediate scanning of up to 1000 items per second. Hence, the bottleneck point in this stage can be avoided. At the moment of identification, a forklift truck, driven by a driver (human energy and drive energy), with chipped pallets (material objects), is going through the ramp with the RFID reader, powered with electrical energy. As a result, the information about the items is shown on the PC display [16].

Another example to show a practical application of information and energy networks coupling in logistics can be regarded osmotic manufacturing [26] and AI. The basic idea of osmotic manufacturing is that a flow field of each production or manufacturing can be shown as a flow network by means of graph theory. Using this graph-theoretical representation, the amount of information can be decreased, which can lead to energy reduction in complex networks, such as AI-based networks [26].

As logistic processes require high energy consumption, it impacts the sustainability of the environment negatively [51]. The energy-based design of information networks can lead to improving energy efficiency. The in [9] presented information model for information and energy integration in production and logistic processes enables producers to identify possibilities for energy saving. These works confirm the significant role of energy and information coupling in logistic systems for sustainable meta-purposes.

To summarise, internal and external processes in logistics, such as packaging, transportation of goods, communication and dispatching, are powered by a different

form of energy (electrical, drive, human, etc.), which drives the information and material flow. Moreover, with the increasing complexity of the world and logistic processes, the number of transportation, information and energy forms is increased. Hence, the new coupling possibilities of energy and information networks arise.

Here order increase is by sensor networks. The actor is related, e.g. to the material flow. And the optimisation is done (mostly) by computers and algorithms. So this can be each regarded as one SR core, or all three as a nested one, as one orgitonal system, as it constitutes or comprises a cybernetic unit of mass, energy and information (cf. also [20, 25]).

3.3 Coupling of Energy and Information Networks in the Cooperation Regime

The increasing coupling of systems can be regarded as a human effort of the connection of science and society [14], but also of a deeper epistemological connection between human and nature, in one scientific framework [15]. According to Eccles [7] the human evolution, triggered by the brain, the hand and the eye, which constitutes a basic cybernetics system—or an orgiton—according to the orgiton theory [20, 25], has led to increasing social and hence social networking capabilities of humans. This has only in the last century led to astonishing results in thinking and technology, through broad cooperation in science and society. Eccles argued that the aggression area in the brain has grown slower than the rest of the brain, leading to relatively lower aggression or concurrence and hence higher cooperation potential. The increasing density of population makes further necessary cultural techniques of aggression reduction and cooperation increase, even necessary for survival. This is then an equilibrating power. But why?

Trees are fractal structures, as well as networks [36], meaning that they have a kind of logarithmic or scalable order. Trees in the context of graphs are a special sort of graphs or networks, they are sparse networks with special properties [5, 50]. The same is true for energy networks, or networks involving different sorts of energy, where transformation in between is possible, e.g. between, thermal, mechanical or electrical energy, for example, and their storage equivalents. There is foreseeable an immense change in tree-like operation compared to network operation, boosting potential efficiency using a back-coupling mechanism, that is intrinsic to network topologies. An interesting feature is here, that the bioprocess principle [38] of diversity for optimisation of the federate of natural species holds as well as in the portfolio theory of Markowitz [37], which then can be regarded as different scaling informational material systems with universal statistical network properties. Finally, the materialisation of increasingly greater and global systems leads to greater vulnerability, back-coupling and to increasing necessity of autonomous units, that are increasingly localised or decentral and global at the same time, which is also called glocal, as a general property. Hence the local autonomy can be seen as a SR, accord-

ing to Axiom 1–3 and Fig. 1, for both better survivability and better survivability by intrinsic cooperation.

4 Conclusion and Outlook

In this paper, we have introduced the system model concept Safety Rings Principle (SRP), which is applicable to cybernetical systems or orgitons that form orders with specific attractors (cf. [27]). This concept consists basically of three axioms and has several interconnected properties related to networks and graph theory in general. The given application in energy networks and logistics shows how some of these SR properties can be built and nested together. Among others the property of forming one system, and one that is stable. In a recent paper, we denoted this also as Ladder of trust [23], which means that it is a SR with different topology. In both cases, the directionality principle is of importance, as by this can be described how order can be built. In the SR, this is done by successive SR cores that enclose the form forming one whole, allowing for osmotic flows throughout the system.

As an outlook, we plan to further apply this system to computer simulations and may derive from these further properties of such systems, first in general and second in specific applications. Another important and difficult research development may be the question to be answered, for which all related researchers are invited to join the effort, which specific algorithms do apply to the specific applications of energy, logistics and many other even far above interdisciplinary ones.

References

1. Arendt H (1993) Philosophie und Politik. Dtsch Z Philos 41(2):381–400
2. Barfield W, Williams A (2017) Law, cyborgs, and technologically enhanced brains. Philosophies 2(1):1–17. https://doi.org/10.3390/philosophies2010006
3. Berkeley G (1979) Eine Abhandlung über die Prinzipien der menschlichen Erkenntnis. Felix Meiner Verlag, Hamburg
4. Berkeley G (1988) Principles of human knowledge and three dialogues. Penguin Books Ltd
5. Dasgupta S, Papadimitriou C, Vazirani U (2008) Algorithms. The McGraw-Hill Companies
6. Descartes R (1973) Regulae ad directionem ingenii - Lateinisch- Deutsch. Felix Meiner Verlag, Hamburg
7. Eccles JC (1989) Evolution of the brain creation of the self. Routledge, London
8. Euclid (2008) Euclid's elements of geometry (edited and provided with a modern English translation, by Fitzpatrick R). The University of Texas at Austin
9. Feng S, Kumaraguru S, Law K, Ahn K (2013) Integration of energy information for product assembly and logistics processes. In: IFIP Advances in information and communication technology. Springer, Berlin, Heidelberg, pp 310–317. https://doi.org/10.1007/978-3-642-41266-0_38
10. Ferrando F (2020) How to become posthuman—Dr. Ferrando lectures at Princeton University, US. Envision conference 2019. Youtube video lecture. https://youtu.be/CwUBLOm7PQQ
11. Frankl VE (2006) Man's search for meaning. Random House LCC US

12. Frankl VE (2017) Wer ein Warum zu leben hat Lebenssinn und Resilienz (Deutsch). Beltz GmbH, Julius
13. Grossberg S (2021) Conscious mind, resonant brain: how each brain makes a mind. Oxford Univ PR, Oxford
14. Götschl J (2006) Selbstorganisation Ein Denksystem für Natur und Gesellschaft. In: Chapter Selbstorganisation: Neue Grundlagen zu einem einheitlicheren Realitätsverständnis. Böhlau Verlag, Köln, pp 35–65
15. Götschl J (2020) Zur Epistemologie der Selbstorganisation: Von Konvergenzen zu Korrelationen zwischen Systemwissenschaften der Natur und Systemwissenschaften vom Menschen. Theorien und Methoden der Psychologie. Verlag Dr. Kovac, Hamburg, pp 85–104
16. Heiden B, Alieksieiev V, Tonino-Heiden B (2020) Scalable logistic cell RFID Witness model. In: Proceedings of the 5th international conference on internet of things, big data and security, vol 1: IoTBDS. Scitepress—Science and Technology Publications, pp 420–427. https://doi.org/10.5220/0009490204200427
17. Heiden B, Kattnig D, Pechmann J, Alieksieiev V, Tonino-Heiden B (2021) Wheelchair climbing construction for industrial logistical applications. WMMES. IOP Conf Ser: Mater Sci Eng 1190:012020. https://doi.org/10.1088/1757-899X/1190/1/012020
18. Heiden B, Leitner U (2018) Additive manufacturing—a system theoretic approach. In: Drstvenšek I (ed) ICAT 2018, Maribor, 10–11 Oct 2018. Interesansa - zavod, pp 136–139
19. Heiden B, Tonino-Heiden B (2020) Key to artificial intelligence (AI). In: Arai K, Kapoor S, Bhatia R (eds) Advances in intelligent systems and computing, vol 1252. Springer, Cham, pp 647–656. https://doi.org/10.1007/978-3-030-55190-2_49
20. Heiden B, Tonino-Heiden B (2021) Philosophical studies—special orgiton theory (Philosophische Untersuchungen - Spezielle Orgitontheorie), English and German edn (unpublished)
21. Heiden B, Tonino-Heiden B (2022) Emergence and solidification-fluidisation. In: Arai K (ed) LNNS 296, Lecture notes in networks and systems. Springer Nature Switzerland AG, pp 845–855. https://doi.org/10.1007/978-3-030-82199-9_57
22. Heiden B, Tonino-Heiden B, Alieksieiev V, System ordering process based on uni-, bi- and multidirectionality—theory and first examples. In: Hassanien AE, Xu Y, Zhao Z, Mohammed S, Fan Z (eds) Business intelligence and information technology. BIIT 2021. Lecture notes on data engineering and communications technologies, vol 107. Springer, Cham, pp 594–604. https://doi.org/10.1007/978-3-030-92632-8_55
23. Heiden B, Tonino-Heiden B, Alieksieiev V (2021) Ladder of trust. In: Arai K (ed) Proceedings of the future technologies conference (FTC) 2021, volume 3. FTC 2021. Lecture notes in networks and systems, vol 360. Springer, Cham, pp 804–813. https://doi.org/10.1007/978-3-030-89912-7_61
24. Heiden B, Tonino-Heiden B, Alieksieiev V, Hartlieb E (2021) Digitalisation model innovation system. In: O'Dell M (ed) 2021 10th International conference on industrial technology and management (ICITM). IEEE, Online, pp 128–133. https://doi.org/10.1109/ICITM52822.2021.00030
25. Heiden B, Tonino-Heiden B, Wissounig W, Nicolay P, Roth M, Walder S, Mingxing X, Maat W (2019) Orgiton theory (unpublished)
26. Heiden B, Volk M, Alieksieiev V, Tonino-Heiden B (2021) Framing artificial intelligence (AI) additive manufacturing (AM). In: Procedia computer science, vol 186. Elsevier B.V., pp 387–394. https://doi.org/10.1016/j.procs.2021.04.161
27. Hilborn RC (1994) Chaos and nonlinear dynamics—an introduction for scientists and engineers. Oxford University Press, New York
28. Hume D (1985) A treatise of human nature. Penguin Books Ltd
29. Hume D, Brandt R (2013) Ein Traktat über die menschliche Natur. Meiner Felix Verlag GmbH
30. Koestler A (1976) The ghost in the machine, Danube edn. Hutchinson
31. Kulenkampff J (2003) David Hume. Beck C. H
32. Lieber LR, Lieber HG (2008) Einstein theory of relativity. Paul Dry Books Inc., Philadelphia
33. Luhmann N (2018) Soziale Systeme, 17 edn. Suhrkamp Verlag AG
34. Lunze J (2008) Automatisierungstechnik, 2nd edn. Oldenburg Verlag, München

35. Läuchli P (1991) Algorithmic graph theory (program practice) (Algorithmische Graphentheorie (Programm Praxis), German). Birkhäuser
36. Mandelbrot BB (1991) Die fraktale Geometrie der Natur. Birkhäuser Verlag, Basel Boston Berlin
37. Markowitz HM (1952) Portfolio selection*. J Finan 7(1):77–91
38. Moser A (1981) Bioprozeßtechnik. Springer, Vienna
39. Nietzsche FW (1990) Unmodern observations (Unzeitgemässe Betrachtungen). Yale University Press, New Haven, London
40. Russell B (2011) Philosophie des Abendlandes - Ihr Zusammenhang mit der politischen und der sozialen Entwicklung. Europa Verlag Zürich, 3. Aufl., History of Western Philosophy (Routledge Classics) (English)
41. Scharmer O (2016) Theory U: leading from the future as it emerges. Berrett-Koehler
42. Schnitzer H (1991) Grundlagen der Stoff- und Energiebilanzierung. Vieweg Verlag, Braunschweig
43. Schrödinger E (1989) Was ist Leben - Die lebende Zelle mit den Augen des Physikers betrachtet. Piper Verlag
44. Shields C (2016) Aristotle. In: Zalta EN (ed) The Stanford Encyclopedia of philosophy, winter 2016 edn. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/entries/aristotle/#LogSciDia
45. Tonino-Heiden B, Heiden B, Alieksieiev V (2021) Artificial life—investigations about a universal osmotic paradigm (UOP). In: Arai K (ed) Intelligent computing, LNNS, vol 285. Springer Nature, pp 595–605. https://doi.org/10.1007/978-3-030-80129-8_42
46. Villari M, Fazio M, Dustdar S, Rana O, Ranjan R (2016) Osmotic computing: a new paradigm for edge/cloud integration. IEEE Cloud Comput 3:76–83
47. von Bertalanffy L (2009) General system theory. George Braziller, revised edn
48. von Foerster H (2003) Understanding systems: conversations on epistemology and ethics. Springer, US
49. von Förster H, Vogl J, Pircher W, Holl U, Schüttpelz E, Schnidgen H, Bitsch A, Kay LE, Hörl E, Segal J, Coy W, Hrachovec H, Baecker D, Brand S, McCulloch WS, Pitts W, Rosenbluth A, Wiener N, Bigelow J, McCulloch WS (2004) Cybernetics—Kybernetik, the Macy-conferences 1946–1953, vol II, band II. Essays and documents (Essays und Dokumente, vol II, band II). Diaphanes, Zürich, Berlin
50. Watts D, Strogatz S (1998) Collective dynamics of 'small-world' networks. Nature 393:440–442
51. Wehner J (2018) Energy efficiency in logistics: an interactive approach to capacity utilisation. Sustainability 10(6):1727. https://doi.org/10.3390/su10061727
52. Wittgenstein L (2021) Tractatus Logico-Philosophicus Logisch-philosophische Abhandlung. First published by Kegan Paul (London), 1922. Side-by-Side-by-Side edn, Version 0.59, containing the original German, alongside both the Ogden/Ramsey, and Pears/McGuinness English translations
53. Wittgenstein L (2019) Philosophical investigations (Philosophische Untersuchungen, German), 9 edn. Suhrkamp Verlag AG

Chapter 40
A Deep Learning Model for Early Prediction of COVID-19 Spread

Ulises Manuel Ramirez-Alcocer, Edgar Tello-Leal, Jaciel David Hernandez-Resendiz, and Barbara Azucena Macias-Hernandez

1 Introduction

The severe acute respiratory syndrome (SARS-CoV-2) virus was identified as the causative of the disease known as coronavirus 2019 (COVID-19) [1]. This virus has a quick-spreading capacity in all its variants through a person-to-person transmission scheme, with a basic reproduction number (*R*0–*R* naught) of 2–2.5 [2]. The *R*0 can be understood as the expected number of secondary cases for a single infection in a susceptible population [3]; that is, it is the average in which each patient can transmit the infection to other individuals [4].

According to the available information on the COVID-19 infection, the spread of the virus outside China was presented by business people and tourists who were in Wuhan or some regions of China between January and February 2020 [5], exporting these cases mainly to countries in Asia, Europe, Oceania, and North America [6, 7]. The transmission potency of the virus has been confirmed to be person-to-person [8], infection occurs through contaminated secretions, when there is prolonged and unprotected exposure to infected people [9], raising concerns about global spread [10].

Estimating changes in transmission over time allows us to know insights into the epidemiological situation [11]. In this sense, a well-fitting prediction model generates values close to the real values contained in the input data. Time series analysis and forecasting problems are considered the most challenging and complicated data mining and machine learning issues. Deep learning (DL) architectures have appeared

U. M. Ramirez-Alcocer · J. D. Hernandez-Resendiz
Multidisciplinary Academic Unit Reynosa-Rodhe, Autonomous University of Tamaulipas, 88779 Reynosa, Tamaulipas, Mexico

E. Tello-Leal (✉) · B. A. Macias-Hernandez
Faculty of Engineering and Science, Autonomous University of Tamaulipas, 87000 Victoria, Tamaulipas, Mexico
e-mail: etello@docentes.uat.edu.mx

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_41

as a promising technique for automatically learning arbitrary complex mappings from inputs to outputs [12]. Long short-term memory (LSTM) and convolutional neural network (CNN)-based DL architectures have been implemented for time series predicting of COVID-19 transmission [13–15]. On the one hand, the LSTM models are used to predict the time series datasets [16]. On the other hand, a hybrid CNN-LSTM was employed to detect COVID-19 automatically using X-ray images [17]. The idea behind forecasting is to predict future values of data based on what happened before.

In this paper, we propose a data-driven analysis using DL techniques to predict the potential growth of COVID-19 infections. We forecast probable future confirmed cases using a time series dataset by country (Argentina, Brazil, Colombia, Ecuador, and Mexico as case studies). A hybrid model based on CNN and LSTM neural networks was employed to develop a mechanism to predict future cases of COVID-19 using a real-life dataset. The performance of the CNN-LSTM-based inference model is evaluated using minimum root-mean-squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and coefficient of determination (R^2) metrics. The experimental results suggest that the proposed method achieves acceptable measures in forecasting the future COVID-19 confirmed cases.

This paper is structured as follows. Section 2 explains the metrics used to validate the performance of the inference model. Section 3 describes the applied methodology with its stages and activities, and the structure of the inference model proposed. In Sect. 4 the prediction results obtained by the DL model are presented. The conclusions of the proposed approach are discussed in the last section.

2 Background: Performance Evaluation Metrics

We use two traditional indicators to evaluate the model's prediction performance and generalization ability, the RMSE, MAE, MAPE, and R^2. The RMSE (see Eq. 1) represents the standard deviation of the prediction results of the model, and the MAE (see Eq. 2) directly gives an average difference between the outcomes of the model and the actual data.

RMSE is the square root of the average squared errors between the predicted and experimental values calculated by Eq. 1. RMSE is always a positive value, and a value 0 indicates a perfect prediction.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \tag{1}$$

where y_i denotes the true value or target value, $\hat{y}_i$ denotes the predicted value, and N denotes the number of evaluation samples. This notation is similar in Eqs. (2), (3), and (4).

MAE is the absolute error average and reflects the error between the actual value (y) and the predicted value ($\hat{y}$). The lower the MAE value, the higher the precision the prediction achieves. The equation is as follows (Eq. 2):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{2}$$

MAPE represents the ratio between the error and the actual value, i.e., the error is defined as the actual value minus the predicted value. The lower the MAPE value, the closer the predicted value is to the true value (Eq. 3).

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i - \hat{y}_i\right|}{y_i} \times 100 \tag{3}$$

The coefficient of determination (R^2) measures how well the predicted values match with the observed values. The formula is as follows (Eq. 4):

$$R^2 = 1 - \frac{\sum\left|y - y^{'}\right|}{\sum\left|y - \hat{y}\right|} \tag{4}$$

3 Methodology

Globally, there are many challenges to face to reduce the social, healthcare, and economic impact caused by the transmission of the COVID-19 infection. Thus, the accurate prediction of future confirmed COVID-19 cases is essential for improving public health emergency response capacity. We formally define the problem to predict new cases as $T = \{t_1, t_2, \ldots, t_k\}$ to denote the time slots of the dataset, where K is the number of the time slots. Given the raw data of the number of cases time series of a country $\{X_t | t = 1, \ldots, T\} \in \mathbb{R}^{\mathrm{T}\times\mathrm{M}}$, we aim to predict the number of new confirmed cases y_{T+1} of the next time slot t_{T+1}.Where M is the total categories of cases types (confirmed, deaths, recovered), and $y_k \in \mathbb{R}$ denotes the number of confirmed cases in the time slot t_k.

Figure 1 shows the methodology implemented to carry out the deployment of the CNN-LSTM neural network. This methodology consists of the stages of input data preprocessing, design of the neural network model, and training and validation of the inference model. The neural network model design stage is described in detail below (see Fig. 2), and the optimized parameters of the inference model are displayed in Table 1.

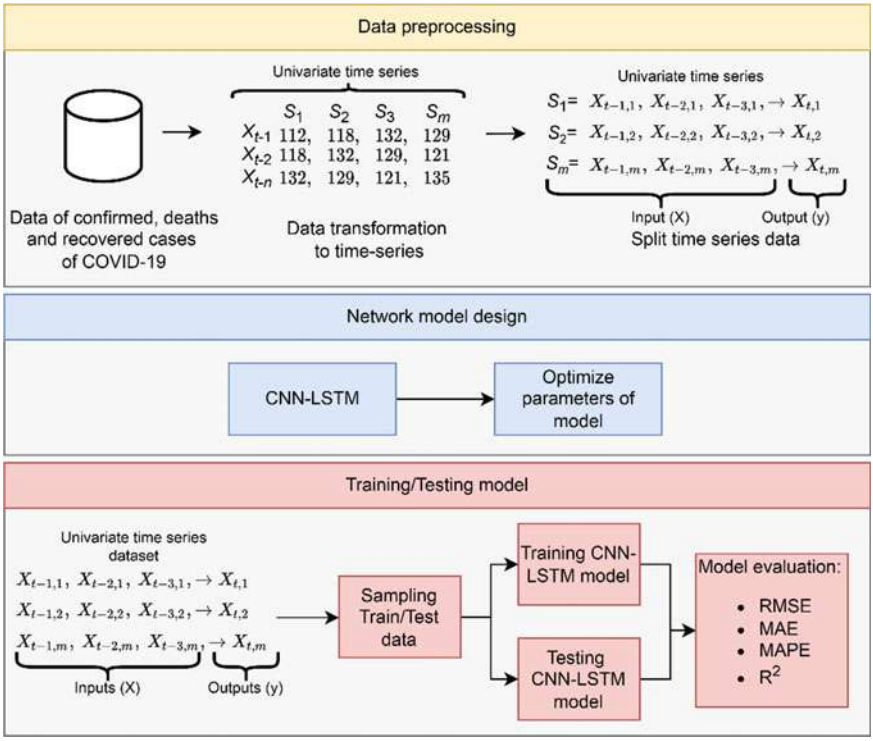


Fig. 1 View of the stages and activities that form the applied methodology

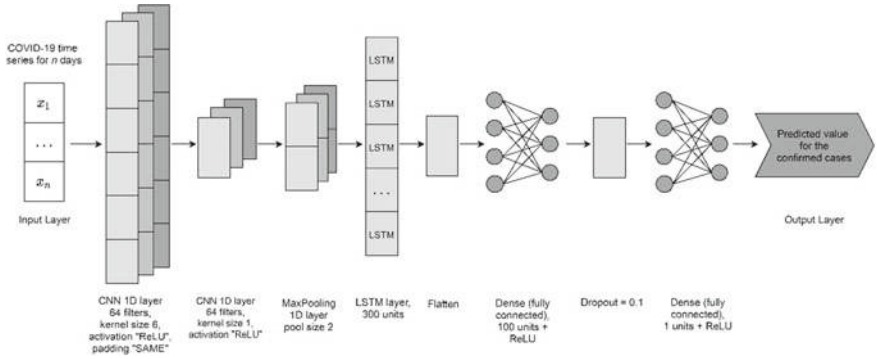


Fig. 2 Structure of the CNN-LSTM model implemented

Table 1 Parameters of the CNN-LSTM network

Model	Hyper-parameters	Selected values
CNN-LSTM	Conv1D filters	64
	Conv1D kernel size	6
	Conv1D activation	Relu
	Conv1D padding	Same
	Conv1D2 filters	64
	Conv1D2 kernel size	1
	Conv1D2 activation	Relu
	MaxPooling1D pool size	2
	LSTM units	300
	Flatten	-
	Dense1 units	100
	Dense1 activation	Relu
	Dropout	0.1
	Dense2 units	1
	Optimizer	Adamax
	Loss	Mse
	Batch size	64
	Epoch	500

The proposed inference model is based on the CNN-LSTM neural network, one of the most prominent DL architectures. The LSTM has shown significant performance on many real-world applications due to its ability to capture long-term dependencies [18, 19]. We design a model with two CNN layers, one max pooling layer, one LSTM layer, one flatten layer, one dense layer (with Relu activation function = 100), one dropout layer, and one dense layer. The CNN-LSTM model is connected in a recurrent network to combine the advantages of a single LSTM layer with two CNN layers, creating a model with the ability to learn higher-level temporal representations. The first CNN layer returns its complete output sequences, while the second layer returns the output sequence with high processing. Thus, LSTM takes the previous output from the CNN layer to calculate the memory for the subsequent LSTM blocks. The network is trained for 500 epochs, a batch size of 64, and an optimizer Adamax is used. Figure 2 shows the designed CNN-LSTM model. Table 1 shows all the parameters defined in the CNN-LSTM architecture used in our experiment.

In this study, we obtained the number of COVID-19 cases time series data in locations where an outbreak has been reported data released by Johns Hopkins University Center for Systems Science and Engineering (JHU-CSSE) [20]. The data repository built by JHU-CSSE includes real measurements of COVID-19 variables through daily reports [21]. Our experiment used three datasets corresponding to confirmed, deceased, and recovered COVID-19 cases, storing 480 records for each dataset. The outbreak locations are represented through a data structure built with attributes such

as province, country, longitude, latitude, and date fields (which contain the number of cases per day). Table 2 shows some sample data of global confirmed cases. The data analyzed in our study cover a period from March 03, 2021, to June 24, 2021. From the dataset, the data corresponding to five countries of the American continent was selected to experiment with the proposed CNN-LSTM model. A training sub-dataset with 80% of the observations from the original dataset is used to train our model. The remaining 20% of the instances (validation sub-dataset) are used to test our model's prediction precision.

4 Results

The CNN-LSTM network can identify trends in the variation in the data and classify the dependency relationship of the data in a time sequence. In the proposed hybrid model, this premise is applied to forecast future cases of persons infected with COVID-19 for the next days through time series. From the available dataset, we present data analysis of five highly representative scenarios of the COVID-19 transmission behavior.

The CNN-LSTM method is the inference model that has acquired knowledge or learned adequately in the training phase. The training errors are shown in Table 3 in terms of RMSE, MAE, and R^2. Then, with a model implemented from an input event or a sequence of input events, the CNN-LSTM inference model allows the prediction of the daily new cases. In our approach, the model automatically and sequentially inserts the value of confirmed cases for two consecutive days, predicting the value of cases that will occur on the next day (third day). The CNN-LSTM network does this on a recurring basis to build a complete time series from the prediction. Hence, time series data of COVID-19 allows estimation of the effect on Y of a change in X over time t_i. Therefore, using the testing sub-dataset, the inference model is validated.

Table 4 summarizes the performance obtained by the CNN-LSTM model in the test stage, represented by the error metrics. Confirming the highest prediction accuracy with which the CNN-LSTM model can predict possible new confirmed cases for the next day, where a smaller value indicates better model performance. For all countries, the test dataset is from March 11, 2021 to June 24, 2021. The values obtained in the performance of the prediction of confirmed cases for Mexico are very acceptable, with metrics of 1401.33, 1027.42, 0.39, and 0.91 for RMSE, MAE, MAPE, and R^2, respectively. CNN-LSTM model generates the lowest RMSE value for confirmed cases as 4040.47, while the MAE value is 2586.85 for Ecuador. The values obtained in Colombia's prediction model's performance are RMSE value of 4215.84 and MAE value of 3309.69. In all scenarios, RMSE and MAE metrics have a very acceptable value.

The smallest value of MAE is 1027.42 for Mexico among the five countries for the confirmed cases (see Table 4). As the number of cases is much more for Brazil, the error measures are also higher for this country than the rest of other countries in actual numbers. The values obtained in the measure performance R^2 very close

Table 2 Excerpt from the time series dataset of confirmed cases of COVID-19

Country-region	01/22/2020	01/23/2020	01/24/2020	01/25/2020	01/26/2020	01/27/2020	…	06/22/2021	06/23/2021	06/24/2021
Argentina	0	0	0	0	0	0	…	4,298,782	4,326,101	4,350,564
Brazil	0	0	0	0	0	0	…	18,054,653	18,169,881	18,243,483
China	4	9	24	0	0	0	…	1051	1051	1052
Colombia	0	0	0	0	0	0	…	3,997,021	4,027,016	4,060,013
Dominican Republic	0	0	0	0	0	0	…	318,368	319,254	320,136
Ecuador	0	0	0	0	0	0	…	447,176	449,107	449,483
France	0	0	0	0	0	0	…	26,647	26,799	26,799
Greece	0	0	0	0	0	0	…	418,943	419,455	419,909
Mexico	0	0	0	0	0	0	…	2,482,784	2,487,747	2,493,087
Norway	0	0	0	0	0	0	…	129,766	129,944	130,270

Table 3 Results of DL models during training stage

Country	RMSE	MAE	R^2
Mexico	5614.35	2420.38	0.99
Argentina	18,023.98	17,235.62	0.91
Brazil	58,990.42	51,314.43	0.95
Ecuador	2401.02	1169.44	0.93
Colombia	6420.27	3808.87	0.98

Table 4 Results of DL models during test stage

Country	RMSE	MAE	MAPE	R^2
Mexico	1401.33	1027.42	0.39	0.91
Argentina	14,878.25	11,608.44	4.19	0.94
Brazil	74,121.21	55,054.80	3.77	0.76
Ecuador	4040.47	2586.85	9.65	0.84
Colombia	4215.84	3304.69	3.07	0.99

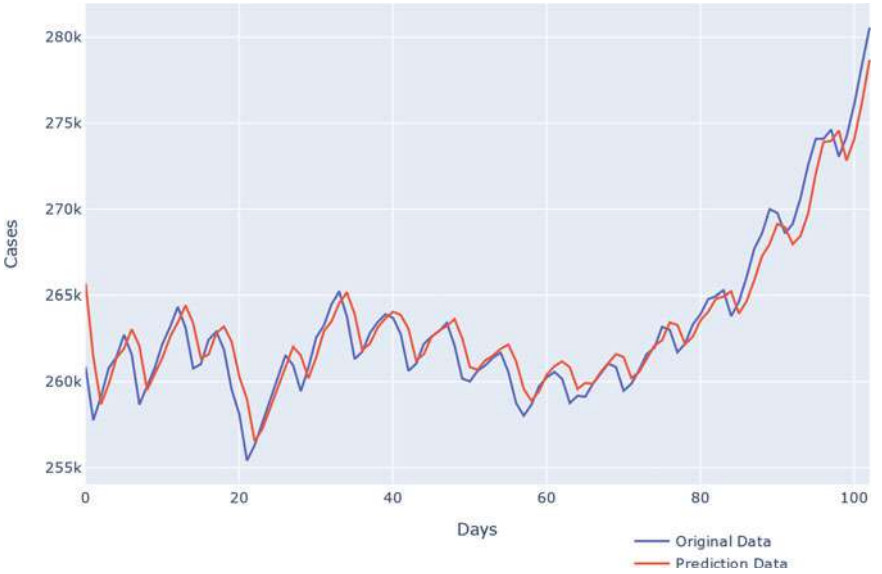


Fig. 3 Prediction results of the probable number of new confirmed cases for the next days for Mexico

to unity (except for the performance obtained for Brazil) indicate that the model is consistent, efficient, and accurate for all cases. The highest values for the R^2 are obtained for Argentina and Colombia with 0.94 and 0.99, respectively.

Fig. 4 Forecasting results for Argentina by using CNN-LSTM predictive model proposed

Figure 3 shows the results predicted by the CNN-LSTM model using data from confirmed cases in Mexico. The blue line indicates the real value, and the red line means the predicting value. The day number (enumerated from March 11, 2021) recorded in the dataset are indicated on the X-axis, and the Y-axis represents the number of confirmed COVID-19 cases. As shown in Fig. 3, both the original data and the prediction data of the CNN-LSTM model fit very well. It indicates that the CNN-LSTM model has the capability to capture the dependencies of the time sequence data and can exemplify the number of confirmed cases variation more accurately. Figure 4 shows the forecasting results utilizing the data of the cases confirmed in Argentina. We can see that although the difference between the two curves, the CNN-LSTM model can detect the variations correctly.

The proposed approach is based on the premise that any phenomenon that has a time component measured in a period can be represented as a time series to find patterns, detect atypical behaviors, and predominantly predict future behavior. In this sense, Figs. 5 and 6 illustrate the predicted new cases and real values with the available data of confirmed cases from Brazil and Colombia. In the specific case of the time series forecasting for Brazil (see Fig. 5), it can be seen that in the first 59 days, the original data and the forecast data fit very well. However, in the final part of the forecasting, the original data are separated from the predicting data, confirmed with the performance indicators RMSE = 74,121.21 and MAE = 55,054.80 (a high measure within the presented scenarios). Nevertheless, we consider that updating the dataset with records with recent dates will decrease the inference model's evaluation

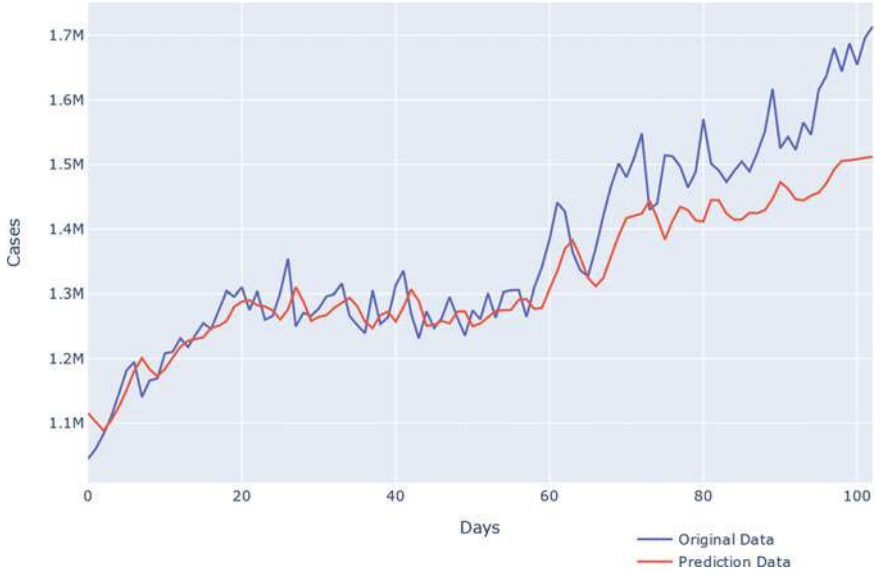


Fig. 5 Forecasting results for Brazil by using CNN-LSTM inference model

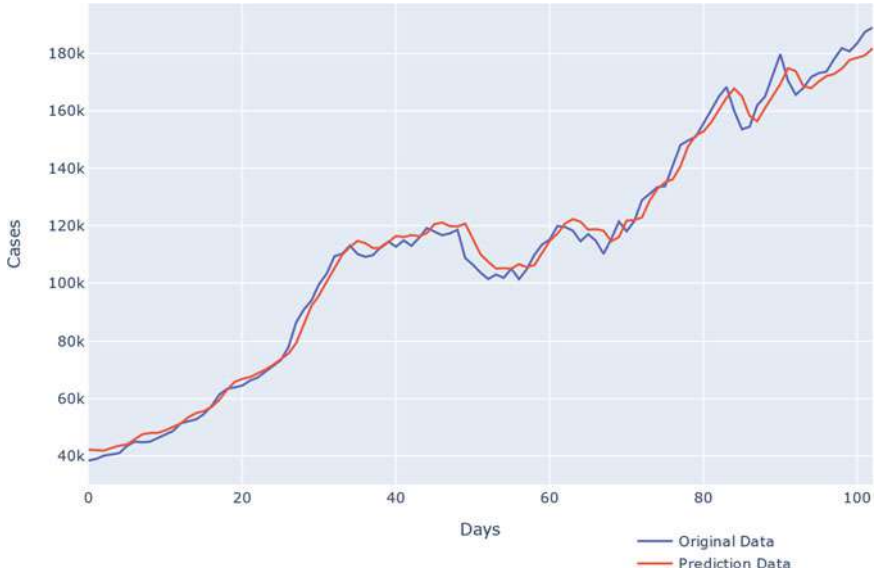


Fig. 6 Comparing original data and prediction of new cases for the next day for Colombia

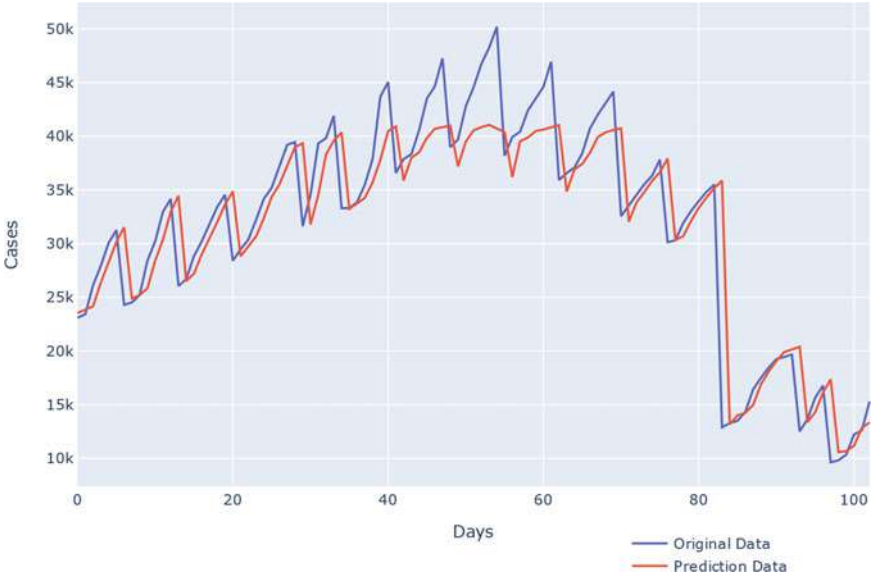


Fig. 7 Prediction of declining confirmed cases for Ecuador

indicators, with values close to the other three scenarios, improving the performance of the CNN-LSTM model for this scenario.

In Fig. 6, it is observed that the method predicts values closer to the real values in most of the days considered in the experiment. The experimental results show that the CNN-LSTM model is efficient in predicting new cases of COVID-19. From these five scenarios, it can be found that CNN-LSTM can better fit the sequence with the data periodicity.

The CNN-LSTM model can predict the peak of confirmed COVID-19 cases and the start of the decrease in the number of people with active infection. Figure 7 shows the prediction for confirmed cases per day for Ecuador. This Fig. shows a decrease in active cases; the decline is gradual, similar to the actual behavior identified in the pandemic in Ecuador. The CNN-LSTM model learned correctly in the training phase and could predict decline per day by using the confirmed cases, deaths, and recovered cases datasets.

5 Conclusions

The experimental results demonstrate that the proposed algorithm (CNN-LSTM model) has excellent generalization capability. Our experiment addresses the challenges of data analytics of the time series in the COVID-19 problem. The experimental results based on key indicators suggest that the proposed CNN-LSTM model

achieves a fit very well on the prediction of the confirmed cases and can be applied when data have high complexity in their behavior and variation in its extension.

The results of the prediction obtained in our experiment indicate that the proposed CNN-LSTM model is valid for the forecasting of COVID-19 confirmed cases, with better performance than other models in terms of precision and stability. The model reached the lowest values metric values of RMSE 1401.33, MAE of 1027.42, and MAPE of 0.39, and the maximum value of R^2 was 0.99. The time series forecasting is very acceptable in the scenario of Brazil, despite the values obtained in RMSE of 74,121.21 and MAE of 55,054.80. The predicted data is very close to the original data and maintains a similar behavior throughout the experiment. In our experiment, the cyclical, seasonal, and noise possible components were not smoothed, as it is recommended to do in some forecasting approaches to maintain the general trend.

There are typically many factors outside of control in forecasting scenarios, which could affect future values substantially. Therefore, an accurate, effective, and stable prediction model is necessary to promote public health strategies to decrease the number of confirmed cases in the coming days, notwithstanding the negative impacts of the outbreak. There are several challenges to COVID-19 data analyses, particularly in real-time. The time series forecasting is preliminary and subject to several limitations. However, it is essential to calculate the possible new cases of the COVID-19 in the early stage of the outbreak and predict daily cases for a city or country. The data-driven predictions can support making decisions to decrease the impact of the epidemic. Finally, the CNN-LSTM model can serve as an alternative tool with reliable prediction to support preventive decision-making to decrease the pandemic's impact.

References

1. World Health Organization (2020) Report of the WHO-China joint mission on coronavirus disease 2019 (COVID-19). https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-nal-report.pdf. Last accessed 17 Mar 2021
2. WHO (2021) Weekly epidemiological update on COVID-19—14 September 2021. https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---14-september-2021. Last accessed 20 Sep 2021
3. Huang NE, Qiao F (2020) A data driven time-dependent transmission rate for tracking an epidemic: a case study of 2019-nCoV. Sci Bull 65(6):425–427. https://doi.org/10.1016/j.scib.2020.02.005
4. Li Q, Guan X, Wu P, Wang X, Zhou L et al (2020) Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia. N Engl J Med 382(13):1199–1207. https://doi.org/10.1056/NEJMoa2001316
5. Bogoch II, Watts A, Thomas-Bachli A, Huber C, Kraemer MUG, Khan K (2020) Pneumonia of unknown aetiology in Wuhan, China: potential for international spread via commercial air travel. J Travel Med 27(2):1–3. https://doi.org/10.1093/jtm/taaa008
6. Du Z, Wang L, Cauchemez S, Xu X, Wang X, Cowling BJ, Meyers LA (2020) Risk for transportation of coronavirus disease from Wuhan to other cities in China. Emerg Infect Dis 26(5):1049–1052. https://doi.org/10.3201/eid2605.200146

7. Wu JT, Leung K, Leung GM (2020) Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. Lancet 395(10225):689–697. https://doi.org/10.1016/S0140-6736(20)30260-9
8. Huang C, Wang Y, Li X, Ren L, Zhao J et al (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan China. Lancet 395(10223):497–506. https://doi.org/10.1016/S0140-6736(20)30183-5
9. Ghinai I, McPherson TD, Hunter JC, Kirking HL, Christiansen D et al (2020) First known person-to-person transmission of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) in the USA. Lancet 395(10230):1137–1144. https://doi.org/10.1016/S0140-6736(20)30607-3
10. Chen TM, Rui J, Wang QP, Zhao ZY, Cui JA, Yin L (2020) A mathematical model for simulating the phase-based transmissibility of a novel coronavirus. Infect Dis Poverty 9(1):1–8. https://doi.org/10.1186/s40249-020-00640-3
11. Kucharski AJ, Russell TW, Diamond C, Liu Y, Edmunds J et al (2020) Early dynamics of transmission and control of COVID-19: a mathematical modelling study. Lancet Infect Dis 20(5):553–558. https://doi.org/10.1016/S1473-3099(20)30144-4
12. Rauf HT, Lali MIU, Khan MA, Kadry S, Alolaiyan H et al (2021) Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks. Pers Ubiquit Comput. https://doi.org/10.1007/s00779-020-01494-0
13. Wang P, Zheng X, Ai G, Liu D, Zhu B (2020) Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: case studies in Russia, Peru and Iran. Chaos, Solitons Fractals 140(110214):1–8. https://doi.org/10.1016/j.chaos.2020.110214
14. Tomar A, Gupta N (2020) Prediction for the spread of COVID-19 in India and effectiveness of preventive measures. Sci Total Environ 728(138762). https://doi.org/10.1016/j.scitotenv.2020.138762
15. Manojkumar VK, Dhanya NM, Prakash P (2021) A hybrid deep learning approach for predicting the spread of COVID-19. In: Paprzycki M, Thampi SM, Mitra S, Trajkovic L, El-Alfy ESM (eds) Intelligent systems, technologies and applications, pp 193–204. Springer, India. https://doi.org/10.1007/978-981-16-0730-113
16. Gautam Y (2021) Transfer learning for COVID-19 cases and deaths forecast using LSTM network. ISA Trans. https://doi.org/10.1016/j.isatra.2020.12.057
17. Islam Z, Islam M, Asraf A (2020) A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Inform Med Unlocked 20(100412). https://doi.org/10.1016/j.imu.2020.100412
18. Pham T, Tran T, Phung D, Venkatesh S (2017) Predicting healthcare trajectories from medical records: a deep learning approach. J Biomed Inform 69:218–229. https://doi.org/10.1016/j.jbi.2017.04.001
19. Maragatham G, Devi S (2019) LSTM model for prediction of heart failure in big data. J Med Syst 43(5):1–13. https://doi.org/10.1007/s10916-019-1243-3
20. Dong E, Du H, Gardner L (2020) An interactive web-based dashboard to track COVID-19 in real time. Lancet Infect Dis 20(5):533–534. https://doi.org/10.1016/S1473-3099(20)30120-1
21. JHU CSSE (2021) COVID-19 data repository by the center for systems science and engineering (CSSE) at Johns Hopkins University. https://github.com/CSSEGISandData/COVID-19. Last accessed 26 Aug 2021

Chapter 41
Photonic Upconversion-Based Millimeter Wave Generation and Transmission for 5G RoF Fronthaul System

Asha and Sandeep Dahiya

1 Introduction

The MM Wave Radio over Fiber (RoF) system has emerged as a viable approach for fulfilling ever-increasing traffic demands required for 5G networks by offering mobile users with multi-gigabit services. However, generating high frequency MM wave with low cost base station are such two main technologies which make MM wave RoF system practical for commercial deployment [1]. MM wave signal can be generated through two approaches: one is electrical approach and another is optical or photonic approach. MM wave generation through electrical approach is quite troublesome due to limited frequency response of devices and high cost [1]. Hence, a lot of research work has been done recently on photonic MM wave generation with transmission over the fiber-based system because RoF provides numerous advantages [2]. Out of the various techniques mentioned in the literature [3–10], frequency multiplication techniques based on external modulation through Mach Zehnder Modulator (MZM) is the most appropriate one due to its higher flexibility, larger tunable range, higher modulation bandwidth, higher stability [5]. Photonic frequency multiplication techniques with different frequency multiplication factor (FMF) such as 2, 3, 4, 5, 6, 8, 10, 12, 14 and 16 based on double sideband with carrier (DSB-C) and double sideband with suppressed carrier (DSB-SC) has been demonstrated. In these multiplication techniques, the electrical signal obtained at the photodetector in the base station is a multiple of the LO frequency depending on the FMF. For example, for 40 GHz MM wave signal generation with FMF of 4, the LO frequency required is of 10 GHz.

The MM wave generation system with high FMF and suppressed carrier techniques are very efficient as they not only eliminates the requirement of high frequency

Asha (✉) · S. Dahiya
Department of ECE, Faculty of Engineering and Technology, BPS Women University, Khanpur Kalan, Sonepat 131305, India
e-mail: ashabalhara89@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_42

LO at the central station (CS) but also provide improved unwanted sideband suppression, power efficient and dispersion tolerant transmission. For achieving high FMF, several series parallel configuration of MZM have been utilized as it is not possible to select and generate higher order sidebands with single MZM. Various 16 tupling techniques have been reported by utilizing dual cascaded MZM [7], dual parallel MZM [8], two MZM [9], and cascaded MZM [10]. Though the above mentioned configuration realized frequency 16 tupling but their setups are complicated and resulted into lower sideband suppression with limited transmission distance of maximum up to 25 km as shown illustrated in Table 1. Therefore, there is a need to design a system that can photonically generate high frequency signal through low RF signal with higher sideband suppression ratios resulting into a high spectral purity signal and could also cover larger transmission distance through fiber from CS to BS.

In this paper, a new filterless frequency 16 tupled MM wave generation scheme is proposed based on quad dual drive MZM configuration. By properly controlling the phase shift of the RF signal driving the DD-MZM, RF Local Oscillator (LO) voltage and modulation index, the unwanted sidebands can be completely eliminated resulting in to a higher OSSR of 65 dB, RSSR of 53 dB and transmission distance of up to 40 km through 2.5 Gbps data modulation.

The rest of the paper is organized as follows: A mathematical model for MM wave OCS modulation with is presented in Sect. 2. The simulation set up of the proposed structure with the transmission process along the fiber is discussed in Sect. 3 and conclusion in Sect. 4.

Table 1 Summary of various frequency 16 tupling techniques

Reference	Structure	OSSR (dB)	RSSR (dB)	Fiber length	Observation
[7]	2 cascaded dual parallel MZM	21.5	38	–	System sensitive toward modulation index (MI)
[8]	Parallel combination of two cascaded MZM	20	28	25 km	Costly and complex structure
[9]	Dual stage cascaded MZM	60	48	20 km	High system sensitivity
[10]	Two cascaded MZM	31.35	24.11	–	High RF voltage required
[11]	Cascaded MZM	54	42	25 km	Infinite extinction ratio (ER): practically unrealizable
[12]	Cascaded parallel MZM	64	31	–	Complex structure

2 Operating Principle

The schematic representation of the MM wave-based RoF system using quad cascaded MZM configuration is depicted in Fig. 1 which consists of one continuous wave laser (CWL) source, four dual drive MZM (DD-MZM), one radio frequency local oscillator (LO), three electrical phase shifters, two erbium doped fiber amplifiers (EDFA), one optical amplifier, one polarization controller, one SSMF, photodetector, BPF, mixer, LPF. There are three parts of this structure (a) central station, (b) optical fiber link and (c) base station. In central station, a CWL and RF LO drives the quad cascaded MZM. The biasing of all the four MZM is done so as to bias them at the peak transmission point (PTP) such that only even order sidebands are generated at the output of modulator.

The continuous light wave from continuous light (CWL) source acting as a carrier signal is represented by:

$$E_{\text{in}}(t) = E_c \mathrm{e}^{j\omega_c t} \tag{1}$$

where E_c and ω_c are the amplitude and frequency of optical carrier signal.

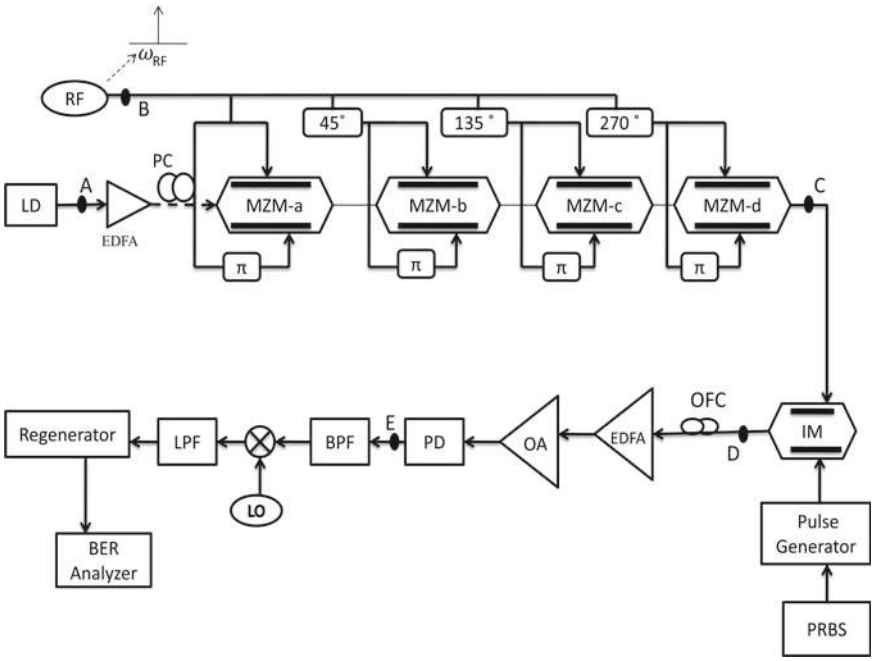


Fig. 1 Schematic diagram of the proposed frequency 16 tupled fiber based transmission system where LD: laser diode, IM: intensity modulator, RF: RF source, BPF: bandpass filter, LPF: low pass filter, LO: local oscillator, PD: photodetector

The RF signal with frequency ω_R and amplitude V_R driving the MZM-a is expressed by

$$V_a = V_R(\mathrm{Cos}\omega_R t) \tag{2}$$

The output from MZM-a is denoted as [2]:

$$E_{\mathrm{MZM}-a}(t) = \frac{E_c \mathrm{e}^{j\omega_c t}}{10^{\alpha/20}} \left\{ \chi \mathrm{e}^{\left\{ j\pi \frac{V_2(t)}{V_\pi} + j\pi \frac{V_{b2}}{V_\pi} \right\}} + (1-\chi)\mathrm{e}^{\left\{ j\pi \frac{V_1(t)}{V_\pi} + j\pi \frac{V_{b1}}{V_\pi} \right\}} \right\} \tag{3}$$

where $V_1(t)$ and $V_2(t)$ are used to drive the DD-MZM such that $V_1(t) = V_R\mathrm{Cos}\omega_R t$ and $V_2(t) = V_R\mathrm{Cos}\omega_R t + \Phi$. V_{b1} and V_{b2} are the DC bias voltages applied to the two arms of DD-MZM, respectively. V_π is the half wave voltage, α is the insertion loss and χ is the splitting ratio. Considering large extinction ratio, small insertion loss and $V_{b2} = 0$ for DD-MZM, then Eq. 3 can be re-written as [4]:

$$\begin{aligned} E_{\mathrm{MZM}-a}(t) = \frac{E_c}{2} \Bigg\{ & \mathrm{Cos}\left(\omega_c t + \frac{\pi V_R}{V_\pi} \cos(\omega_R t + \Phi)\right) \\ & + \mathrm{Cos}\left(\omega_c t + \frac{\pi V_R}{V_\pi} \cos(\omega_R t) + \frac{\pi V_{b1}}{V_\pi}\right) \Bigg\} \end{aligned} \tag{4}$$

where $m = \frac{\pi V_R}{V_\pi}$ is the modulation index. Different modulation schemes can be generated by simply adjusting DC bias voltage between the two arms of DD-MZM. When $\Phi = \pi$ and $V_{b1} = V_{b2} = \frac{V_\pi}{2}$ such that the net bias voltage $V_b = V_{b1} - V_{b2} = 0$, then only even order sidebands are generated at the output of MZM-a. This output of this modulator is then inserted to MZM-b. The electrical phase shift of RF signal driving this configuration is kept in such a manner that there is 0°, 45°, 135° and 270° phase drift of RF signal with MZM-a, MZM-b, MZM-c, MZM-d, respectively, and modulation index, m, is kept to be 2.82 with $V_R = 3.59995$ so that there are only $+8$ and -8 order sidebands left at the output of fourth modulator and all other sidebands are completely eliminated as discussed in detail in [13] and is represented as:

$$E_{\mathrm{MZM}-d}(t) = \frac{\alpha^4}{2} J_8^2(m) E_c \mathrm{e}^{j\omega_c t} \left\{ \mathrm{e}^{j8\omega_R t} + \mathrm{e}^{-j8\omega_R t} \right\} \tag{5}$$

From Eq. 5, it can be seen that the frequency spacing between the two sidebands is 16 times the input RF. Then 2.5 Gbps NRZ data is intensity modulated on both the sidebands through the MZM and transmitted over the SSMF to the base station (BS). Then the optical signal after passing through a hybrid combination of amplifiers for eliminating the transmission losses is photodetected by a PIN photodetector and its photocurrent is expressed as:

$$I_{\mathrm{ph}}(t) = \mu_{\mathrm{ph}} |E_{\mathrm{MZM}-d}(t)| * |E_{\mathrm{MZM}-d}(t)|^* \tag{6}$$

$$I_{\text{ph}}(t) = \mu_{\text{ph}} \left\{ \frac{\beta}{2} E_c \left[J_{-8}(m) \mathrm{e}^{j(\omega_c - 8\omega_{\text{R}})t} + J_8(m) \mathrm{e}^{j(\omega_c + 8\omega_{\text{R}})t} \right] \right\} * \left\{ \frac{\beta}{2} E_c \left[J_{-8}(m) \mathrm{e}^{j(\omega_c - 8\omega_{\text{R}})t} + J_8(m) \mathrm{e}^{j(\omega_c + 8\omega_{\text{R}})t} \right] \right\}^* \tag{7}$$

where μ_{ph} being the responsivity of the photodiode. The above equation can be further simplified as:

$$I_{\text{ph}}(t) = \frac{\mu_{\text{ph}} \beta^2 |E_{\text{MZM}-d}(t)|^2}{4} \text{Cos}(16\omega_{\text{R}} t) \tag{8}$$

This optical MM wave gets converted into electrical signal through PIN photodiode. While transmitting through the SSMF, fiber chromatic dispersion takes place due to which data which is used for modulation will lose synchronization and the two eighth order modulated sidebands are transmitted with different velocities in the fiber and thus suffers from bit walk off effect. The edges of the data shifts with the increasing transmitting distance which ultimately degrades the quality of the signal. The difference in time between the two eighth order sidebands carrying the dispersed codes is stated as time shift denoted by $\Delta t_{-8,+8}$. When this time shift approaches code duration, W, i.e., $(\Delta t_{+8-8} \rightarrow W)$ where $W = \beta / R_b$ then the eye diagram completely closes. The transmission distance is given by Eq. 9 [2]:

$$T_D = \frac{\beta c}{8 R_b \lambda_c^2 D f_{\text{R}}} = \frac{W c}{8 \lambda_c^2 D f_{\text{R}}} \tag{9}$$

where $c = 3 * 10^8$ m/s is the speed of light, radio frequency $f_{\text{R}} = 10$ GHz, fiber dispersion $D = 15.95$ ps/nm-km, wavelength of the carrier signal $\lambda_c = 1552.5$ nm, code coefficient $\beta = 1$ for NRZ coding and bit rate $R_b = 2.5$ Gbps. After carrying out calculation, it was found that the longest transmission distance is 40 km. Hence, the eye closes completely as the transmitting distance approaches T_D.

3 Simulation Results and Discussion

The transmission performance of the proposed structure is examined by intensity modulating 2.5 Gbps data on both the sidebands over the SSMF with the help of Optisystem simulation platform. A polarization controlled CW laser source with a central frequency of 193.1 THz with 10 MHz spectral width and 5 dBm power after passing through an EDFA having gain of 20 dB and noise figure of 4 dB is launched into the quad cascaded MZM configuration for generating eighth order sidebands. The upper arm of MZM-a, MZM-b, MZM-c and MZM-d is driven by 10 GHz RF LO with 0°, 45°, 135° and 270° phase shifted frequency, respectively. The output from CWL at point A and RF LO output at point B in the proposed structure is shown in

Fig. 2a, b, respectively. All the MZM are operated in their PTP so that the odd order sidebands are suppressed for which the biasing voltages ($V_1 - V_2$) are set to zero and the switching voltages are set to 4 V. The extinction ratios (ER) of all four MZMs are set to 70 dB for suppressing the unwanted sidebands with insertion loss of 4 dB. The output from MZM-d shows an optical carrier suppressed signal (OCS) which consists of only eighth order sidebands at 193.02 and 193.18 THz which are 160 GHz apart and 65 dB higher than the second order sidebands and all other sidebands are eliminated due to low power as shown in Fig. 2c. This OCS signal is then modulated on both the sidebands with 2.5 Gbps non return to zero (NRZ) data with the help of an intensity modulator (IM) and the optical spectrum from the IM output is shown in Fig. 2d. The data bearing optical signal is then transmitted to the BS through SSMF over 0, 5, 10, 15, 30, 40 and 50 km long optical fiber with 15.95 ps/nm-km dispersion coefficient and attenuation coefficient of 0.25 dB/km.

At the BS, prior to optical to electrical conversion it is passed through a hybrid amplifier configuration consisting of 5 m long EDFA and an optical amplifier having 10 dB gain used for compensating the losses occurred during transmission. At the base station, a PIN photodetector of 0.9 A/W responsivity and 10 nA dark current is used to detect the modulated signal and convert back into an electrical signal. The photodetected signal consists of 160 GHz data modulated signal having RSSR of 53 dB and the baseband spectrum which is shown in Fig. 2e. For separating the data modulated over the 160 GHz MM wave the detected signal is then passed through a band pass filter (BPF) with bandwidth of 1.5 times the symbol rate. An electrical mixer is used for down converting and demodulating the signal which is then filtered through a LPF having cut off frequency 0.75 times the bit rate and then sent to a BER analyzer for testing after passing through a regenerator.

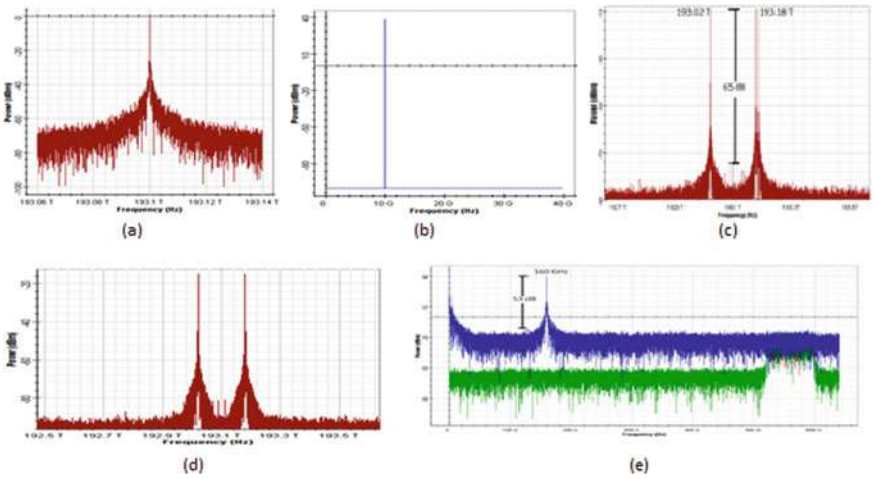


Fig. 2 **a** Optical carrier signal from CWL at point A **b** RF signal of 10 GHz at point B **c** MZM-4 output constituting eighth order sidebands at point C **d** data modulated signal after intensity modulation with 2.5 Gbps at point D **e** photodetected signal at point E in the proposed structure

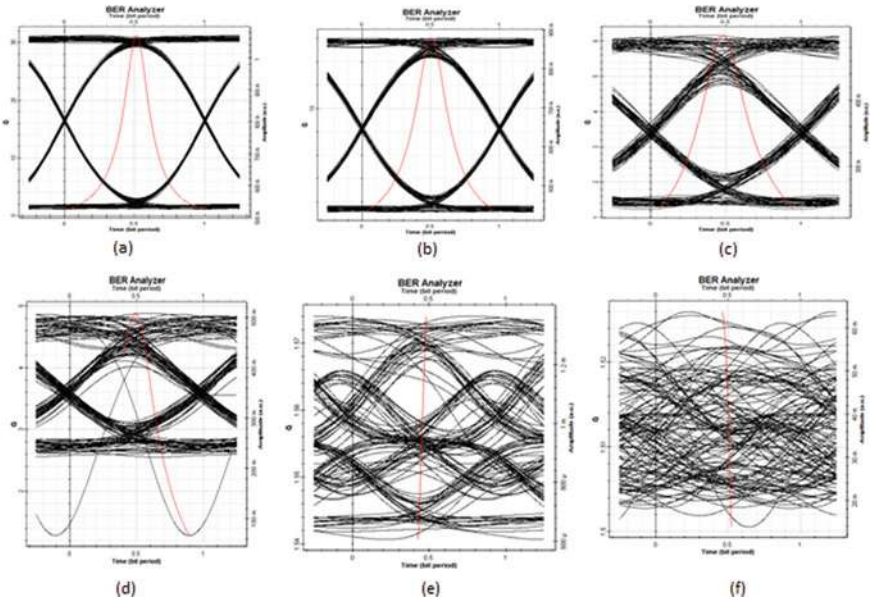


Fig. 3 Eye diagrams of the baseband signal demodulated from 160 GHz at various fiber lengths **a** B-T-B with 31.2285 Q factor **b** 5 km with 16.0173 Q factor **c** 10 km with 10.16061 **d** 30 km with 4.5536 **e** 40 km with 2.329 **f** 50 km with 0.875 Q factor

The system performance is measured in terms of eye diagrams over various fiber lengths. The eye diagrams at various lengths are shown in Fig. 3. It is observed that with the proposed quad MZM configuration and using hybrid amplifier (EDFA + Optical amplifier) the transmission distance of this double sideband modulation as compared with the traditional approaches in [6–10] is improved but it is still limited by bit walk off effect and it degrades the system BER performance.

With the increase in transmission distance, the eye carried by 160 GHz signal closes gradually. The eye almost closes after the optical mm wave is transmitted over 40 km due to the code edge shift, while the eye will fully close after transmission over 50 km which goes well with our estimation in Eq. 7. However, this bit walk effect can be compensated by employing dispersion compensation fiber (DCF) during transmission or by employing single tone modulation instead of dual tone modulation by filtering out one of the sidebands by using Fiber Bragg Gratings (FBG) to be examined in our next research work.

4 Conclusion

The transmission performance of filterless frequency 16-tupled optical MM wave generated through OCS technique with 2.5 Gbps intensity modulation was theoretically studied in this paper. With appropriate adjustment of MZM biasing, RF phase shift, RF voltage, quad MZM and hybrid amplifier configuration the unwanted harmonics can be completely suppressed resulting into a 160 GHz MM wave with higher OSSR of 65 dB and RSSR of 53 dB. The transmission distance can be enhanced up to 40 km by using this technique as compared to previous research work. Further, it was found that the data signals carried on both the sidebands by OCS MM wave signal suffer from time shifting due to fiber chromatic dispersion as the transmission distance was increased, thus limiting its transmission distance and degrading the signal quality.

References

1. Ma J, Xin X, Yu C, Zhang Q, Yu J, Sang X, Zeng J (2007) Milimeter-wave optical subcarrier generation by using an external modulator and optical carrier suppression. In: IEEE ICTON, pp 273–276
2. Chen X (2020) Improved filterless sextupling optical MM-wave generation and RoF transmission. Int J Light Electron Opt 20(6):1–5
3. Zeb K (2020) Photonic generation of spectrally pure millimeter wave signals for 5G applications. In: IEEE international topical meeting on microwave photonics. Ottawa, Canada, pp 1–4
4. Perez Galacho D, Sartiano D, Sales S (2019) Analog radio over fiber links for future 5G radio access networks. In: IEEE ICTON, pp 1–4
5. Asha, Dahiya S (2020) A comprehensive review of millimeter wave based radio over fiber for 5G front haul transmissions. Indian J Sci Technol 14(1):86–100
6. Kamissoko D, He J, Ganame H, Tall M (2020) Performance investigation of W-band millimeter wave radio over fiber system employing optical heterodyne generation and self homodyne detection. Opt Commun 474(1):1–8
7. Zhu Z, Zhao S, Chu X, Dong Y (2015) Optical generation of millimeter wave signals via frequency 16 tupling without an optical filter. Opt Commun 354(1):40–47
8. Esakki Muthu K, Sivanantha Raja A (2016) Bidirectional MM-wave radio over fiber transmission through frequency dual 16-tupling of RF local oscillator. J Eur Opt Soc 12(24):1–9
9. Sivanantha Raja A, Shanmugapriya G, Essaki Muthu K (2017) Frequency 16 tupled optical millimeter wave generation using dual cascaded MZMs and 2.5 Gbps RoF transmission. Optics 30(1):338–346
10. Wang D, Tang X, Xi L, Zhang X, Fan Y (2019) A filterless scheme of generating frequency 16-tupling millimeter-wave based on only two MZMs. Opt Laser Technol 116(1):7–12
11. Baskaran M, Prabakaran R (2018) Optical millimeter wave signal generation with frequency 16 tupling using cascaded MZMs and no optical filtering for radio over fiber system. J Eur Opt Soc 14(13):1–8
12. Dar AB (2020) Filterless 16-tupled optical millimeter-wave generation using cascaded parallel Mach-Zehnder modulators with extinction ratio tolerance. Prog Electromagn Res Lett 91(1):129–135
13. Asha, Dahiya S (2021) Large tunable 16-tupled millimeter wave generation utilizing optical carrier suppression with tunable sideband suppression ratio. Front Phys 1–9

Chapter 42
Customizing Backend Logic Using a Chatbot

Shounak Bhattacharya, Abhishek Kushwaha, Sharmila K. Banu, and B. K. Tripathy

1 Introduction

A developer occasionally would get extremely drawn-out and tedious while working on creating backend structures for various sites [1, 2] or even applications like applications of IOT gadgets [3–5]. The whole interaction may be tedious and an excessive amount of time burning-through. Automating this tedious development work into a conversational experience is the fundamental goal.

An attempt has been made to put together a demonstration that provides a chatbot interface to assist a developer build along the backend logic for different other tools. Using user defined routes based on the conversational interface, the schema can be defined. This enables the utility of a text to speech interface, with multilingual inputs as well to enable users to create tables as per their convenience.

The main motivation for this idea is to utilize the fact that the semantics and syntax of different functional as well as object-oriented programming languages like python allows people to work on code that can be autocomplete or semantically guessed or built up based on certain rules, like the Hidden Markov Model. Chatbots are computer programs that simulate human conversation through voice commands or text chats or both. These are artificial intelligence (AI) features that can be embedded and used through any major messaging applications. Many applications of chatbots are found in the literature [6–8].

It could work on this logic to make a simple neural network model [9] that could interpret this human thought into logic, hence a chatbot with intents different from normal chatbots.

S. Bhattacharya · A. Kushwaha · S. K. Banu
School of Computer Science and Engineering, VIT, Vellore, TN 632014, India
e-mail: sharmilabanu.k@vit.ac.in

B. K. Tripathy (✉)
School of Information Technology and Engineering, VIT, Vellore, TN 632014, India
e-mail: tripathybk@vit.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_43

Furthermore, this chatbot is not just limited to synthesizing human speech, but there is also a need to build a model, to translate one's requirements into tangible first order logic that can be correspondingly converted to the machine code written by a developer.

2 Literature Review

The idea was thought of after a thorough research in this domain. The deployment and automation on a cloud infrastructure along with its maintenance is managed in Gropengießer and Sattler [10]. In Sulakhe et al. [11] the Genome Sequencing stages have been automated along with database storage, and storage of results and annotations. Domain specific chatbots have been the talk of the town ever since the ecommerce steam rolled forward and banking developed its niche on the digital foray. With Liu and Dong's idea of enabling a chatbot for university-based activities and utilizing the human mind as a knowledge base in Liu and Dong [12]; the immense capabilities that a chatbot could bring in day to day processes was realized. Since chatbots utilize many tools of utilizing user data and concepts entwined with state-of-the-art technology, it creates the platform for exploring a host of endless possibilities. The idea of customizing backend logic, according to user needs is an absolutely novel idea where radical improvements in time utilization by the developer by several exponents using basic natural language processing techniques agglomerated with logic developed using different backend framework supporting intelligence autocomplete as the medium to make some useful backend logic tailored for user needs. The ultimate goal hence is to raise the developer niche equilibrium higher and spur developers to be able to develop their products faster, with essential support and essential code, customized to user convenience. In Chugh et al. [13], the research focuses on techniques of developing HTML pluggable components that can be prepared using similar methods. In Hwang et al. [14], a data driven approach was used employing customer service chatbots and analysis of cost per click of data. Through [15] a Dialog tool was used that allows the intended users to obtain the structure, semantics and behavior of a database. The main information being extracted is the data conveyed by the database designer and subsequent discussions are made on how syntax and semantics of the natural language input can be analyzed or interpreted and then transferred on to a special EER model. These transformations are supposed to yield a rough, barebones skeletal design including structural and semantic constructs as well as behavior. Uma et al. [16] introduced the need for a system that enables the users to access the information in the database, and instead of the normal textural input governed by the usual syntaxes, it aims to develop such a system using NLP by giving structured natural language question as input and receiving SQL query as the output, to access the related information from the railways reservation database with ease. The steps involved in this process are tokenization, lemmatization, parts of speech tagging, parsing and mapping. It gave an overall view of the usage of natural language processing (NLP) and use of

regular expressions to map the query in the English language to SQL. The work in Husár et al. [17] deals with retargetable compiler generation, with an initial onset to application-specific instruction set processor design. Automatic approach to instruction semantics extraction from ISAC models which result is usable for backend generation is presented. This approach was successfully tested on three models of MIPS, ARM and TI MSP430 architectures with further development and research revealing that extracted instruction is semantics and this was tested on the MIPS architecture and some preliminary results were highlighted. A theoretical approach to generate a mock backend that minimizes the dependency between frontend developers and backend developers and behaves exactly like the actual backend is dealt with in Singhal and Jain [18]. In Zamanirad et al. [19] each state covers intent and contains a nested state machine to help manage tasks associated with the conversation intent.

In this tool emphasis has been made on the natural language pipeline, the association with how a conversational interface could work with handling text from different regional languages. A focus on backend creation followed by creation of routes, retrieval and manipulation tasks with custom pipelining is one of the major goals of this paper. The motive is to create a foundation block to automate the different layers from the user to the end data starting with the backend.

3 Methodology

A careful procedure and framework was crafted for efficiently executing this task.

The below architecture model referred in Fig. 1 elucidates the flow of operations from the user's input across the chatbot layers to an intermediary logic processor interacting with the logic tables and the backend.

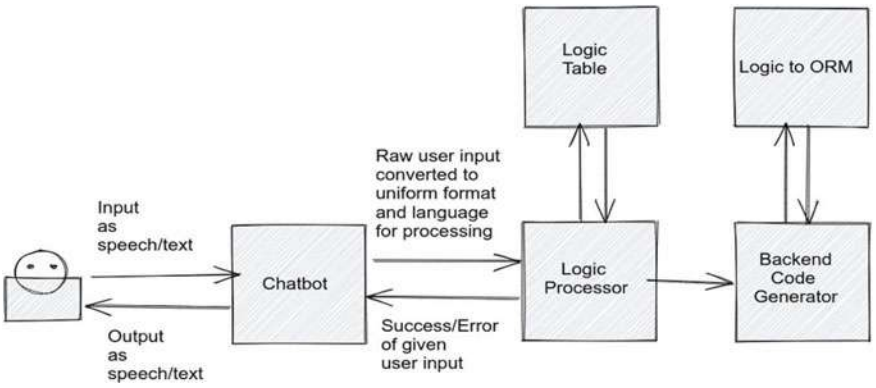


Fig. 1 Process architecture

The below architecture has been further broken down into 2 parts—the first part handles the chatbot, and the second one deals with the backend logic, both of which are explained in detail, in subsequent sections below.

This is a schematic flow of the entire proposed system that is supposed to give a high-level overview. In this diagram, the user has the freedom of giving inputs both in text and audio format.

The chatbot is the first point of contact to the user input. It internally interprets the speech/text to a common format that can be consumed by the proceeding Logic Processor. The chatbot is a simple pluggable component of the system, aimed at analyzing and converting any form of input into the format suitable for the logic processor. This ensures that additional features/inputs on the user side can be appended without reworking the architecture entirely, i.e., it ensures maintainability and extensibility.

Similarly, the Logic processor uses the data obtained to further generate 'tasks' for the backend generation using the logic table as a collection of templates for the basic task layout. These 'tasks' are logical units of code that can be used to deal with the system, and these tasks are designed in such a way that they will be atomic.

These atomic tasks, in their order of sequence, can still have a mismatch/inconsistent state, because the user might just issue inputs that might be conflicting. For example, imagine a task where the user intends to query a table by descending order of its primary id, but the id fields might be UUID fields that do not have a specific order, etc., or it might just be a schema flaw, like comparing users based on the 'age' attribute in the table, whereas they might not even exist. Hence, this is where the backend generator decides conflicts and resolves it for the system to be consistent.

The consequent feedback regarding the inconsistency of the states is reported back to the chatbot which then reports back to the user in the appropriate format.

3.1 Training the Chatbot

Based on Fig. 2 the design consisted of a comprehensive set of intents covering all facets that the chatbot needs to function properly.

Intent Defined Pipeline

1. The user can choose a unified language like English or enter in some other language and translate to this language.
2. Based on this, backend routes can be created based on user choice.
3. Each tag can eventually be taken as a label.
4. Label was converted into a numeric form from 0 to 9. The goal was to tokenize the text. Activation of OOV—out of vocabulary tokens that actually take care of words that do not come in the set of the vocabulary and get appended right at the beginning as the first element. Transformation sequencing and padding was performed.

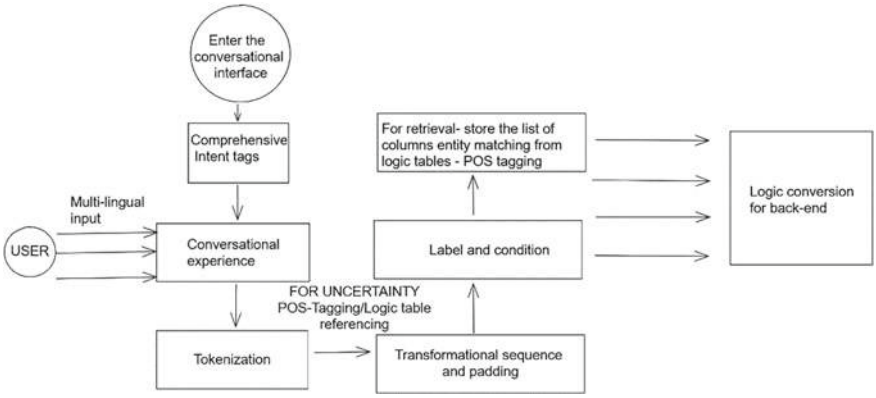


Fig. 2 Conversational pipeline

5. The model was trained with the labels and then pathways were created for the execution of the chatbot. Different conditions were defined to create a new project, addition of a new model, followed by creation of routes to actually perform the requisite crud operations of insert, update, delete and read.

Uncertainty Handling

For texts which do not come under the intent definition, are learnt into the system. These texts come under uncertainty. While queries should not be learnt into the system, they could be understood as specific tasks which enable it to perform crud operations.

For retrieval tasks, it is necessary to store the list of columns which can be done by means of a logic unit for common interaction between the chatbot and the backend. When a user enters a query, the command should achieve a low confidence in the pre-trained tags and hence a routine is triggered where the text can be parsed, followed by part of speech tagging (POS tagging) and entity matching from the logic table, the respective data can be retrieved.

3.2 Backend Integration

Additionally for query parsing, regex can be used preceded by lemmatization and stemming and pattern matching in addendum to it.

Custom Pipelining

This is essential to convert data from the chatbot to a series of logical outcomes that can quantitatively be justified (see Fig. 3).

This included gathering the intent of action meant by the user statement toward generating logical statements toward manipulating the end result. For example, based

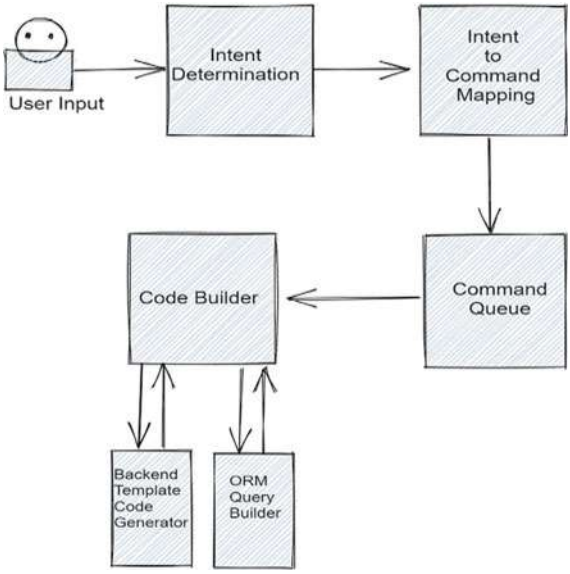


Fig. 3 Backend code generation pipeline

on whether the user needs to define a model or a route, there is a need to have the pipeline take a different set of actions to create a model, and a different set of actions to create a route.

Similarly in order to create a route, a requirement would be to get further details along for the route, for example, the type of logical outcome needed by the user, the type of inputs accepted, the type of data to be retrieved, the permission set for the route, etc. to name a few. In the current scope, the model is currently considering simple use cases of basic models being defined, and similarly for routes, simple routes to perform queries for listing database objects for some specific model based on certain parameters, and hence the logic conversion algorithm will be covered in this part.

The main features (in current scope of the researched idea) for this pipeline from a backend point of view are as follows:

1. It is a simple intent-based trigger
2. It is supposed to pick up data from user input and get relevant data portions, pre-processed and classified according to different labels to be fed to the generator functions further on (these labels are supposed to be the parameters that functions consume and produce an output).
3. The reason for choosing this approach is that with incremental utility these intents can be used to trigger other intents, hence this pipeline can be extended on lines of scalability and project design expansion without much complicacy and breaking design changes.

4. The intent functions will be responsible for generating 'tasks' that are atomic operations that the backend generator module will consume and utilize them to construct the code.
5. The reason the model is not directly performing the manipulation for code creation and having an independent unit to be interpreted for the conversion purposes is so that this architecture can be extended down the line with each of these components being reworked and extended on in a different fashion without total overhaul. So, for example, if tomorrow, the logic interpretation architecture changes/revamps, the backend generator can still continue working on this common 'task' consumption standard.
6. The tasks are not applied to the code generation directly, instead they are part of a task queue that is currently just a simple queue of tasks to serve the purpose, but with additional research, could be used for a lot of other smart stuff like interpreting the flow of tasks, and featuring out the tasks which are not correct, like if the user tries creating a route for a model that does not exist, or tries to query objects from a model with a non-existent field or performs an operation that is not data-type correct, then all these can be deduced at task aggregation time and alternate workflows/solutions be provided to the user for additional usability and error avoidance.

Logic to Backend Processing

This highlights the backend code generation part after the logic interpretation and task generation phase, and the primary goal of this part is to consume the tasks generated in the previous phase. It starts with a basic work breakdown structure for different tasks needed to be accomplished.

Different steps in incremental order:

1. Start the project creation.
2. Initialization of the apps for making APIs.
3. Configuring settings to include all the changes made in the application at each step as well as defining the different crucial parameters such as environment variables.
4. Working on creating comprehensive and basic database models (Additional work needed for determining type of database needed.).
5. Generating endpoints for users using view sets and generics.
6. Writing custom pipelines for deploying this service to PaaS (Like Heroku).

In order to achieve this, it needs to be understood how the backend generation would consume the tasks. In order to generate code for the backend, the model being followed is the templating model. So in these templates of code that accept parameters, and plugin user inputs based on what type of input is suited to which type of route, along with switch cases to determine the edge cases that may arise due to user inputs being inconsistent are being used. For example, if a certain user wishes to filter students by age in the student table, but the column age does not exist in the table, he might end up making an incorrect query and consequently, an incorrect action.

Similarly, his specified user input and the database schema might not be compatible, hence it might also be a logical fallacy in his route when it is constructed, etc. So the queries need to be type safe and properly validated at runtime before the code is constructed, in order to ensure an error free, working endpoint. For deployment, there could be a small general deployment script integrated with version control systems like Github, to trigger Github actions and workflows to deploy it to a PaaS like Heroku or even deploy it to a Virtual Machine.

3.3 Working

Now a comprehensive discussion on this topic might take days, or even weeks, of how someone could interpret human logic to machine code. Figure 4 highlights the entire workflow and algorithm.

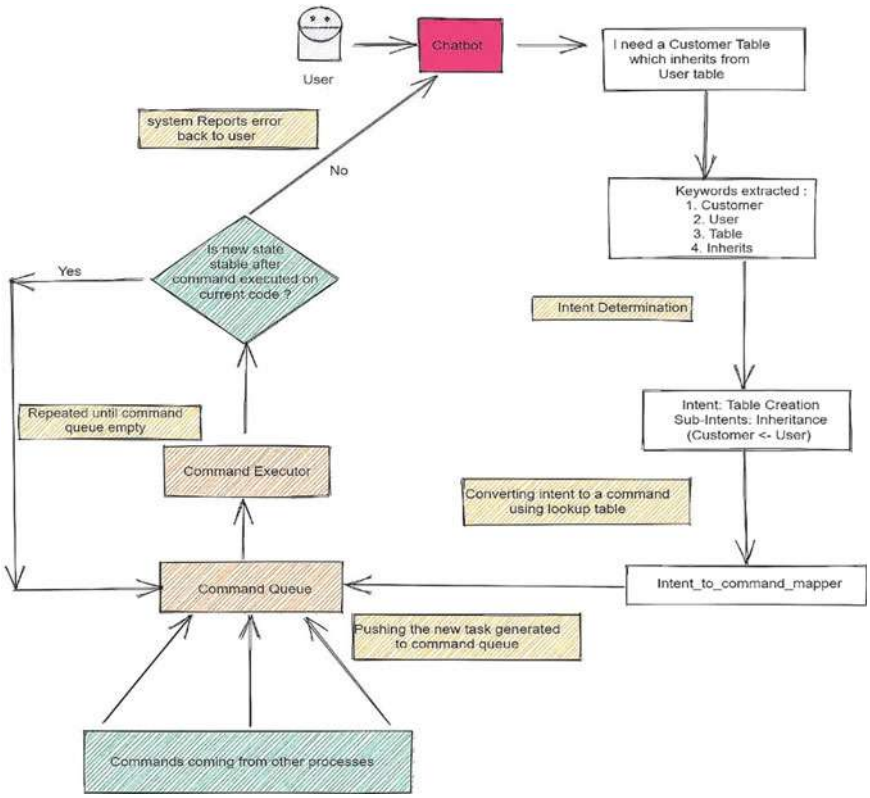


Fig. 4 An algorithm approach for the workflow

Table 1 Pipeline

Utility	Structure of development	
	Modules	Working
1	Model script	Abstract body
2	Crud operation	To create update delete and retrieve based on user requirement
3	Intents	txt file to tailor make the chatbot according to developers convenience
4	Starting project	To generate the project for the user
5	Logic processor	Converts data from the trained chatbot to a sequence/series of logical outcomes that can be quantitatively be justified
6	Query generator	To take the inputs and generate queries

An AI algorithm could be used, to translate and integrate some basic immutable logical ideas, to accomplish the task. This logic might not be perfect, but it could definitely be fine-tuned to respond correctly to tasks that are composed of these smaller discrete tasks that the machine can 'utilize'. Another 'easier' and less efficient way to accomplish this, is via some generic templates, that can plugin in a large number of variables, and generates a random logic, catering to some specific needs.

A model trained on some predefined intents, enabled with machine translation to convert local and regional text to a standard or universal language like English. Based on user inputs and confidence of the intents, the model decides the further steps. If the confidence is lower than a threshold, generally <70–80%, then it can classify it as an uncertainty. This can be used to trigger non creation tasks like different manipulation operations.

Brief descriptions of the modules that have been defined have been elucidated below in Table 1.

There was an effort to interpret this human logic into some simple mappings of SQL queries that would be consumed by the Django ORM to give back correct output to users. Some generic functions were taken into account, like parsing files (excel, csv) to accommodate some predefined data into the database.

For such a use case there might be a need to utilize another use case.

4 Conclusions

In this article, an approach was developed in which one has to observe and track down a reasonable method of making a logic that stimulates human discourse to the actual code developed by software engineers and designers. It can be utilized to tailor one's own backend logic for any application with an equivalent one. A few utilities of this setup include saving development time, automating the customizable part, virtual environment management, customizing routes tailored according to user's needs along with automatic handling of different middleware, serializes, models, permissions, etc.

5 Future Scope

The approach in this paper can be migrated toward a multimodal input. The backend creation in its entirety can just be a conversational experience. A subsequent iteration may be to implement a UI and customize it with a conversational/chatbot experience.

References

1. Panda M, Tripathy BK (2017) Internet of things and artificial intelligence: a new road to the future digital world. In: Tripathy BK, Anuradha J (eds) Internet of Things (IoT): technologies, applications, challenges and solutions, Chapter-3, 1st edn. CRC Press, pp 41–58
2. Tripathy BK, Anuradha J (2017) Internet of Things (IoT): technologies, applications, challenges and solutions, 1st edn. CRC Press
3. Panda GK, Tripathy BK, Padhi MK (2017) Evolution of social IoT world: security issues and research challenges. In: Tripathy BK, Anuradha J (eds) Internet of Things (IoT): technologies, applications, challenges and solutions, Chapter-5, 1st edn. CRC Press, pp 77–98
4. Sooraj TR, Mohanty RK, Tripathy BK (2017) Naming services in Internet of Things. In: Tripathy BK, Anuradha J (eds) Internet of Things (IoT): technologies, applications, challenges and solutions, Chapter-9, 1st edn. CRC Press, pp 167–188
5. Tripathy HK, Tripathy BK (2017) Applications of IoT to address the solutions for children affected by autism spectrum disorders (ASDs). In: Tripathy BK, Anuradha J (eds) Internet of Things (IoT): technologies, applications, challenges and solutions, Chapter-14, 1st edn. CRC Press, pp 295–326
6. Katiyar A, Srividya V, Tripathy BK (2021) TagIT: a system for image auto-tagging and clustering. In: Bhateja V, Satapathy SC, Travieso-González CM, Aradhya VNM (eds) Data engineering and intelligent computing, advances in intelligent systems and computing, vol 1. Springer, Singapore, pp 259–268. https://doi.org/10.1007/978-981-16-0171-2_25
7. Srividya V, Tripathy BK, Akhtar N, Katiyar A (2020) AgentG: a user friendly and engaging bot to chat for e-commerce lovers. Comput Rev J 7:7–19
8. Srividya V, Tripathy BK, Akhtar N, Katiyar A (2021) AgentG: an engaging bot to chat for e-commerce lovers. In: Tripathy A, Sarkar M, Sahoo J, Li KC, Chinara S (eds) Advances in distributed computing and machine learning. Lecture notes in networks and systems, vol 127. Springer, Singapore, pp 271–282. https://doi.org/10.1007/978-981-15-4218-3_27
9. Tripathy BK, Anuradha J (2015) Soft computing-advances and applications, 1st edn. Cengage Learning publishers, New Delhi
10. Gropengießer F, Sattler KU (2014) Database backend as a service: automatic generation, deployment, and management of database backends for mobile applications. Datenbank-Spektrum 14(2):85–95
11. Sulakhe D, Rodriguez A, D'Souza M, Wilde M, Nefedova V, Foster I, Maltsev N (2005) Gnare: automated system for high-throughput genome analysis with grid computational backend. J Clin Monit Comput 19(4):361–369
12. Liu R, Dong Z (2019) A study of user experience in knowledge-based QA chatbot design. In: International conference on intelligent human systems integration. Springer, Cham, pp 589–593
13. Chugh K, Solis AY, LaToza TD (2019) Editable AI: mixed human-AI authoring of code patterns. In: IEEE symposium on visual languages and human-centric computing (VL/HCC), Memphis, TN, USA, pp 35–43. https://doi.org/10.1109/VLHCC.2019.8818871
14. Hwang S, Kim B, Lee K (2019) A data-driven design framework for customer service chatbot. In: International conference on human-computer interaction. Springer, Cham, pp 222–236
15. Shinde R, Kulkarni R, Patwardhan M, Sarda S, Mantri P (2015) Conceptual schema extraction using POS annotations and weighted edit distance algorithm. In: International conference on

information processing (ICIP). IEEE, Pune, pp 719–724. https://doi.org/10.1109/INFOP.2015.7489476
16. Uma M, Sneha V, Sneha G, Bhuvana J, Bharathi B (2019) Formation of SQL from natural language query using NLP. In: International conference on computational intelligence in data science (ICCIDS). IEEE, Chennai, India, pp. 1–5. https://doi.org/10.1109/ICCIDS.2019.8862080
17. Husár A, Trmac M, Hranac J, Hruska T, Masarik K (2011) Automatic C compiler generation from architecture description language ISAC. In: Sixth doctoral workshop on mathematical and engineering methods in computer science (MEMICS'10)—selected papers. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
18. Singhal N, Jain H (2017) Automock automated mock backend generation for JavaScript based applications. Glob J Comput Sci Technol
19. Zamanirad S, Benatallah B, Rodriguez C, Yaghoubzadehfard M, Bouguelia S, Brabra H (2020) State machine based human-bot conversation model and services. In: International conference on advanced information systems engineering. Springer, Cham, pp 199–214

Chapter 43
All-Digital ADC Using Time-to-Digital Converter

Darshan R. Shaha, Mahadev S. Patil, and Sachin Magdum

1 Introduction

Current versions of FPGAs have become extremely puissant logic devices. Interfacing different sensors and further data processing have become valuable collection nodes for the physical sciences. Applications designed are verbally expressed to be feasible and reliable when they are

- Compact form factor (i.e., small footprint)
- Ultra-low-power compared to discrete system having same functionality.
- Use of limited input–output pins.
- Less affected to external parameters.

When an application is designed by taking all the above parameters into account, it amasses data from external sensors and processes it in FPGAs. The sensors used in many applications are the analog sensors, i.e., processing of continuous signal; due to advancement in digital field, analog approaches are losing their advantages to process and analyze output, so it can be further transformed using ADC into a digital signal. In many applications, ADC consequently plays a critical role, since it is the first phase of any design that uses analog-based circuitry/sensors.

The implementation of time-based ADC has increased in many of the emerging applications over traditional ADC. Currently, the key method used to implement an

D. R. Shaha (✉) · M. S. Patil · S. Magdum
Department of Electronics and Telecommunication, Rajarambabu Institute of Technology, Islampur 415414, India
e-mail: dshah6492@gmail.com

M. S. Patil
e-mail: mahadev.patil@ritindia.edu

S. Magdum
e-mail: sachin.magdum@ritindia.edu

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_44

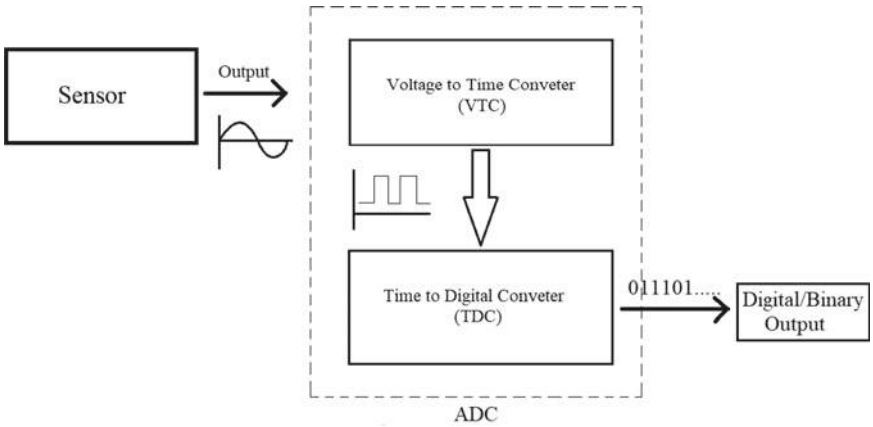


Fig. 1 Basic block diagram

ADC in an FPGA is the technique of delta-sigma modulation, from which a high-precision data range of 10–15 bits can be accessed [1]. But since the feedback loop mechanism is used for the delta-sigma modulation technique, it is intrinsically more gradual than without the feedback loop process. Since the sampling rate of FPGA TDCs is hundreds of MHz, it can be obtained using TDCs if we opt to transform the analog signal to a time domain vector. Time-based ADCs have high resolution, and conversion time is proportional to the input signals value. The simplest type of a TDC is an input and clock-enabled counter [2]. There are various topologies of TDC where tens of picoseconds and femtoseconds of resolution can be obtained [3, 4].

In this paper, full digital analog to digital converter (FD-ADC) based on pulse shrinking time-to-digital converter is proposed. This design is of mainly two stages: voltage-to-time converter (VTC) and time-to-digital converter (TDC) as shown in Fig. 1. The rest of the paper is organized as follows: Sect. 2, the architecture of proposed VTC design, Sect. 3, proposed TDC design, Sect. 4, the simulations and results. Determinately, conclusion presented in Sect. 5.

2 Proposed VTC

The proposed VTC has three rudimentary circuit; current mirror circuit, basic current starved circuit, and inverter. The proposed VTC is notion of current starved inverter principle with several of the modifications to the fundamental current starved circuit. The proposed single delay element from the ring VTC is schematically shown in Fig. 2. The signal is sent to the M1 and M2 transistor, with M3 acting as current starving element. The basic current starved circuit is made up of M1, M2, and M3. The current mirror circuit is used to increase the voltage sensitivity of the circuit as well as its linearity [5]. The current is engendering proportional to the V_{ctrl} voltage. The

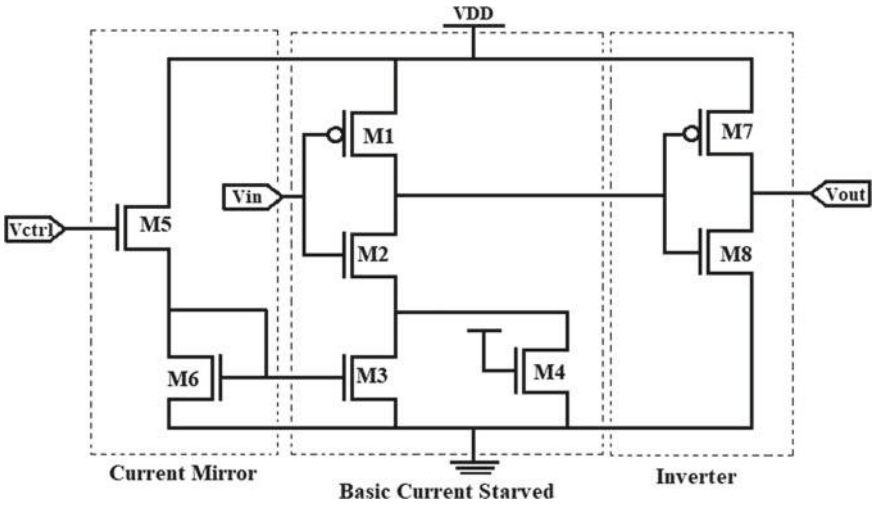


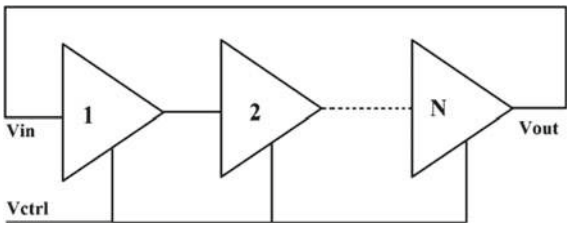
Fig. 2 Schematic of VTC with single delay element

current M6 and M3 are mirrored from bias stage to each inverter stage. Using V_{ctrl}, the oscillation frequency can be tuned for wide range by transmuting the value of control voltage. The width of M3 is much larger than that of M6 so that minute transmutation in V_{ctrl} will show drastic vicissitude in output current and which will increase the voltage sensitivity of VTC. The M7 and M8 act as inverter which increases the linearity of circuit. V_{ctrl} controls the delay of rising edge rather than falling edge of V_{out} as buffer inverter is present at end [5].

This Fig. 2 is a single delay element of our proposed VTC. We have connected these elements in ring structure so that each element will contribute its delay. By changing the supply voltage, the delay through each inverter will transmute. For higher voltages, the delays will be decrementing and which will increase the frequency of output signal.

Figure 3 shows the ring structure of proposed VTC, where each element that is reiterated is basic current starved circuit from Fig. 2. The frequency of each current starved circuit can be calculated as

Fig. 3 Ring structure VTC



$$f_{\text{osc}} = \frac{I_{\text{d}}}{2 * N * C_{\text{total}} * V_{\text{ctrl}}} \tag{1}$$

where

I_{d}	Drain current
N	No. of stages
C_{total}	Total capacitance
V_{ctrl}	Controlling voltage

3 Proposed TDC

Figure 4 is the block diagram of proposed pulse shrinking TDC. There are three main blocks such as coupling stage block, pulse shrinking block, and counter. The output of VTC is victualed to the coupling stage. The coupling stage consisted of two nor gates as shown in Fig. 5. The pulse is given in the commencement when reset command is inactive. The pulse was given to coupling stage rather than shrinking chain, because during diminutive vigor of pulse, there is possibility of data loss. Also another reason of coupling stage was that it omits any possible noise that input carries; hence, the pulse was regenerated. As NOR gates are utilized for coupling purport,

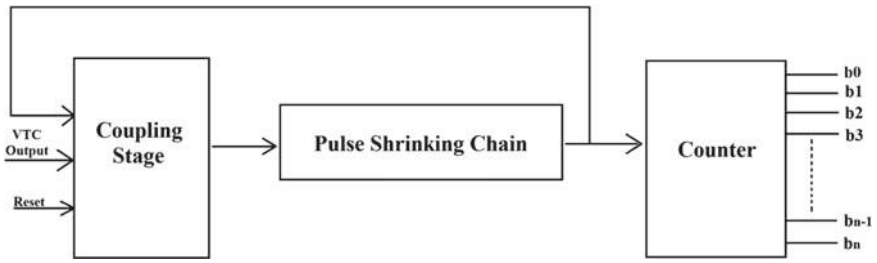


Fig.4 Block diagram of proposed TDC

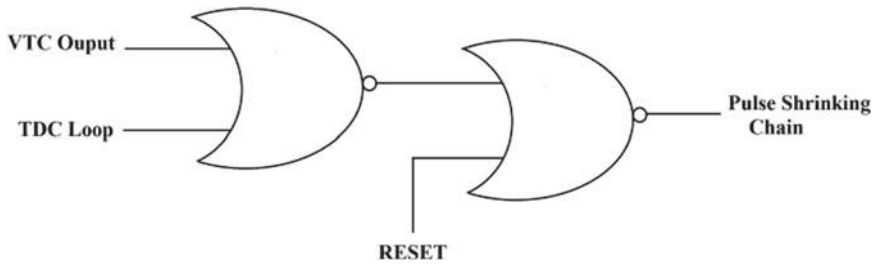


Fig. 5 Coupling stage

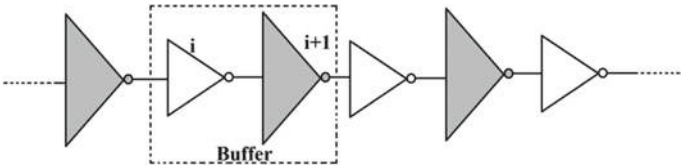


Fig. 6 Pulse shrinking chain

but the shrinking chain is built using inverters which leads to linearity quandary for our design.

Using pulse shrinking topology, high-resolution low area and power consumption are achieved [6]. The pulse shrinking chain is based on buffers delay lines design, and buffers are built using two inverters [7] in series as shown in Fig. 6. Here, each inverter "i" and "i + 1" shrink the pulse, which increases the resolution than that of traditional pulse shrinking loop and that abbreviate the conversion time.

$$\Delta C_i = C_{PLH,i} - C_{PHL,i} \tag{2}$$

$$\Delta C_{i+1} = C_{PHL,i+1} - C_{PHL,i+1} \tag{3}$$

Here using felicitous, W/L ratio for i and $i + 1$ plays crucial role otherwise the chain will not shrink the pulse. So to i be the shrinking element, it must $C_{PLH,i} > C_{PHL,i}$ and for $i + 1$ its precisely antithesis, it must be $C_{PHL,i+1} > C_{PHL,i+1}$ [7]. So, to gain high resolution, both the parameters should have close values. The number of pulses is counted till the input pulse gets thoroughly vanished, this is done using counter, and that output of counter is digital equipollent of input signal. The 5-bit counter is used.

Although looking at the TDC loop architecture, each clock pulse occurs approximately every 2.574 ns, leading to maximum operation frequency of 388.5 MHz. Therefore, the operation speed achieved by this counter is more than enough to fulfill the requirements.

The following Fig. 7 presents the proposed architecture for the ripple counter, based on D flip-flops.

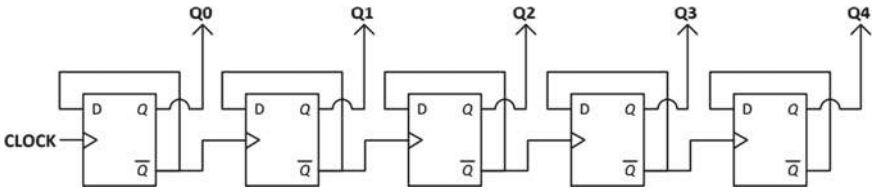


Fig. 7 5-bit ripple counter proposed architecture

4 Simulations and Result

4.1 VTC Sensitivity and Linearity

The variations in control voltage (V_{ctrl}) determine the frequency range and linearity of VTC for the different fixed input voltage [8] as shown in Table 1. The output waveform of VTC is shown in Fig. 8 with different input frequency.

It is visually examined that pulse width changes with the frequency of input signal. This linearization method used in VTC gives the linearity within the 1% of linear slope, i.e., about 900 mV, and the sensitivity is about 1.26 ps/mV. The pulse delay time is shown in Fig. 9.

Table 1 Output oscillating frequency at different control and supply voltage

V_{ctrl}	f_{osc} (GHz)		
	V_{in} (1 V)	V_{in} (2 V)	V_{in} (3 V)
0.2	0.1232	0.2431	0.4528
0.4	0.1734	0.3820	0.6189
0.6	0.8470	1.2583	1.5985
0.8	1.1991	2.1976	2.0416
1.0	1.2248	2.7533	2.8817

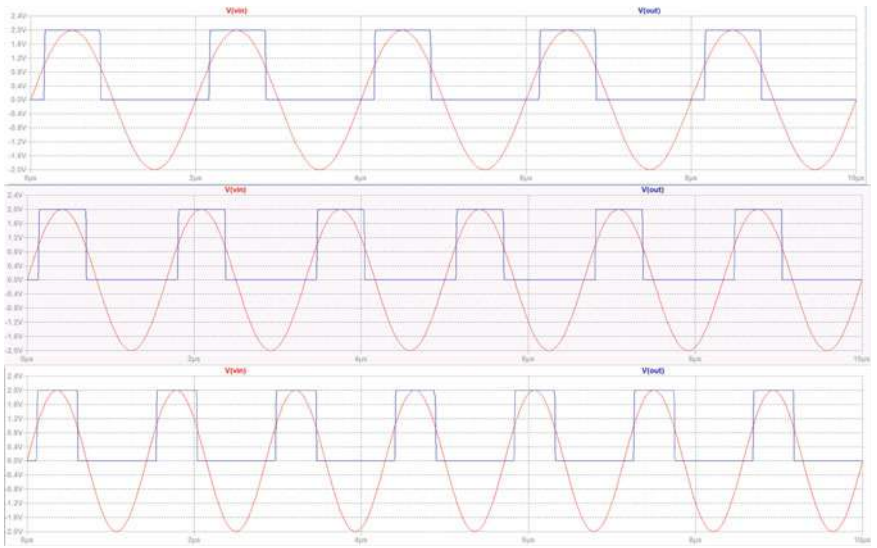


Fig. 8 VTC output waveforms at 500, 600, and 700 kHz

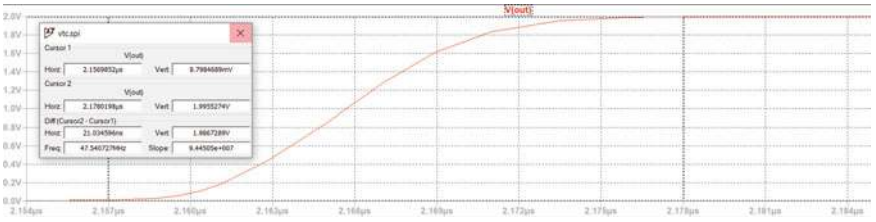


Fig. 9 Pulse delay time

4.2 DNL and INL Measurement

Differential nonlinearity (DNL) describes the deviation between the two output values corresponding to adjacent input quantities [9]. It is an important specification for measuring error of a converter, translating its accuracy. The results shown in each corner in Table 2 were normalized to one T_{LSB}. The following Table 2 resumes the maximum and minimum values achieve by the DNL, for each of the analyzed corner:

In analysis, DNL is the description of existent errors or missing codes in the converter dynamic performance, so we need to normalize it. Each time the DNL is different from zero means that sample presents an error in the output code. The first conclusion is that the proposed TDC could present multiple missing codes among all the dynamic range. Moreover, as mentioned values in Table 2, these missing codes fall up to −2.2 LSB.

Integral nonlinearity (INL) is a measure of the deviation between the ideal output value and the actual measured output value for a certain input code [9]. This measurement is performed after gain errors have been compensated. To obtain it, a best fit line of the obtained transfer function is used against the transfer function itself. The results shown in each corner in Table IV were normalized to one T_{LSB}. The following Table 3 resumes the maximum and minimum values achieve by the INL, for each of the analyzed corner:

Table 2 DNL results

T (°C)	Vcc (V)	DNL (LSB)	
		Minimum	Maximum
50	1.2	−0.97	0.99
−40	1.08	−2.2	1.89
	1.32	−1.51	1.55
120	1.08	−2.2	1.98
	1.32	−1.86	1.45
−40	1.08	−1.35	1.6
	1.32	−1.29	1.23
120	1.08	−0.96	0.93
	1.32	−1.45	1.2

Table 3 INL results

T (°C)	Vcc (V)	INL (LSB)	
		Minimum	Maximum
50	1.2	−1.53	1.25
−40	1.08	−2.31	2.22
	1.32	−1.65	1.61
120	1.08	−1.86	1.53
	1.32	−2.2	2.31
−40	1.08	−1.87	1.57
	1.32	−1.81	1.52
120	1.08	−1.56	0.96
	1.32	−1.78	1.44

In a macroscopic view, INL represents the description of the bending of a converter characteristic, describing the deviation of each step position from the ideal curve of the converter. As the name suggests "integral," the INL sums all the nonlinear events that lead to the deviation of the obtained fit line from the obtained results.

After the analysis of all the presented results for all of the tested corners, it can be concluded that in fact the TDC transfer function follows the intended linear curve, with some deviations in the slope due to the process corners. However, when we look to the INL and especially to the DNL results, it suggests that the converter presents some missing codes.

4.3 TDC

The shrinking of pulse depends upon the W/L ratio of MOSFETS in pulse shrinking inverter chain. So culling appropriate W/L is paramount so that pulse will shrink and not increase. So simulation results shown in Fig. 10 below are for "i" having the W_p/L_p ratio as 0.24/0.12 μm and W_n/L_n ratio as 0.7/0.12 μm and for "i + 1" with W_p/L_p ratio as 0.24/0.12 μm and W_n/L_n ratio as 0.74/0.12 μm. So using these designations, the maximum resolution achieved for proposed TDC is 0.76 psec, and whole time domain ADC resolution is 0.007 V.

The first thing that needs to be defined before building the counter will be the number of bits that it will possess, which determines the maximum number of loop cycles. As it was discussed before the TDC, delay line was dimensioned to hold a 2 ns pulse, however, due to restrictions, it will hold a maximum time of 2.574 ns, max Δt. Considering that in each delay stage, the resolution will be approximately 0.76 ps, and the maximum number of loops in the chain will be given by

$$N_{\text{count}} = \Delta t_{\max}/(M * T_{\text{LSB}}) = 2.574n/(128 \times 0.76p) = 26.45 \tag{4}$$

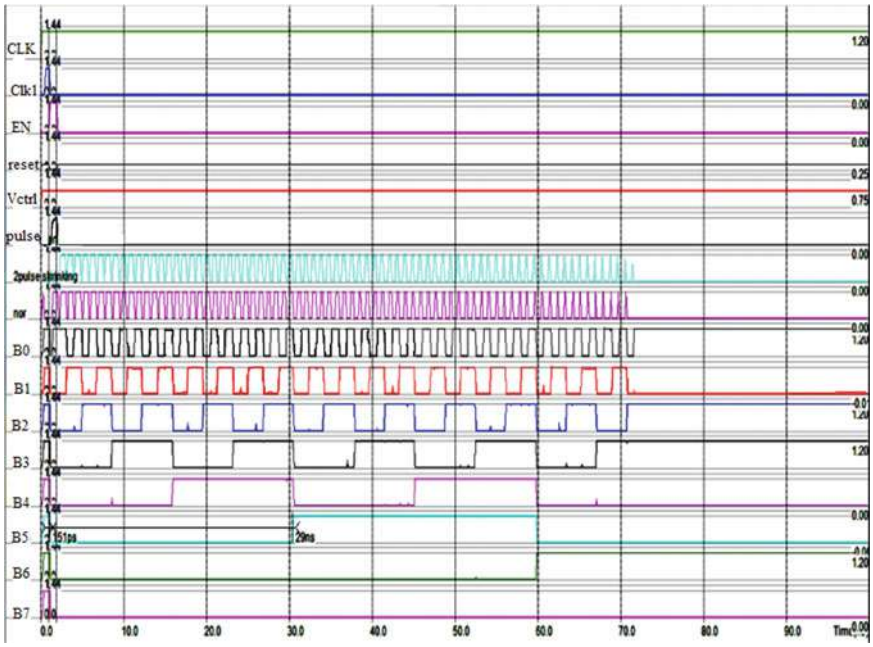


Fig. 10 TDC output waveforms

where max Δt is the maximum hold time in the line, M is the number of elements in the chain and T_{LSB} is the TDC resolution. Therefore, a 5-bit counter will be built for this purpose.

4.4 Power Consumption

To obtain the consumed power by the TDC proposed circuit, some measurements test was performed, from low to high measurement times in the dynamic range. The TDC circuit has higher power consumption; this is due to the initialization of the circuit, meaning VTC, delay line, and counter. Here, the power consumption has peaks at max 28 mW. As it can be seen, during the pulse measurement as we can see the power consumption is constant independently of the length of the measurement pulse. From there, it is possible to conclude that the power consumption during the measurement itself is around 17.8 mW from Fig. 11.

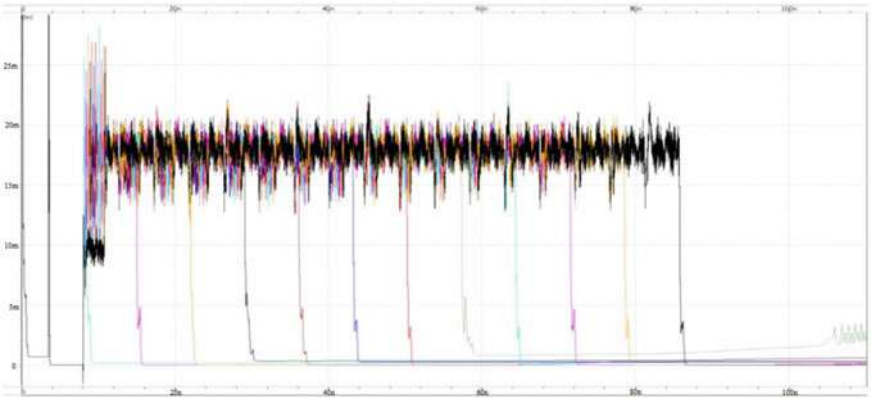


Fig. 11 Power consumption

4.5 Effective Number of Bits

The proposed TDC architecture uses 5 bits to deliver the output binary word. This means that the converter should be able to deliver $2^5 = 32$ different time codes, that with a time resolution of 0.76 ps results in a maximum measurement time pulse of 3.35 ns. However, above section shows that the maximum time that the delay line can measure to prevent it from overflow is 2.574 ns, in typical conditions. Hence and considering the corners, variations in all the measurements performed the maximum number of samples tested were around 300. Therefore, the ENOB of the proposed TDC is going to be

$$\text{ENOB} = \log_2(40) = 5.35\,\text{bits} \tag{5}$$

5 Conclusions

This proposed work presents a time-based ADC design consisting of ring structured VTC based on current starving principle and pulse shrinking TDC blocks. Output signal pulse width of proposed VTC circuit output is proportional to the input signal frequency with high linearity (about 1% linear slope), and voltage sensitivity (1.26 ps/mV) is proposed with 5-bit asynchronous counter as final output of TDC. As a result, a 128-inverter delay-chain was able to measure time pulses up to 2.574 ns. Being a loop structure, a 5-bit ripple counter was added to the system in order to count the number of completed loops that the time measurement pulse took for input signal. The DNL analysis shows that the corner has a maximum variation between −0.97 and 0.99 LSB, and in the worst corner, corner with a Vcc = 1.08 V

and T = 120 °C a variation between −2.2 and 1.98 LSB. The INL analysis presents the corner with a variation of −1.53 to 1.25 LSB, and the worst process corner, corner with a Vcc = 1.32 V, and T = 120 °C, a variation between −2.2 and 2.31 LSB. A compact and modular design based on a squared structure was achieved. However, the main objective for this kind of structure was to assure that all elements were subjected to the same conditions, and the global circuit has a symmetrical design. This will prevent nonlinearity due to parasitic elements in the circuit. The obtained structure occupies 0.587 mm^2. The post-layout simulation (extracted circuit simulated) performed against the results of the typical case with a 1.2 V power supply, and a temperature of 50 °C revealed a maximum achievable resolution of 0.76 ps, a DNL variation of −0.97 to 0.99 LSB, and an INL variation of −2.31 to 2.22 LSB. For the power consumption, the circuit revealed a stable consumption of 17.8 mW.

References

1. Gupta AK, Nagaraj K, Viswanathan TR (2011) A two-stage ADC architecture with VCO-based second stage. IEEE Trans Circuits Syst 58
2. Toraskar DD, Mattada MP, Guhilot H (2016) Time domain ADC using pulse shrinking TDC. In: I4C. IEEE
3. Khalirbaginov R (2020) Verilog design of all-digital phase-locked loop with two-cycle conversion time-to-digital converter. IEEE, EIConRus
4. Ugur C, Korcyl G, Michel J, Penschuk M, Traxler M (2013) 264 Channel TDC platform applying 65 Channel high precision (7.2psRMS) FPGA based TDC. In: IEEE Nordic-Mediterrean workshop on time-to-digital converters (NoMe TDC), pp 1–5
5. Mostafa H, Ismail YI (2013) Highly-linear voltage-to-time converter (VTC) circuit for time based analog-to-digital converters (T-ADCs). In: IEEE 20th ICECS
6. Harpe P, Baschirotto A, Makinwa KAA (2014) High-performance AD and DA converters, IC design in scaled technologies, and time-domain signal processing. In: Advances in analog circuit design 2014 book published by Springer
7. Granja R, Santos M, Guilherme J, Horta N (2018) 11.7b time-to-digital converter with 0.82 ps resolution in 130 nm CMOS technology. In: 14th conference on Ph.D. research in microelectronics and electronics
8. Samimian N, Mousazadeh M, Khoie A (2019) A time-based all-digital analog to digital converter for IOT applications. In: 27th Iranian conference on electrical engineering (ICEE)
9. Henzler S (2010) Time-to-digital converters. Springer Science and Business Media LLC

Chapter 44
Breast Cancer Subtypes Prediction Using Omics Data and Machine Learning Models

Shiekhah AL Binali, Souham Meshoul, and Hadil Shaiba

1 Introduction

Breast cancer (BC) is among the most serious diseases that women face today. The number of cases is continuously increasing. In 2018, over two million new instances with BC have been reported [1]. The widespread of the disease along with the availability of a variety of data helped to learn more about this disease. Several studies have been conducted and resulted in identifying different types of BC based on clinical, pathological, and molecular characteristics. This gives a great opportunity to find a suitable treatment for each case which would result in significant advances in personalized medicine or precision medicine. Availability of omics data has enabled investigation of the molecular typing of BC. In fact, breast cancer is defined as a heterogeneous disease that demonstrates diversity in the molecules [2]. Several types have been identified depending on the biological markers used [1] such as immunohistochemistry (IHC) markers including those for estrogen receptor (ER), progesterone receptor (PR), and human epidermal growth factor 2 (HER2).

Breast cancer subtyping is useful to recommend the right treatment to the right patient which would result in a significant gain in terms of efficiency and cost. During recent years, genetic datasets of cancer patients have become available publicly resulting in a proliferation of research on developing computational solutions for the purpose of diagnosis and prognosis. BC subtyping is basically a multi-classification

S. AL Binali (✉) · S. Meshoul · H. Shaiba
College of Computer Science and Information Technology, Princess Nourah Bint Abdulrahman University, Riyadh, Kingdom of Saudi Arabia
e-mail: shiekhah.albinali@gmail.com

S. Meshoul
e-mail: Sbmeshoul@pnu.edu.sa

H. Shaiba
e-mail: HAShaiba@pnu.edu.sa

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_45

task that has been tackled by several previous studies. In this work, we are particularly interested in the omics data available at the Cancer Genome Atlas (TCGA), namely the mRNA, DNA methylation, and copy number variation (CNV). These datasets represent various omics data about the same patients and provide information about the type of cancer. Therefore, they can be very useful to generate subtype prediction using machine learning algorithms. These datasets are characterized by a huge number of descriptive features compared to the very small number of instances. Furthermore, they are highly imbalanced. This makes the training of machine learning models a very challenging task. Some recent works proposed the use of deep learning models to deal with this problem [1, 3–7]. Good performance has been achieved in binary classification. However, multi-class classification still needs further improvements. The idea behind this work is that using deep learning for these datasets may not be an appropriate choice because of the small size in terms of the number of instances. Therefore, in this paper, we describe approaches for solving these issues with the intention of determining the top approach in respect of several performance metrics and also, the omics data that is more relevant in BC subtyping.

The remainder of the paper is carried out as follows. An overview of related studies is covered in Sect. 2. The used datasets, the proposed framework and investigated approaches are described in Sect. 3. Section 4 is devoted to the experimental study and obtained results. Finally, a conclusion and plans for future work are provided.

2 Related Work

Various models have been proposed for BC subtyping. They can be classified depending on the formulation they adopted for BC subtyping, namely as a binary or multi-class classification problem, on the omics data used, on the way these omics data have been used and on the machine learning model used to perform prediction. Most of the recent works proposed deep learning models. The model described in [8] was created to distinguish between the subtypes using high dimensional genomic data. The model requires large dataset for training and cannot perform well with small sets [8]. On the other side, authors in [9] used various clustering methods, namely direct integrative clustering, clustering of the clusters, and integrative regulatory cluster [9]. The challenge was to characterize genomic, epigenomic, and transcriptomic features of large-scale samples simultaneously in a short time and less cost [9].

The study in [3], proposes a deep hierarchal learning method that includes a stacked autoencoder neural network to learn high-level representations for each kind of data and combines gene expression and transcriptome alternative splicing profiles to specify subtypes. The next model would use genomic data to combine FirebrowR, data normalization methods, and EDL to redefine known breast cancer subtypes and highlight substantial ERB2, ESR1, and PgR alterations and CCNB1 [10]. The new classification method has been found reliable, achieves good results,

and automatically provides a mechanism for feature selection [10]. This model's limitations are that the model works on a small dataset [10]. Another model described in [4] employed an integrated deep learning architecture to classify breast cancer molecular subtypes. The model results were the best when integrating the data source by the model's concatenation layer without the weights [4].

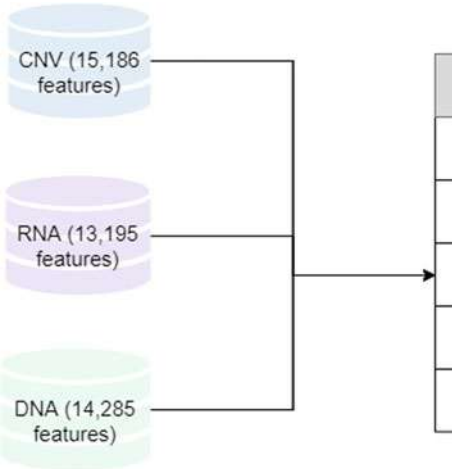
Another interesting approach employs a hierarchical integrated deep flexible neural forest framework to identify breast cancer subtypes using integrated multi-omics data rather than only gene expression data similar to most methods [7]. HI-DFNForest integrated gene expression, miRNA expression, and DNA methylation with stacked autoencoder to classify subtypes [7]. To learn the forms of each omics dataset and its sophisticated representations based on learnt features, a stacked autoencoder has been employed [7]. The complex representation is used as an input to the DFNForest to classify subtypes [7]. HI-DFNForest resulted in accuracy improvement compared to utilizing gene expression data, and this model has outperformed other conventional methods [7]. This model is used to classify cancer tissues, and molecular subtypes using RNA expression data and artificial intelligence since knowing the patient's underlying malignancy remains crucial to their therapy [5]. CUP-AI-Dx model applies 1D inception convolutional neural network to classify a tumor's primary tissue [5]. CUP-AI-Dx has been able to identify the primary site with a cross-validation accuracy of 98.54% and a test dataset accuracy of 96.70% [5]. Deepathology is a deep multi-task learning model that classifies molecular pathology, using deep neural networks to various biological samples' properties [6]. Deepathology has 99.4% accuracy rate in detecting the right subtypes of cancer and resistant to noisy and missing data, and has outstanding accuracy in the baseline algorithm [6].

One of the most recent models is DeepMo, which uses deep neural networks to multi-omics data and classify breast cancer subtypes [1]. Despite the results being better than those of other methods for the multi-classification task in terms of accuracy and area under the curve (AUC) [1], there is still a need for further improvements [1].

Thus, this research aims to enhance the predictive performance of BC subtype by investigating the impact of data type, various methods for feature selection, and various machine learning algorithms.

3 Materials and Models

This section explains the datasets being used in this research work along with the models that have been designed for the purpose of BC subtyping.



Sample type	Number of sample
Luminal A	277
Luminal B	40
TNBC	70
HER2(+)	11
Unclear	208

Fig. 1 Description of the used datasets, (right): number of features per dataset (left): distribution of BC subtypes in each dataset

3.1 Data Collection and Understanding

The original data is a public dataset from the Cancer Genome Atlas (TCGA) [11]. It is a widely utilized database for cancer research since it contains a wide variety of cancer forms, as well as numerous omics and clinical data like mRNA, DNA methylation, and copy number variation (CNV) [11]. In TCGA, there are more than 200 types of cancer that have been described and characterized by different molecular profiles [11]. The data in this study is collected from [1]. CNV data has 15,186 features, mRNA data contains 13,195 features, while DNA methylation data has 14,285 features. The total number of samples is 606 [1]. Figure 1 shows the total number of features in each dataset and the BC type distribution in each dataset.

3.2 Modeling with Single Omics

To deal with the aforementioned issues, the idea was to investigate several alternatives and combinations of:

- Dimensionality reduction using feature selection methods such as Chi-square, ensemble learning and recursive feature elimination (RFE).
- Data imbalance handling using simple oversampling methods because of the sensitivity of biological data.
- Classification models: In our study, we focused on machine learning models such as random forest (RF), K-nearest neighbors (KNN), and support vector classification (SVC).

- Dimensionality reduction

As the name suggests, dimensionality reduction refers to the process of reducing the number of features in a highly dimensional dataset through either feature extraction or feature selection. Due to the sensitivity of the data, feature selection is more suitable since the values would not change and feature selection would give the best features to distinguish the subtypes. In order to have an in depth-understanding, we selected a method from the feature selection types. Chi-square filtering, recursive feature elimination (RFE) which is a wrapper method, and random forest ensemble learning. Chi-square is used to determine if a subset of data would match a population, and this method is used to observe the correlations and associations among different categories [12]. RFE is known as a greedy optimization algorithm [12]. This method creates models of different subsets and calculates the best performing feature and adds their scores according to an elimination ranking [12]. Ensemble learning would use randomization to have an increase in diversity [13]. The base learner is a CART-based tree, and this algorithm will search through a significant possible binary split for each feature to find the optimal split for each node [13]. The criterion for the splits is the measure of information gain and the information gain measure results from a reduction in entropy from the split and would use bagging to generate each tree's training set [13].

- Handling data imbalance

After reducing the data dimensionality, the data imbalance issue needs to be solved. As mentioned before, there are two major classes and three minor classes. When the data is imbalanced, the model would not recognize the minority classes properly. Since the data is sensitive, the option of creating similar instances using methods like SMOTE is not recommended here. Thus, a simple oversampling is used where the minority classes instances are duplicated. For comparison purposes, we also investigated the same imbalance handling as DeepMo in [1] where upsampling minorities using weights is adopted to give the minority class instances higher weights than majority class instances.

- Classification models

Three ML models, namely RF, KNN, and SVC have been designed, built and applied to each of the three used datasets after feature selection and dimensionality reduction problems have been solved. Random forest is a tree-based ensemble that relies on a set of random features per tree [14]. The real-valued respond is represented by the random feature Y in the p-dimensional random vector that represents the real-value input variable [14]. Assuming a joint distribution that is unknown to construct a prediction function for predicting Y [14]. This prediction function is calculated using a loss function to minimize the expected loss value, measure how near f(X) is to Y [14], and penalize numbers that are distant from Y [14]. KNN is a similarity-based learning approach [15]. In the training phase, the model is built based on the nearest neighbor simply, including storing the training instance in memory [15]. In the prediction stage, the model will predict the new query instance, and the distance

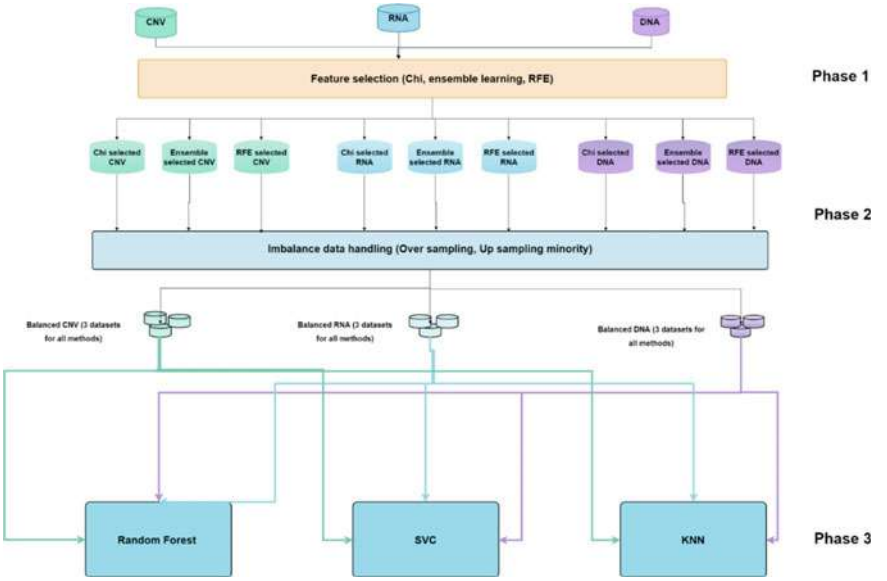


Fig. 2 The proposed framework

in the feature space among the query and instances in memory is computed [15]. The model will give a prediction according to the nearest to the query in the feature space [15]. C-support vector classification is a binary classification that belongs to the support vector machine family [16]. The model will learn based on recent advances in statistical, pattern recognition, and machine learning [16]. The decision function is used to predict the class label y for any input x based on which class has the majority voting or the decision function [16]. If there are more than two classes with the same number of votes, the input will be unclassified [16]. Figure 2 shows an outline of the modeling used with single omics data.

4 Experimental Study and Results

To evaluate the proposed models, hold-out sampling has been applied where the training set covers 70% of the data and the testing set the remaining 30%. Furthermore, performance measures such as accuracy, recall, precision, and F1-score were being used to assess the presented models' efficiency. First experiments have been conducted in order to set the models parameters. For RF, the number of trees is between 300 and 1500 trees depending on the dimensionality technique used. The next parameter is the criterion which specifies the type of measurement of the split quality, and in this study, Gini impurity is used.

Table 1 Number of the selected features in each used omics dataset and DR method

Dataset	Initial number of features	Dimensionality reduction methods		
		Chi-square	Ensemble learning	RFE
CNV	15,186	5000	3867	5000
RNA	13,195	5000	2638	5000
DNA	14,285	5000	2892	5000

For the SVC classifier, the kernel used is radial basis function. The next parameter is the regulation parameter and is set to five. Breaking the ties option would break the ties based on the confidence values of the decision function. For the final classifier KNN, the number of neighbors to use or k value is either three or five, depending on the dimensionality reduction method used. BallTree algorithm and Euclidean distance are used to calculate the nearest neighbors.

4.1 Feature Selection Results

As can be shown in Table 1, the features were reduced in a significant way. Chi-square filtering and RFE reduced the data into the best 5000 for each dataset, whereas ensemble learning using random forest achieves more significant reduction with less than 4000 selected features. The DeepMo model described in [10] used the Chi-square method, therefore the set of features has been reduced to 5000 features as well.

4.2 Handling Class Imbalance Results

To show the effectiveness of the used oversampling technique, results with and without balanced datasets have been recorded. We can clearly see in Table 2 which shows results of the application of RF on CNV dataset reduced with Chi-square, that significant improvements in terms of the used performance measures have been achieved. Same significant improvements have been obtained with the other models and the other datasets. This indicates the importance of handling imbalanced data.

4.3 Proposed Models' Results

After balancing the data, each dataset was divided into two groups: training and testing, where the testing set equals 30% of the dataset. The results of applying the classifiers on all balanced datasets for all dimensionality reduction methods, have

Table 2 Results of the random forest with and without balancing the data for CNV CHI-RF

Labels	Precision		Recall		F1-score	
	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced
Luminal A	0.5943	0.7403	0.8182	0.7403	0.6885	0.7403
Luminal B	0.4286	0.9524	0.5000	1.0000	0.4615	0.9756
TNBC	0.6000	0.8545	0.5714	1.0000	0.5854	0.9216
HER2	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000
Normal	0.5238	0.7581	0.3284	0.5732	0.4037	0.6528
Macro average	0.4293	0.8611	0.4436	0.8627	0.4278	0.8580
Accuracy	Imbalanced	0.5659		Balanced		0.8678

been recorded and summarized in Tables 3, 4, and 5, respectively, related to CNV, mRNA, and DNA methylation datasets. In these tables, the name of the method refers to the feature selection method used (CHI, ensemble, and RFE) and the classifier

Table 3 Results on CNV for multiple feature selection methods and classifiers

Method	Accuracy	Precision	Recall	F1-score
CHI-RF	0.8678	0.8611	0.8627	0.8580
CHI-SVC	0.8558	0.8466	0.8505	0.8480
CHI-KNN	0.7548	0.7361	0.7483	0.7411
Ensemble-RF	**0.8942**	**0.8980**	**0.8919**	**0.8885**
Ensemble-SVC	0.8438	0.8422	0.8469	0.8436
Ensemble-KNN	0.7620	0.7517	0.7671	0.7577
RFE-RF	0.8702	0.8814	0.8841	0.8711
RFE-SVC	0.8486	0.8485	0.8577	0.8506
RFE-KNN	0.7356	0.7247	0.7480	0.7300

Table 4 Results on mRNA dataset for multiple feature selection methods and classifiers

Method	Accuracy	Precision	Recall	F1-score
CHI-RF	0.9327	0.9337	0.9299	0.9299
CHI-SVC	0.9255	0.9310	0.9221	0.9222
CHI-KNN	0.8029	0.7969	0.7973	0.7924
Ensemble-RF	**0.9567**	**0.9571**	**0.9571**	**0.9554**
Ensemble-SVC	0.9399	0.9393	0.9400	0.9380
Ensemble-KNN	0.9038	0.9044	0.9040	0.9001
RFE-RF	0.9183	0.9193	0.9194	0.9183
RFE-SVC	0.8990	0.8988	0.9014	0.8996
RFE-KNN	0.8149	0.8136	0.8189	0.8157

Table 5 Results on DNA methylation dataset for multiple feature selection methods and classifiers

Method	Accuracy	Precision	Recall	F1-score
CHI-RF	0.9423	0.9442	0.9419	0.9415
CHI-SVC	**0.9567**	**0.9567**	**0.9567**	**0.9564**
CHI-KNN	0.7764	0.7738	0.7734	0.7683
Ensemble-RF	0.9423	0.9460	0.9388	0.9390
Ensemble-SVC	0.9447	0.9451	0.9414	0.9415
Ensemble-KNN	0.7572	0.7347	0.7473	0.7381
RFE-RF	0.9399	0.9426	0.9369	0.9360
RFE-SVC	0.9327	0.9325	0.9293	0.9281
RFE-KNN	0.8438	0.8313	0.8367	0.8293

(RF, KNN, and SVC). The tables present the accuracy, the macro average of the precision, recall, and F1-scores. The best results are highlighted in bold.

As can be seen in Tables 3, 4, and 5, the ensemble-RF model achieved the best performance in terms of all measures for CNV and mRNA datasets, whereas CHI-SVC outperformed the other models for the DNA methylation dataset but not significantly compared to ensemble-RF and ensemble-SVC. In all datasets, the KNN classifier performed poorly with all feature selection methods.

To show the appropriateness of the used ML models compared to deep learning models, the obtained results have been compared to those of the DeepMo method a state-of-the-art method [1]. Tables 6, 7, and 8 show the results in terms of accuracy for each used classifier compared to the DeepMo model. In all cases, the proposed models outperformed the DeepMo model. This can be justified by the fact that DeepMo

Table 6 RF versus DeepMo

Dataset	DeepMo	Dimensionality reduction technique		
		Chi-square	Ensemble learning	RFE
CNV	0.525	0.868	**0.894**	0.870
RNA	0.743	0.933	**0.957**	0.918
DNA	0.766	**0.942**	**0.942**	0.940

Table 7 SVC versus DeepMo

Dataset	DeepMo	Dimensionality reduction technique		
		Chi-square	Ensemble learning	RFE
CNV	0.525	**0.856**	0.844	0.849
RNA	0.743	0.926	**0.940**	0.899
DNA	0.766	**0.957**	0.945	0.933

Table 8 KNN versus DeepMo

Dataset	DeepMo	Dimensionality reduction technique		
		Chi-square	Ensemble learning	RFE
CNV	0.525	0.755	**0.762**	0.736
RNA	0.743	0.803	**0.901**	0.815
DNA	0.766	0.776	0.757	**0.844**

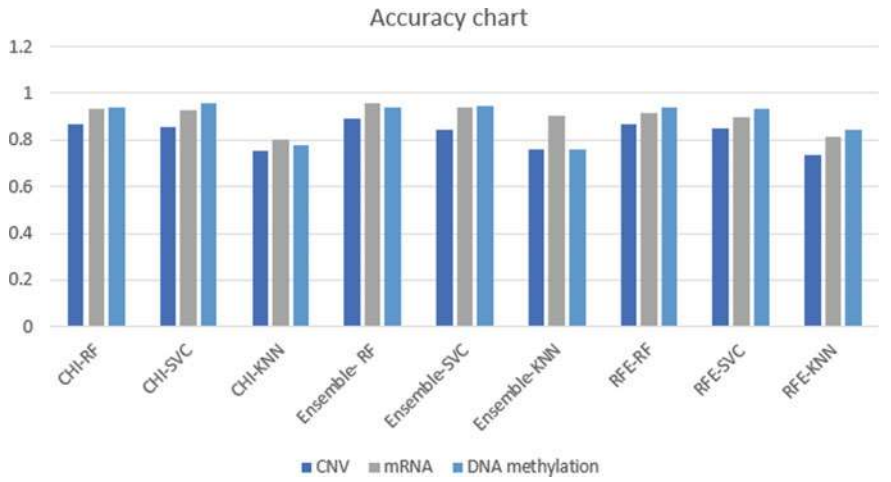


Fig. 3 Accuracy results for all models using single omics data

requires much more data for good training of the model. Therefore, the proposed approaches seem to be a more suitable choice.

Finally, to answer the question: which omics data is more relevant in identifying the BC subtype, the accuracy results have been organized in terms of a bar chart as shown in Fig. 3 where we can easily see that CNV gave the worst results, and competitive performance obtained by mRNA and DNA methylation datasets.

5 Conclusion

This study aims to provide an improved model that can identify breast cancer subtypes using single omics data with better performance compared to previous studies. Several models have been built using various combinations of dimensionality reduction techniques, data imbalance handling, and classification models. The research has been conducted by considering multi-class classification with single omics data. Dimensionality reduction has been achieved through the Chi-Square method, RFE, and ensemble learning random forest. Data imbalance has been handled using a

simple random oversampling since the data is sensitive and generating new data similar to the previous ones is unacceptable. The models used are traditional ML models, which are random forest, KNN, and SVC. The results for random forest and SVC were better than KNN in most cases. mRNA and DNA methylation have better results in identifying the subtypes compared to CNV. Furthermore, the proposed models outperformed DeepMo a recent state-of-the-art method. As plans for future work, we intend to investigate a multi-omics scenario and compare the results to decide which scenario provides the best results.

Acknowledgements Authors would like to thank Yuqi Lin from Jilin University for providing the datasets which enable us to explore multiple scenarios and conduct a comparative study.

References

1. Lin Y, Zhang W, Cao H, Li G, Du W (2020) Classifying breast cancer subtypes using deep neural networks based on multi-omics data. Genes. https://doi.org/10.3390/genes11080888
2. Dass SA, Tan KL, Selva Rajan R, Mokhtar NF, Mohd Adzmi ER, Wan Abdul Rahman WF, Tengku Din TADAA, Balakrishnan V (2021) Triple negative breast cancer: a review of present and future diagnostic modalities. Medicina 57. https://doi.org/10.3390/medicina57010062
3. Guo Y, Shang X, Li Z (2019) Identification of cancer subtypes by integrating multiple types of transcriptomics data with deep learning in breast cancer. Neurocomputing. https://doi.org/10.1016/j.neucom.2018.03.072
4. Mohaiminul Islam M, Huang S, Ajwad R, Chi C, Wang Y, Hu P (2020) An integrative deep learning framework for classifying molecular subtypes of breast cancer. Comput Struct Biotechnol J. https://doi.org/10.1016/j.csbj.2020.08.005
5. Zhao Y, Pan Z, Namburi S, Pattison A, Posner A, Balachander S, Paisie CA, Reddi HV, Rueter J, Gill AJ, Fox S, Raghav KPS, Flynn WF, Tothill RW, Li S, Karuturi RKM, George J (2020) CUP-AI-Dx: a tool for inferring cancer tissue of origin and molecular subtype using RNA gene-expression data and artificial intelligence. EBioMedicine. https://doi.org/10.1016/j.ebiom.2020.103030
6. Azarkhalili B, Saberi A, Chitsaz H, Sharifi-Zarchi A (2019) DeePathology: deep multi-task learning for inferring molecular pathology from cancer transcriptome. Sci Rep. https://doi.org/10.1038/s41598-019-52937-5
7. Xu J, Wu P, Chen Y, Meng Q, Dawood H, Dawood H (2019) A hierarchical integration deep flexible neural forest framework for cancer subtype classification by integrating multi-omics data. BMC Bioinform. https://doi.org/10.1186/s12859-019-3116-7
8. Chen L, Xu J, Li SC (2019) DeepMF: deciphering the latent patterns in omics profiles with a deep learning method. BMC Bioinform. https://doi.org/10.1186/s12859-019-3291-6
9. Wang D, Gu J (2016) Integrative clustering methods of multi-omics data for molecule-based cancer classifications. https://doi.org/10.1007/s40484-016-0063-4
10. Deng M (2018) Predicting rules for cancer subtype classification using grammar-based genetic programming on various genomic data types
11. Tomczak K, Czerwińska P, Wiznerowicz M (2015) Review the cancer genome atlas (TCGA): an immeasurable source of knowledge. Contemporary Oncology/Współczesna Onkologia 68–77. https://doi.org/10.5114/wo.2014.47136
12. Aggarwal CC (2014) Data classification: algorithms and applications. CRC Press
13. Winkler J, Lawrence N, Niranjan M (2005) Deterministic and statistical methods in machine learning: first international workshop, Sheffield, UK, 7–10 Sept 2004. Revised Lectures. Springer, Berlin

14. Zhang C, Ma Y (2012) Ensemble machine learning: methods and applications. Springer, New York
15. Kelleher JD, Mac Namee B, D'Arcy A (2015) Fundamentals of machine learning for predictive data analytics
16. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. Taylor & Francis

Chapter 45
Abusive Bangla Comment Detection from Social Media Using Machine Learning Approach

Pratim Saha, Naznin Sultana, Ashraful Haque Khan, and Shibli Noman

1 Introduction

Social media paves an important part in the field of communications. People cannot only share pictures or videos on social media, but also they can chat or comment at the same time. They can even do voice call or video conference using different social media. Among different social media's Facebook has become the most popular medium of communication around the world. People from all walks of life irrespective of all ages mostly use Facebook to communicate virtually with friends, families, and others. It opens an environment for all kinds of people to express their feelings with others through online. This is at the same time good and bad as well since, some people often share offensive pictures and comments which may hamper others socially or mentally. This type of incidence is also increasing day by day in Bangladesh.

Bangla language is spoken by more than 228 million native speakers and another 37 million as second speakers. About 98% of Bangladeshi people use Bangla as their first language.[1] In Bangladesh, total Internet users are 66.44 million, among

[1] https://en.wikipedia.org/wiki/Bengali_language.

P. Saha (✉)
University of Alabama at Birmingham, Birmingham, AL, USA
e-mail: shahapratim@gmail.com

N. Sultana · A. H. Khan · S. Noman
Daffodil International University, Dhaka, Bangladesh
e-mail: naznin.cse@diu.edu.bd

A. H. Khan
e-mail: ashraful15-899@diu.edu.bd

S. Noman
e-mail: shibli15-916@diu.edu.bd

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_46

them 36.0 million are active social media users.[2] Every day, this type of users is increasing, and cyber bullying is also increasing.

A number of researchers have been carried out in detecting abusive comments. In our work, we have implemented a number of classic and ensemble machine learning (ML) algorithms to detect such abusive comments from online social media datasets and compare their performance by some evaluation matrices. Our experimental study suggests that support vector machine (SVM) and SGD with modified Huber outperform than other existing machine learning algorithms in detecting abusive reviews from Bangla texts. SVM is a linear machine learning model that determines the classes by creating a line between the data points. SVM tries to find closest point of both the classes from the line. This points are regarded as support vector and hence the name of the model. SVM works by drawing a boundary between the classes. In cases, when data cannot be separated linearly, SVM creates an extra dimension to it and then set the decision boundary.[3] SGD is just an optimization technique, and it is a way of training a model. It works by implementing a plain stochastic gradient descent learning routine. It allows different loss functions and penalties for classification. In our case, we used modified Huber loss function and decision boundary was drawn in a similar pattern to that of linear SVM.[4] The rest of the paper consists of the following parts:

Section 2 contains the related work. Section 3 contains discussion on our proposed methodology, Sect. 4 highlights the experimental results and finally conclusion in drawn in Sect. 5 with future works.

2 Related Work

Numerous researchers have worked on abusive review detections. In this section, we will discussing some of the contributions made by different researchers found in the literature in this field. Authors in [1] proposed a root level algorithm and unigram string feature technique to detect abusive Bangla comment. They have applied their algorithm in three different sized dataset (100, 200, 300) and obtained an accuracy of 70%, 65%, and 71.7%, respectively. Authors in [2] proposed random forest (RF), MNB, and SVM with linear, radial basis function, polynomial, and sigmoid kernel. They used both CountVectorizer and TF-IDF Vectorizer. They achieved the highest performance with SVM linear kernel and trigram TF-IDF Vectorizer features. Authors in [3] proposed different machine learning and deep learning techniques to detect abusive Bangla text. They labeled their dataset into seven different classes and achieved best performance using RNN with an accuracy

[2] https://datareportal.com/reports/digital-2020-bangaldesh.

[3] https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989.

[4] https://scikit-learn.org/stable/modules/sgd.html?fbclid=IwAR2IwLILKWIq-n3UfcAF3BNXgqmKSuvAPSF0m71GXkO7DlpFY-1z5d75xMA.

of 82.20%. Authors in [4] used Bangla-English mixed text and transliterated Bangla text. They used features like unigram, bigram, number of likes, emojis, sentiment scores, offensive, and threatening words and proposed SVM, RF, and AdaBoost. They achieved a highest accuracy of 72.14%. In [5], authors made a comparative study between classical machine learning model and deep learning models. They have used CNN, RNN with SVM for this purpose. They consider both n-gram and word embedding features. They used nine different dataset for this purpose. They showed that for imbalanced data deep learning performs better and word embedding works effectively when pretrained on the same data source as the training dataset. SVM performs better when there is a balance between class labels, and balancing can be obtained through the process of oversampling. In [6], authors proposed Vowpal Wabbit's regression model on finance and news dataset collected from Yahoo. They considered n-gram, linguistic, syntactic, and distributional semantics features and obtained an F-score of 0.795 and 0.817 for finance and news data, respectively. Authors in [7] proposed Naïve Bayes, SVM, and random forest decision tree classifier for detecting abusive reviews from Indonesian tweet. They considered word n-gram and char n-gram features. An F-score of 70.06% was obtained while detecting offensive, non-offensive abusive, and non-abusive reviews. Their F-score increased to 83% for classifying abusive and non-abusive reviews only. Authors in [8] proposed an approach to detect both emotion and abusiveness from texts. They used single-task learning and obtained an F-score of 74.40 using Bi-LSTM and attention classifier with GloVe and ELMo embeddings for O ensEval dataset and F-score of 78.33 using max pool and MLP classifier with GloVe embeddings for Waseem and Hovy dataset. They also used multi-task learning and obtained an F-score of 76.03 with GatedDEncoder on OffensEval dataset and 79.55 for Waseem and Hovy dataset. Authors in [9] proposed machine learning algorithm to detect abusive accounts from Arabic tweets. They used three different sets of features for this purpose. However, the highest accuracy of 90% was obtained for Naïve Bayes classifier with 10 tweets and 100 features. In [10], authors proposed a semi-supervised approach for detecting offensive content in Twitter. They compiled a dictionary of 338 most offensive words and manually removed frequently used but less offensive words from tweets and implemented four machine learning algorithms, namely LR, SVM, decision tree, and random forest and obtained best performance using LR with a true positive rate of 75.1% over 4029 testing tweets.

3 Methodology

This section will describe the methodology we have adapted for abusive Bangla comment detection. Our proposed methodology is shown in Fig. 1. Though there are numerous machine learning algorithms available that can be applied for classification purpose but different algorithm shows different performance based on the size and type of dataset. Again, accuracy may not always be a good parameter to

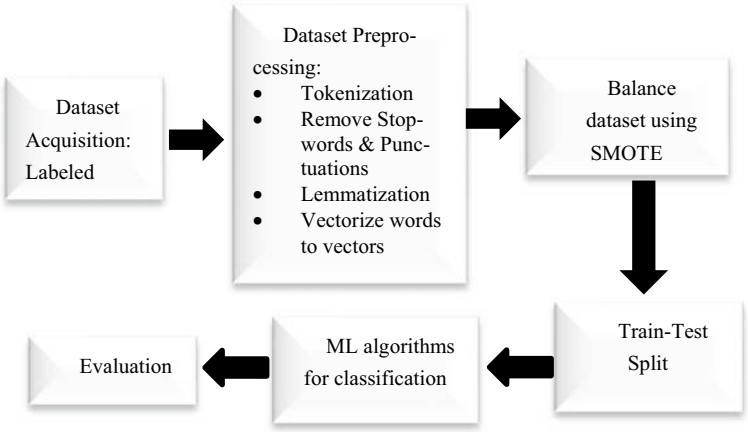


Fig. 1 Block diagram of proposed methodology

detect the best model. It depends on a number of other factors as well. In this research, we have used classic machine learning algorithms such as logistic regression (LR), support vector machine (SVM), multinomial Naïve Bayes (MNB), and SGD with modified Huber (stochastic gradient descent) along with ensemble models like AdaBoost, gradient boosting (GB) and bagging classifier (BC) to evaluate the result and found the SVM and SGD with loss function set to modified Huber algorithms using TF-IDF Vectorizer as the most suited model for our research work.

Our work is divided into the following steps:

3.1 Dataset Preprocessing

Tokenization: Tokenization is an important step for fitting text data into ML models. Tokenization tokenize each individual word of a sentence in order to create word vector from them. We consider blank space in order to identify words from a sentence. Below is a sample Bangla comment with its tokenized form:

Sentence: এবাররে বজিয় দবিস একটু বশেইি কাঙ্ক্ষতি।

Tokenized text: ‘এবাররে’, ‘বজিয়’, ‘দবিস’, ‘একটু’, ‘বশেইি’, ‘কাঙ্ক্ষতি।’.

Removal of stopwords and punctuations: Stopwords are the words that have little or almost no importance in abusive comment detection. But stopwords increase the burden for a machine to be trained since it has to deal with a lot of unnecessary words. So, we removed all the stopwords. In English text, there are dictionaries for stopwords, but in Bengali, we did not find such dictionary. So, we used a total of 398 stopwords of Bengali language from our previous work [11]. Eg. অতএব, ইত্যাদি, এই, etc. Again, since we are dealing with only texts, we removed all the punctuations used in the dataset.

Lemmatization: Lemmatization is the process of transforming a word into its root word. A word can be found in different form depending on geographical location of people speaking in it and other factors. Since, all forms of a word carry the same meaning, and we transform them into their basic form in order to reduce the workload of machine. E.g., খলেতিছেে, খলো, খলেনা, খলেছলিাম all of these words have root word খলো.

3.2 Balancing the Dataset

Our dataset was highly imbalanced. We had only 18.6% of our dataset labeled as abusive leaving a huge number of non-abusive comments. This could result in biasing of the model. Therefore, we have used synthetic minority oversampling technique (SMOTE) which oversample the abusive class by generating synthetic data. Thus, we had equal number of abusive and non-abusive comments.

3.3 Splitting into Training and Testing Dataset

In order to evaluate the performance of our model, we have splitted our dataset into 80% and 20% for training and validation purpose, respectively. After that, we train our model and evaluate the performance using the validation dataset.

3.4 Vectorizer

Since, machine learning models can only deal with numeric values, texts should be converted to number. A number of vectorizers work for this purpose. We have implemented TF-IDF Vectorizer for our research work. Term frequency (TF) gives a measure of a frequency of a word in a document. Term frequency for a word is a ratio of no. of times a word appears in a document to total no. of words in the document. Inverse document frequency is a measure of the importance of the word. It measures how common a particular word is across all the document in the corpus. However, for comparison purpose, we have also tested with HashingVectorizer and CountVectorizer as well. We used a combination of unigram, bigram, and trigram feature extraction techniques to have more features for text categorization. Unigram, bigram, and trigram are used to express a string of words. Unigram tokenize each individual word belonging to a sentence, bigram picks two successive words and tokenize them, whereas in case of trigram, three successive words are tokenized together. We combined unigram, bigram, and trigram in an attempt to find the feature importance of each individual word as well as how words are interrelated with each other in a sentence in an attempt to express the abusiveness.

4 Experimental Results and Evaluation

This section describes the dataset, and we have used and detail about our experimental results that we found from our research work.

4.1 Dataset Description

We have used a labeled dataset consisting a total of 24,266 Bangla text collected from different online repositories out of which 4502 were abusive and 19,764 were non-abusive. Table 1 shows some sample data from our dataset with label. In our dataset, 0 represents non-abusive reviews, and 1 represents abusive reviews. Figure 2a represents a pie chart showing the ratio between two labels in percentage in original dataset, and Fig. 2b shows a pie chart after oversampling.

4.2 Results and Discussion

We have used seven different machine learning algorithms and used TF-IDF Vectorizer to generate word vector from our dataset. We explored that combination of unigram, bigram, and trigram feature helps our proposed model to perform better than using them alone. The reason behind this is when all the features are combined, it creates a lot of information for the machine to learn effectively. Therefore, we considered the combination of unigram, bigram, and trigram feature extraction technique. Then, we trained each model and generate a ROC-AUC to find the best model. Table 2 shows the accuracy generated by all the models using unigram, bigram, and trigram features both for individual and combined approach. Figure 3 shows the ROC-AUC curve obtained from our study for all the algorithms used. The maximum AUC value of 93% was obtained from our experiment by SGD and SVM model. Since both SGD and SVM generated equal AUC value, we considered other parameters like accuracy, precision, recall, and F-score to compare the performance of the models. Table 3 shows the value of these parameters obtained from our experiment for TF-IDF Vectorizer. The number of true negative, false positive,

Table 1 Sample dataset

Comments	Label
অযোগ্য বলেই মাসকি বতেন ২ লাখ ডলার নতিে পারনেি!	0
ভাষা হারয়িে ফলেলাম!	0
স্যার মনে হই প্রসেডিন্টে হতে চায়	0
ওরে পাঠার পনে পাঠা।	1
তনেরে জুতা দয়িা পডিামনে	1

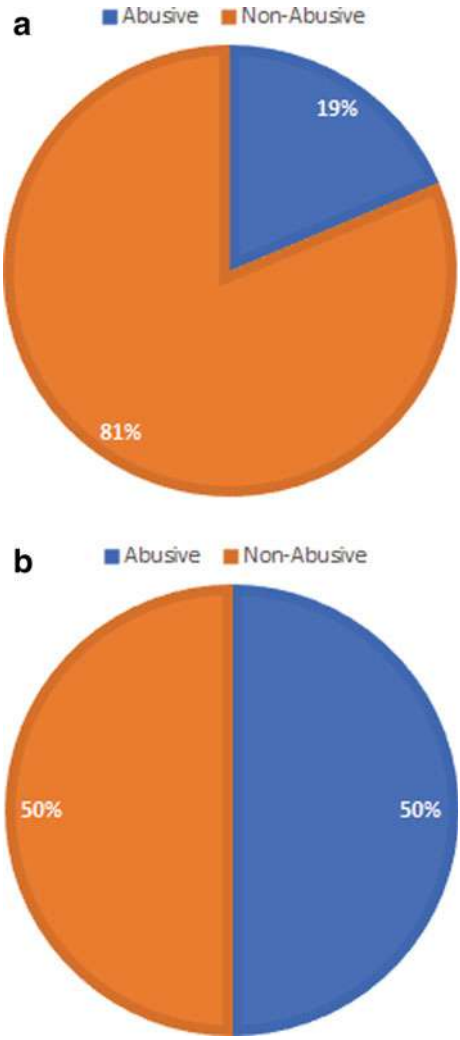


Fig. 2 **a** Ratio between abusive and non-abusive comments (original dataset). **b** Ratio between abusive and non-abusive comments (after oversampling)

false negative, true positive are, respectively, 3530, 464, 120, 3792 for SGD and 3536, 458, 127, 3785 for SVM.

The performance of a model is dependent largely on the method of vectorization. So, we tested with two other vectorizers called CountVectorizer and HashingVectorizer on our preprocessed text and fit the data into all the models and evaluated their performance on these setups. The HashingVectorizer generated some negative vectors which is not supported in MNB, so we could not evaluate the performance of MNB using HashingVectorizer. Figures 4 and 5 show the ROC-AUC curve of the different models using HashingVectorizer and CountVectorizer, respectively. Tables 4 and 5 show the accuracy, precision, recall,

Table 2 Comparison of models' accuracy using unigram, bigram, and trigram features alone and combination of them

Models	Accuracy (unigram) (%)	Accuracy (bigram) (%)	Accuracy (trigram) (%)	Accuracy (combined) (%)
LR	89	**81**	61	91
SVM	90	79	**70**	**93**
SGD	90	**81**	59	**93**
MNB	82	80	55	92
AdaBoost	80	58	51	80
GB	83	60	53	83
BC	**91**	76	58	91

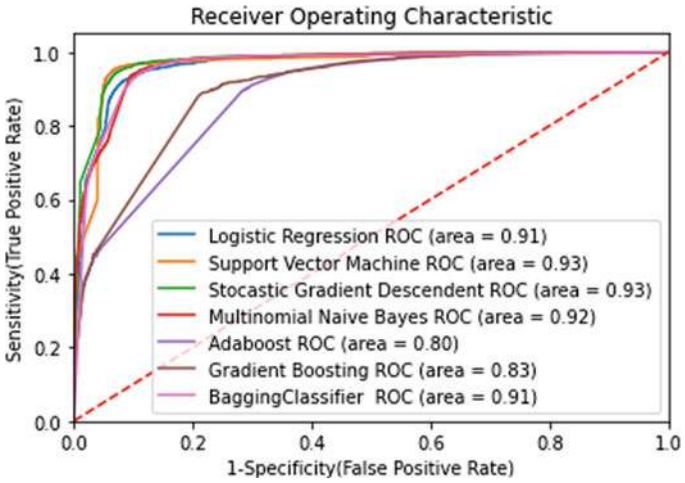


Fig. 3 ROC-AUC curve of seven different models

Table 3 Performance of the model for different classifiers (using TF-IDF)

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	ROC-AUC (%)
LR	91	88	95	91	91
SVM	**93**	**89**	**97**	**93**	**93**
SGD	**93**	**89**	**97**	**93**	**93**
MNB	92	**91**	93	92	92
AdaBoost	80	75	90	82	80
GB	83	78	91	84	83
BC	91	88	95	91	91

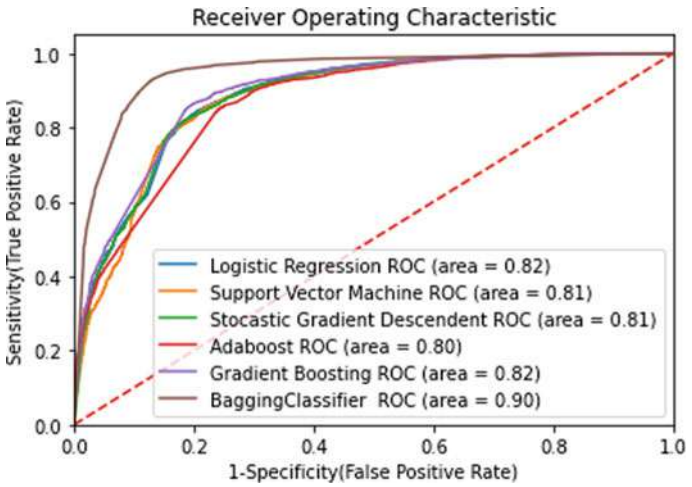


Fig. 4 ROC-AUC with HashingVectorizer

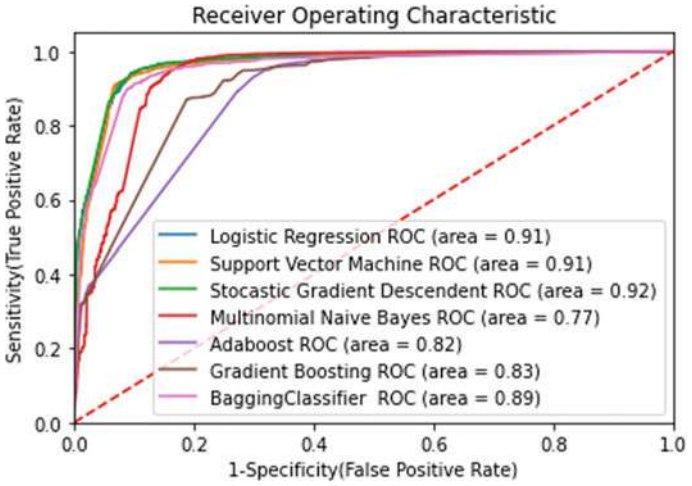


Fig. 5 ROC-AUC with CountVectorizer

and F-score obtained for the models using hashing and CountVectorizer, respectively.

Figure 6 shows the model accuracy for different classifiers using three different vectorizers used in our experiment.

From the above experimental analysis, it has been found that bagging classifier generates the accuracy of 90% for HashingVectorizer. Also, bagging classifier generated the best F-score and AUC of 90%. For CountVectorizer, SGD with modified Huber was found to be the best models with an accuracy of 92% and

Table 4 Performance of the model for different classifiers (using HashingVectorizer)

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	Roc-AUC (%)
LR	82	80	86	82	82
SVM	81	77	89	82	81
SGD	81	79	85	82	81
AdaBoost	80	76	87	81	80
GB	82	78	90	84	82
BC	**90**	**88**	**92**	**90**	**90**

Table 5 Performance of the model for different classifiers (using CountVectorizer)

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	ROC-AUC (%)
LR	91	88	**96**	**92**	91
SVM	91	87	95	91	91
SGD	**92**	**89**	**96**	**92**	**92**
MNB	77	**89**	62	73	77
AdaBoost	82	76	93	83	82
GB	83	77	95	85	83
BC	89	84	95	89	89

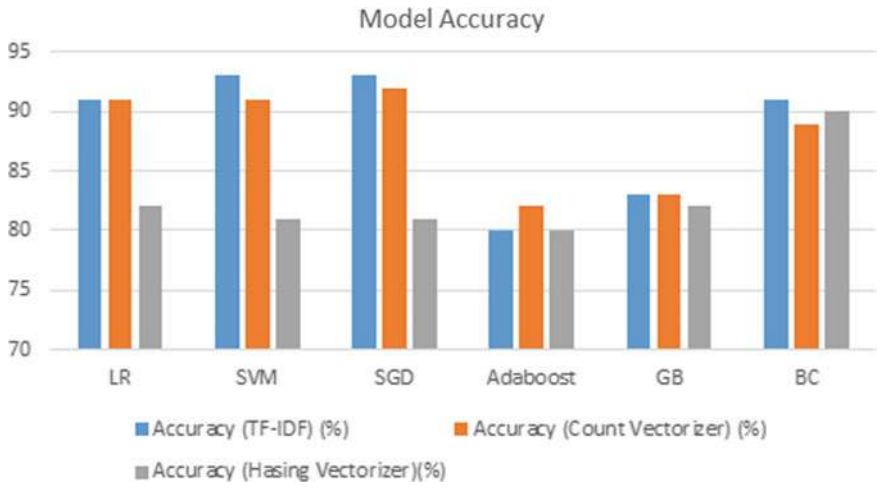


Fig. 6 Accuracy of different classifiers for three different vectorizers

F-score of 92%. So, based on the comparison among different vectorizers, TF-IDF showed the best performance for our proposed model. So, in summary, support vector machine and stochastic gradient descendent (SGD) with loss function set to

modified Huber algorithms using TF-IDF Vectorizer generate an accuracy of 93%, AUC of 93% with F-score of 93% which is more than all other models used in our experiment.

5 Conclusion and Future Work

Abusive comment detection is a very important aspect of natural language processing. While detecting abusive review from Bangla text is becoming a must since people are now making a lot of posts and comments in different social media's and blogs in Bangla. In this paper, we ran a comparative study among different machine learning algorithms to find the most suitable model for abusive comment detection from Bangla text. We first preprocessed the dataset for making it adaptable for the models and then oversampled the abusive class for balancing both classes using SMOTE. We used three different vectorizers and considered combination of unigram, bigram, and trigram feature selection approach to generate word vector from our dataset. Support vector machine (SVM) and stochastic gradient descendent (SGD) with loss function set to modified Huber algorithms using TF-IDF Vectorizer generate highest accuracy and F-score than all other ML models used in our experiment. We still faced some difficulties in proper preprocessing of our dataset. The reason behind it is the improper labeling of the dataset, lack of adequate stopwords, and lemmatizer does not perform well compared to works on other languages. In this paper, we have done a binary classification to detect either a comment is abusive or non-abusive. In our future work, we will focus on detecting the nature of abusiveness through multi-level classification of reviewers' comments. We will also try using deep learning approach like RNN, ANN in our future work.

References

1. Hussain MG, Mahmd AIT (2019) A technique for perceiving abusive Bangla comments. GUB J Sci Eng 4(1):11–18
2. Eshan SC, Hasan MS (2017) An application of machine learning to detect abusive Bengali text. In: 2017 20th International conference of computer and information technology (ICCIT). IEEE, December, pp 1–6
3. Emon EA, Rahman S, Banarjee J, Das AK, Mittra T (2019) A deep learning approach to detect abusive Bengali text. In: 2019 7th international conference on smart computing & communications (ICSCC). IEEE, June, pp 1–5
4. Jahan M, Ahamed I, Bishwas MR, Shatabda S (2019) Abusive comments detection in Bangla-English code-mixed and transliterated text. In: 2019 2nd international conference on innovation in engineering and technology (ICIET). IEEE, December, pp 1–6
5. Chen H, McKeever S, Delany SJ (2018) A comparison of classical versus deep learning techniques for abusive content detection on social media sites. In: International conference on social informatics. Springer, Cham, September, pp 117–133

6. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y (2016) Abusive language detection in online user content. In: Proceedings of the 25th international conference on world wide web, April, pp 145–153
7. Ibrohim MO, Budi I (2018) A dataset and preliminaries study for abusive language detection in Indonesian social media. Procedia Comput Sci 135:222–229
8. Rajamanickam S, Mishra P, Yannakoudakis H, Shutova E (2020) Joint modelling of emotion and abusive language detection. arXiv preprint. arXiv:2005.14028
9. Abozinadah EA, Mbaziira AV, Jones J (2015) Detection of abusive accounts with Arabic tweets. Int J Knowl Eng-IACSIT 1(2):113–119
10. Xiang G, Fan B, Wang L, Hong J, Rose C (2012) Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM international conference on Information and knowledge management, October, pp 1980–1984
11. Saha P, Sultana N (2021) Sentiment analysis from Bangla text review using feedback recurrent neural network model. In: Communication and intelligent systems. Springer, Singapore, pp 423–434

Chapter 46
A Reconfigurable Antenna for 5G, 4G, and WLAN Applications

T. A. Balarajuswamy and Nakkeeran Rangaswamy

1 Introduction

Reconfigurable antennas can change their frequency, polarisation, and radiation qualities dynamically in a controlled and reversible manner. In reaction to changes in their surroundings, they adjust their shape and behaviour to enhance antenna function. They use PIN diodes, varactors, radio-frequency microelectromechanical systems (RF-MEMS), field effect transistors (FETs), parasitic pixel layers, photoconductive components, mechanical actuators, metamaterials, ferrites, and liquid crystals to create a dynamical reaction. The ability to reconfigure the operating frequency, radiation pattern, and polarizations of antennas is extremely desirable due to the increasing rise of wireless communications and the increasing need for the integration of many wireless standards onto a single platform [1]. Reconfigurable antennas change their operating frequency, impedance bandwidth, polarization, and radiation pattern in response to the host system's operational needs. Multiple single-purpose antennas provide the same functionality as a reconfigurable antenna. Costs, weight, bulk, and maintenance/repair resources are all reduced as a result of this. One of the most essential properties of antennas is their radiation pattern. It is a diagram that shows the antenna's far-field radiation characteristics. It also refers to an antenna's capacity to transmit and receive signals in a certain direction.

When a PIN diode is forward biased, it has low resistance and tends to lengthen the radiating arms, which results in a lower resonance frequency. When a PIN diode is reverse biased, it becomes capacitive, and the length of the radiating arms tends to shorten, shifting the resonant frequency higher.

The design of a multiband reconfigurable seven-shaped planar monopole antenna was described by Shah et al. in [2]. The antenna is made of FR4 substrate and has

T. A. Balarajuswamy (✉) · N. Rangaswamy
Department of Electronics Engineering, School of Engineering and Technology, Pondicherry University, Pondicherry 605014, India
e-mail: tabalarajuswamy67@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_47

a thickness of 1.6 mm with a truncated metal ground plane. It works in single band (WiMAX at 3.5 GHz) and dual band (Wi-Fi at 2.45 GHz and WLAN at 5.4 GHz) depending on the switch position (ON and OFF state). For all three frequency bands, the antenna design exhibits impedance matching of 51 to 45 ohms and a VSWR of less than 1.3. At 2.45, 3.5, and 5.4 GHz, reasonable gains of 1.92, 2.41, and 3.02 dBi were attained, respectively.

Goutham and Sahay in [3] used a single element to create a multiband frequency-reconfigurable antenna. The frequency and reflection coefficient of this antenna are considerably affected by the two rectangular slots. Using a single switch, the developed antenna provides two extremely useful frequency bands, one for WiMAX (3.5 GHz) and the other for Wi-Fi (5.8 GHz). In the ON state, the antenna has dual bands at 3.5 GHz and 5.8 GHz, but in the OFF state, it only has a single band at 5.8 GHz. The assessment reports demonstrate that the simulated and created results are quite similar. Boukarkar et al. in [4] suggested a unique multiband monopole antenna for UMTS, ISM, INSAT, and WLAN bands for diverse wireless applications. This multiband antenna has a small footprint and a straightforward design, allowing it to cover more than three frequency bands with just two switches. By employing a small number of switches, the complexity of DC biasing is lessened. The antenna is built and measured, and the results reveal that the return loss is in good accord. Mun et al. in [5] suggested a frequency-reconfigurable MIMO antenna for laptop applications that is small and light (multiband LTE services). Every MIMO antenna consists of two planar inverted-F antenna (PIFA) components connected by a T-shaped DC line and two proximity-coupled feed PIN diodes (D1 and D2). The T-shaped DC line that was built effectively reduces the line's size and interference. The planned MIMO antenna encircles the LTE 17/13 bands when the diodes are turned on, and the LTE 20/7 bands when the diodes are turned off. The antenna is designed to provide excellent isolation (>20 dB) and overall efficiency (<50%). Majid et al. in [6] proposed frequency-reconfigurable microstrip patch-slot antenna with reflector at the back of antenna, capable to reconfigure up to six different frequency bands from 1.7 to 3.5 GHz.

Abdulraheem et al. in [7] suggested a multiband reconfigurable antenna for WiMAX and WLAN applications that uses two PIN diodes as switches. The antenna may vary between up to ten different frequency bands between 2.2 and 6 GHz. The suggested antenna is made of FR4 substrate and has a 50×45 mm^2 area, according to CST microwave studio software. The optimized antenna is tested and modelled return loss characteristics reveal that it meets the requirements of WLAN (2.4/5.8 GHz) and WiMAX (3.5 GHz) applications. Furthermore, there is good agreement in terms of gain and radiation pattern between the tested and simulated findings. Ejaz et al. in [8] proposed a tunable multiband micro strip antenna is designed, capable of tuning its operating frequency in 5 GHz band independently. Two slots of E and U shape are etched to achieve multiband resonance. Two capacitors of fixed value and a varactor diode are used to achieve tuning. Yeole and Khot in [9] suggested a reconfigurable microstrip patch antenna that can operate in many frequency bands, including L, S, C, and X. On ADS software, the rectangular patch is created with a U-shaped slotted antenna with dimensions of 38 mm $\times$ 61 mm and three pin diodes as switches. As

a result, the antenna may be used for a variety of wireless applications, including Wi-Fi, WiMAX, UWB, and 3G, with a single patch. The main advantage of this antenna design is its low profile and light weight. A novel circular slotted multiband antenna for wireless communication applications was presented by Saleem and Quddus in [10]. The antenna is built on a FR4 substrate with circular slots at the ground plane, a 50 Ω feeding line, and a rectangular shape slit connected to the circular slots. Furthermore, by aiming PIN diodes towards the circular slots of DGS, multiband reconfigurable resonances may be created. The proposed antenna has three tunable resonant frequency bands: 2.5, 3.5, and 5.7 GHz, which are employed in WLAN, WiMAX, and UWB applications. With dimensions of 28 mm × 16 mm × 0.8 mm, it is a good contender for a variety of wireless communication applications. Kaur and Asok De in [11] demonstrated a low-cost, multiband, and wideband reconfigurable antenna. These requirements are met by the antenna design support. By which these antennas provide a variety of benefits while also allowing the reconfigurable antennas' performance to be increased via the use of the reconfiguration idea. The reconfigurable fractal antenna that was built combines the advantages of both circumstances. Ansoft high frequency structure simulator is used to design, model, and optimize a reconfigurable fractal antenna (HFSS). The vector network system is used to construct and measure the optimal antenna. Reconfigurable radios are becoming increasingly common in wireless communication systems due to their ability to operate over a wide frequency range with the same hardware.

Jin et al. in [12] presented a differential frequency-reconfigurable antenna based on dipoles, and the structure of the antenna consists of two pairs of vertical arms forming two dipoles, the feeding structure, mode switching structure, and pin diodes. By switching on and off pin diodes, the antenna can resonate at two states at 3.5 and 5.5 GHz, respectively. The antenna is also simple in structure and wide in bandwidth. The measured −10 dB impedance bandwidths are 2.89–4.07 GHz (33%) and 5.1–6.19 GHz (19.8%) for the two states, respectively.

Desai et al. in [13] presented a dual branch multiband compact slotted antenna for digital broadcasting, wireless and sub 6 GHz 5G. Antenna achieves multiband characteristics at (750–790 MHz) digital broadcasting, (1.41–1.45 GHz) wireless medical telemetry, (2.10–2.14 GHz) universal mobile telecommunication service (UMTS), (3.44–3.51 GHz) WiMAX/sub 6 GHz 5G, (3.80–3.87 GHz) sub 6 GHz 5G, (5.17–5.20 GHz) wireless local area network (WLAN), and (5.97–6.38 GHz) fixed satellite communication by introducing slots in the form of inverted U and E shapes. The proposed antenna achieves compact operation along with an omnidirectional radiation pattern having gain and efficiency in the range over 1.1 dBi and 78% for the proposed frequency bands.

Two PIN diodes are employed as switching devices for frequency reconfiguration in this work. These two PIN diodes switch the antenna for four distinct applications depending on the state of the switch, including 5G upper band at 24.25–27.5 GHz, WLAN at 5.5–5.6 GHz, 5G lower band at 3.3–3.4 GHz, and 4G applications at 2.3–2.4 GHz. T-shaped patch, inner C-shaped rectangular patch, and outer C-shaped rectangular patch on top, substrate in the middle, and ground plate at the bottom make up the structure.

1.1 Materials and Methods

Two C-shaped rectangular patches and one T-shaped patch make up the proposed antenna. Patch 1 is a smaller T-shaped structure constructed of pure copper that helps create the frequency range of 24.25–27.5 GHz. Patch 2 is a smaller C-shaped structure constructed of pure copper that aids in the generation of the 3.3–3.4 GHz frequency. Patch 3 is a bigger C-shaped structure constructed of pure copper that aids in the generation of the 2.3–2.4 GHz frequency. The 5.5–5.6 GHz frequency is generated with the aid of the substrate and ground plate. The ground plate is composed of pure copper while the substrate is constructed of FR4. The substrate measures 75 × 99 mm^2, has a relative dielectric constant of 4.7, loss tangent of 0.019, and a thickness of 1.67 mm, and has a relative dielectric constant of 4.7. The ground plate is 75 × 99 mm^2 in dimension and 0.035 mm in thickness. The 5.5–5.6 GHz frequency is generated with the aid of this substrate plate and ground plate. This antenna is made up of two PIN diode switches labelled SW1 and SW2. SW1 is positioned between patch 1's top and patch 2's bottom. SW2 is sandwiched between patch 2's top and patch 3's bottom. The two switches provide four separate situations, each of which provides a distinct frequency for each application. The design of the reconfigurable antenna is shown in Fig. 1. Table 1 shows the numerical values of the antenna's design parameters.

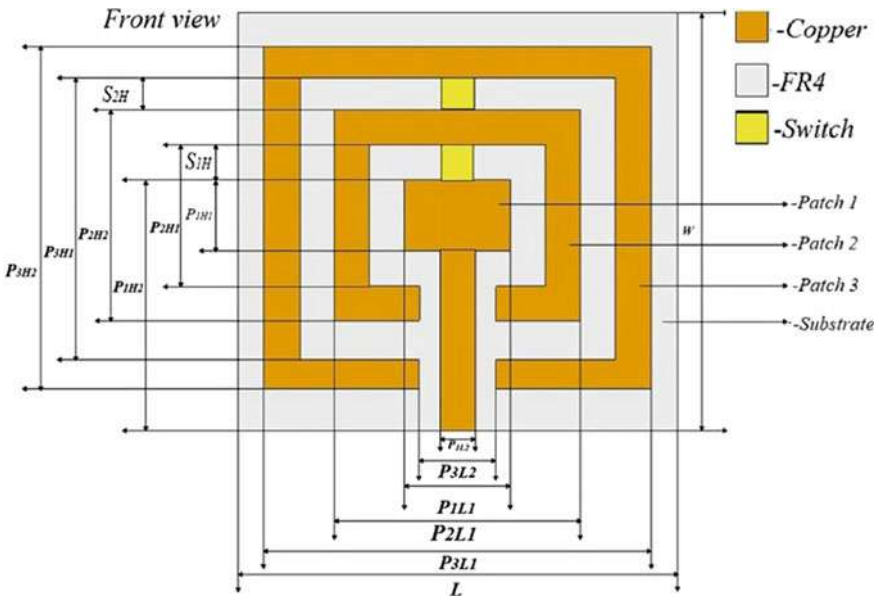
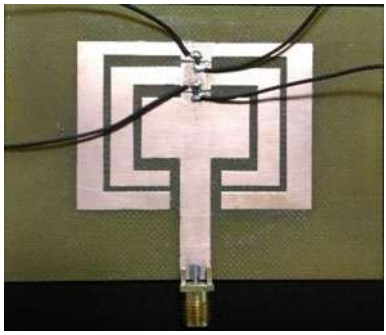


Fig. 1 Design of reconfigurable antenna

Table 1 Dimension of the proposed antenna

Parameter	Value (mm)	Parameter	Value (mm)
W	75	S2H	3
L	99	S1H	3
P3H2	52	P3L1	70
P3H1	35	P2L1	41
P2H2	28	P1L1	25
P2H1	20	P3L2	10
P1H2	43	P1L2	6
P1H1	14		

Fig. 2 Prototype of reconfigurable antenna



2 Results and Discussion

CST Studio Suite software was used to create the proposed antenna's design and development. Return loss, gain, bandwidth, efficiency, and directivity were all considered while evaluating the antenna's performance. The prototype of an optimized antenna construction was created, and the measured and simulated values were found to be in agreement. However, the discrepancies are related to the influence of soldering on switches and connectors, as well as manufacturing tolerances. Figure 2 depicts the proposed antenna's prototype, while Fig. 3 depicts the antenna's test setup. Table 2 depicts various reconfigurable antenna switching circumstances. Table 3 shows the measured value of proposed structure.

3 Switch S1 off and Switch S2 off

When S1 and S2 are both turned off, the antenna resonates at 26.8 GHz shown in Fig. 4. The T-shaped patch aids in the generation of 5G upper band frequencies between 24.25 and 27.5 GHz. When both switches are turned off, higher current flows through the T-shaped patch structure, while medium current flows through the

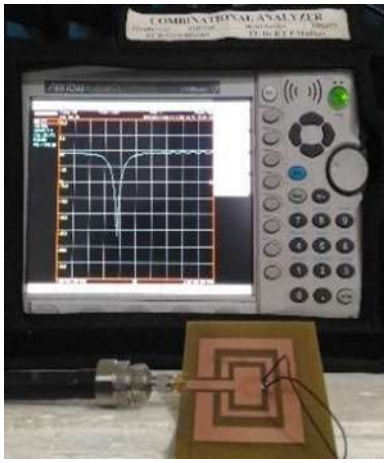


Fig. 3 Test setup of antenna

Table 2 Different switching conditions of reconfigurable antenna

Switching condition	Sw 1	Sw 2	Resonant frequency	Application of the frequency
Condition 1	Off	Off	24.25–27.5 GHz	5G upper band
Condition 2	Off	On	5.5–5.6 GHz	WLAN
Condition 3	On	Off	3.3–3.4 GHz	5G lower band
Condition 4	On	On	2.3–2.4 GHz	4G lower band

Table 3 Measured value of proposed structure

Switching condition	Frequency (GHz)	Return loss (dB)	VSWR	Gain (dBi)	Radiation efficiency (%)	Application	
S1 off S2 off	26.83	−34.47	1.03	7.2	83	5G upper band	
S1 off S2 on	5.55	−16.11	1.29	5.5	84	WLAN	
S1 on S2 off	3.364	−11.11	1.6	1.65	62	5G lower band	
S1 on S2 on	2.303	−17.85	1.19	6.5	76	4G band	

top and bottom of the inner and outer C-shaped rectangle structures. The antenna has a return loss of −36.8 dB and a VSWR of 1.04. The antenna's directivity is 13.3 dBi. Figure 5 illustrates the gain and efficiency of antenna, and Fig. 6 shows return loss of Antenna. Figures 7 and 8 show the E-Plane and H-Plane of antenna, respectively.

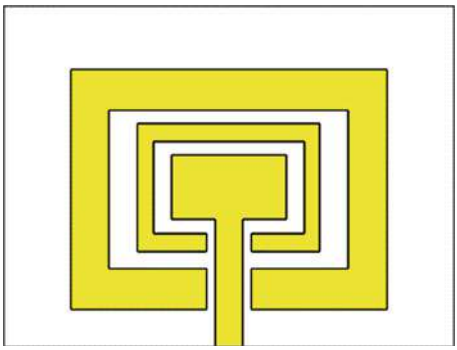


Fig. 4 Simulated structure of antenna

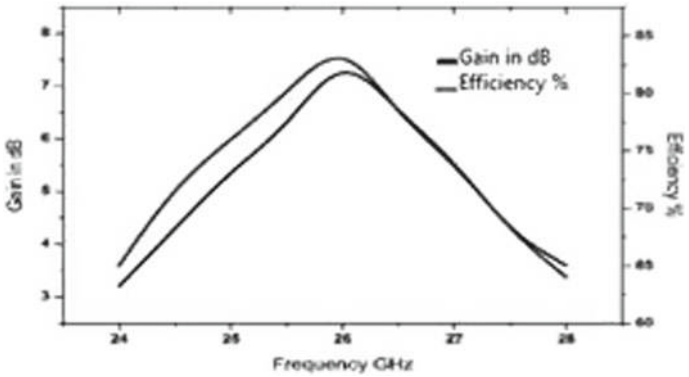


Fig. 5 Gain and efficiency of antenna

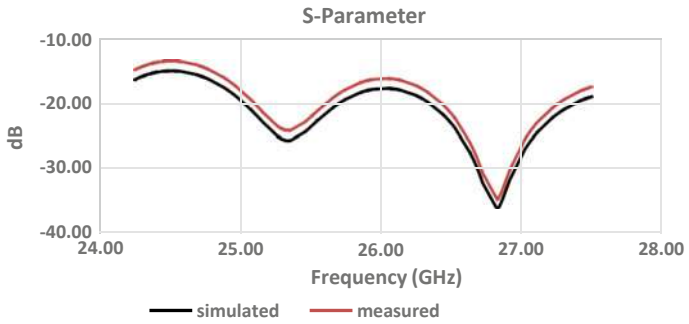


Fig. 6 Return loss of antenna

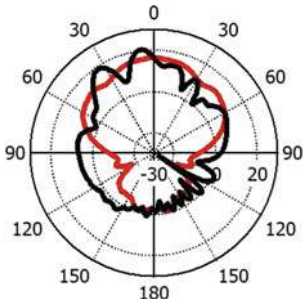


Fig. 7 E-Plane of antenna

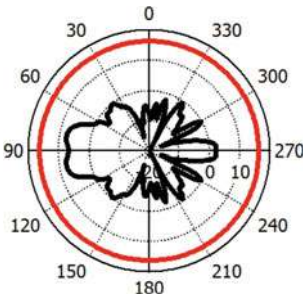


Fig. 8 H-Plane of antenna

3.1 Switch S1 off and Switch S2 on

When S1 is turned off and S2 is turned on, the antenna resonates at 5.5 GHz shown in Fig. 9. When S1 is turned off and S2 is turned on, the T-shaped patch receives higher current, while the inner and outer C-shaped rectangular patches get medium current. When S2 is forward biased, it has a low resistance and, as a result, the resonant frequency is reduced to 5.5 GHz. Figure 10 shows the gain and efficiency of antenna. With a VSWR of 1.3, the antenna has a return loss of −18 dB shown in Fig. 11. The

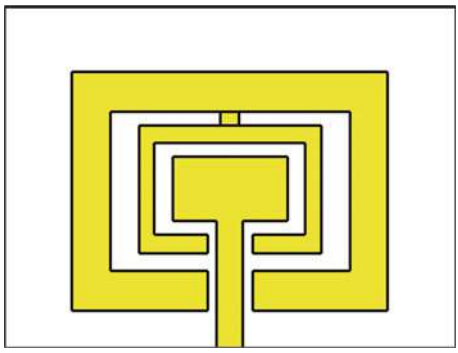


Fig. 9 Simulated structure of antenna

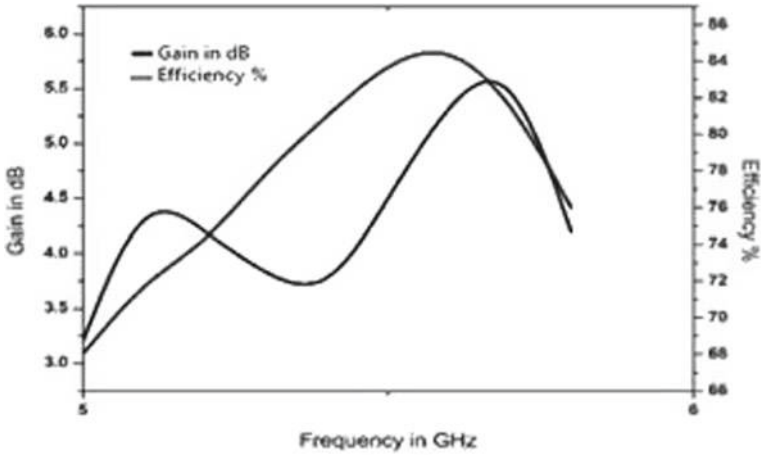


Fig. 10 Gain and efficiency of antenna

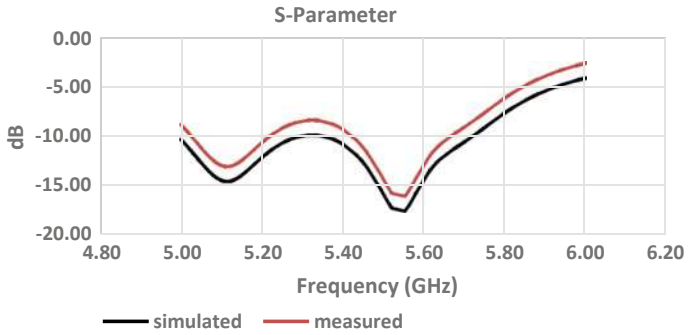
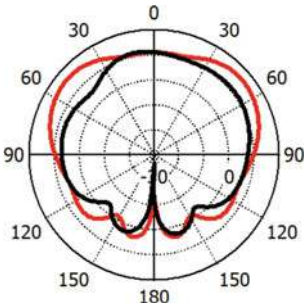


Fig. 11 Return loss of antenna

Fig. 12 E-Plane of antenna



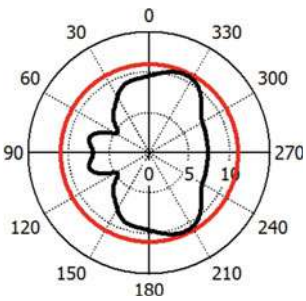


Fig. 13 H-Plane of antenna

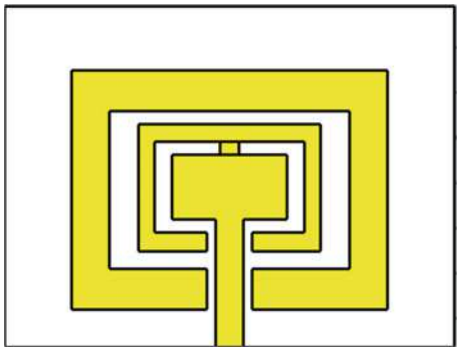


Fig. 14 Simulated structure of antenna

antenna's directivity is 6.14 dBi. Figures 12 and 13 depict the E-Plane and H-Plane of antenna, respectively.

Switch S1 on and Switch S2 off. When S1 is on and S2 is off, antenna is resonating at 3.36 GHz shown in Fig. 14. The inner C-shaped rectangular patch helps to generate 3.3–3.4 GHz of 5G lower band. When S1 is on and S2 is off, more current flows through T-shaped patch and the inner C-shaped rectangular patch and very fewer current flows through outer C-shaped rectangular patch. When S2 is forward biased, it offers low resistance and hence, lowers the resonant frequency to 3.36 GHz. Figure 15 shows the gain and efficiency of antenna. The return loss of antenna is −12 dB shown in Fig. 16 with VSWR of 1.6. The directivity of antenna is 5.38 dBi. Figures 17 and 18 illustrate the E-Plane and H-Plane of antenna, respectively.

Switch S1 on and Switch S2 on. When S1 and S2 are turned on, the antenna resonates at 2.3 GHz shown in Fig. 19, allowing the 4G band to be generated between 2.3 and 2.4 GHz. More current flows via T-shaped patch, inner and outer C-shaped rectangular patch when S1 and S2 are both active. S1 and S2 give low resistance when both are forward biased, lowering the resonant frequency to 2.3 GHz. Figure 20 shows the gain and efficiency of antenna. With a VSWR of 1.19, the antenna has a return loss of −19.58 dB shown in Fig. 21. The antenna's directivity is 6.63 dBi. Figures 22 and 23 illustrate E-Plane and H-Plane of antenna.

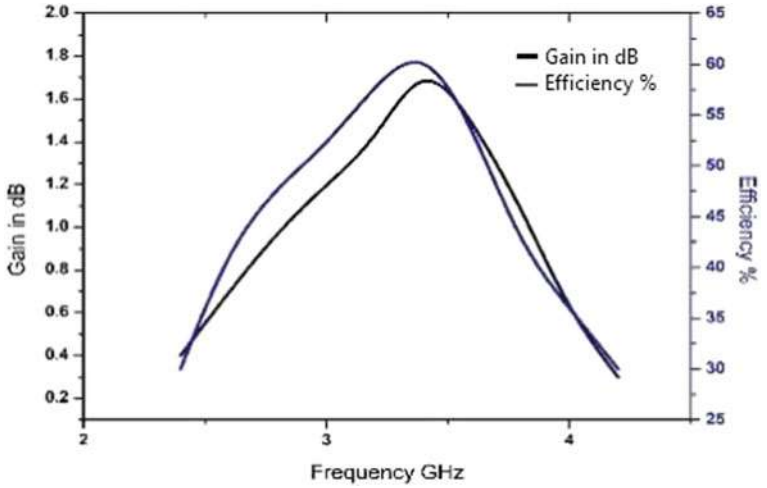


Fig. 15 Gain and efficiency of antenna

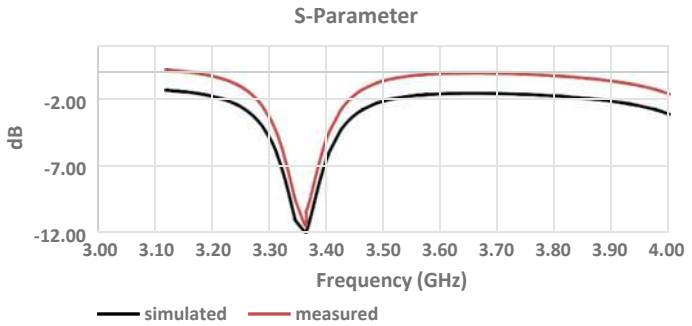
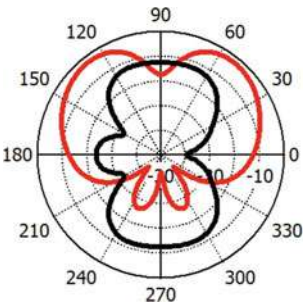


Fig. 16 Return loss of antenna

Fig. 17 E-Plane of antenna



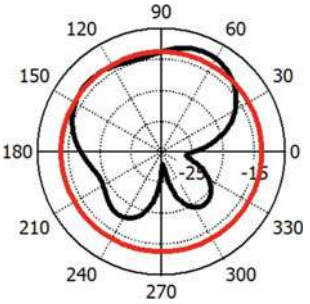


Fig. 18 H-Plane of antenna

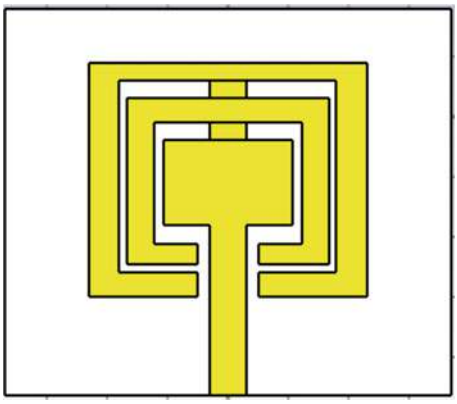


Fig. 19 Simulated structure of antenna

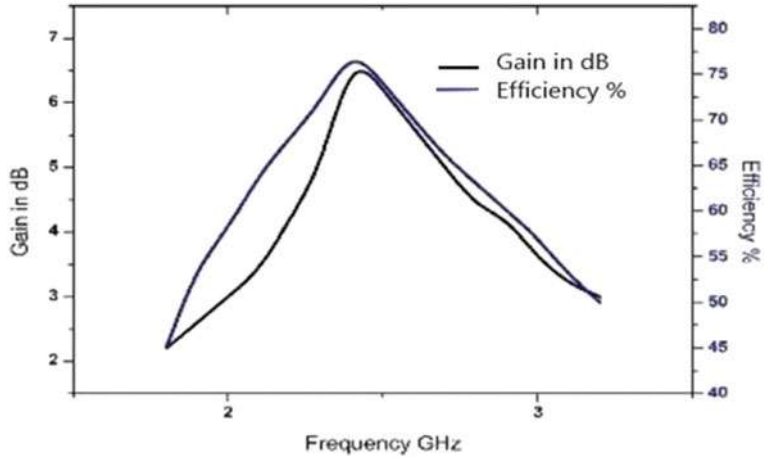


Fig. 20 Gain and efficiency of antenna

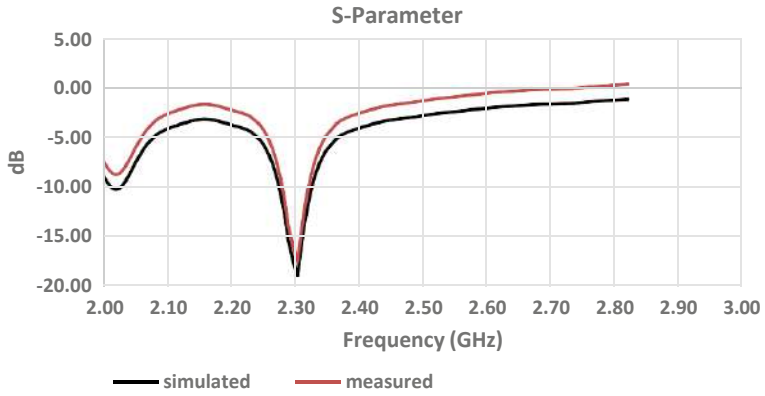


Fig. 21 Return loss of antenna

Fig. 22 E-Plane of antenna

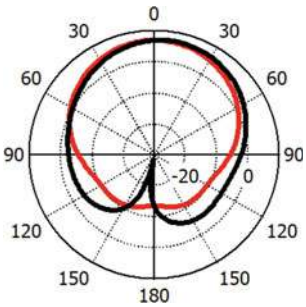
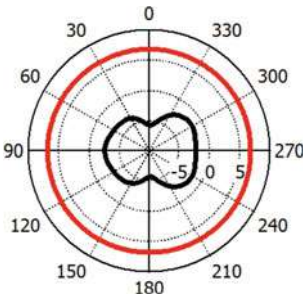


Fig. 23 H-Plane of antenna



4 Data Analysis and Interpretation

Sahar et al. [14] demonstrated a reconfigurable multiband for RFID and GPS use. The construction consists of a C-shaped patch with a dipole antenna loaded on one end and a switch on the other. When the switches are turned off, the resonance frequency

for GPS applications is 1.2275 GHz. The antenna was resonant at 0.89 GHz for the UHF band and 2.45 GHz for the ISM band when the switches were turned on.

Instead of a dipole antenna, the structure was expanded with two C-shaped patches and a T-shaped patch in the current article. The switching function is performed by two PIN diodes. The antenna was resonant at 26.83 GHz for 5G upper band application when both switches were turned off. The antenna was resonant at 5.55 GHz for WLAN use with S1 off and S2 on. The antenna was resonant at 3.36 GHz for 5G lower band application with S1 on and S2 off. The antenna was resonant at 2.3 GHz for 4G band application when both switches were turned on. The structural expansion of the antenna was developed for diverse applications such as 4G, 5G, and WLAN with better radiation performance and meeting the current state-of-the-art technology in wireless communication as a result of this data analysis and interpretation shown in Table. 4.

5 Conclusion

With the aid of two PIN diodes, a multiband reconfigurable antenna for various purposes was presented. The structure is made up of two C-shaped rectangular patches that are loaded with a T-shaped patch that feeds microstrip lines. By switching the PIN diodes, this design was able to reconfigure the antenna, resulting in four different frequencies for future applications: 2.3–2.4 GHz for fourth-generation mobile applications, 3.3–3.4 GHz for fifth-generation lower band applications, 24.25–27.5 GHz for fifth-generation higher band applications, and 5.5–5.6 GHz for WLAN applications. The planned structure was built, and the measured results matched those predicted by the simulation.

Table 4 Proposed structure comparison with study in literature

Reference paper	Structure	Switch condition	Resonant frequency (GHz)	Return loss (dB)	VSWR	Bandwidth (GHz)	Directivity (dBi)	Application of the frequency
Sahar et al. [14]	C-shaped patch loaded with dipole antenna	off	1.2275	−23.14	1.2	0.14	3.14	1.2275 GHz—GPS application
		on	0.89	−18.89	1.4	0.07	1.414	850–930 MHz—UHF band
			2.45	−21.90	1.2	0.13	3.63	2.41–2.54 GHz—ISM band
Proposed Structure	Two C-shaped patches loaded with T-shaped patch antenna	S1 off S2 off	26.83	−34.47	1.03	1.44	13.3	5G upper band
		S1 off S2 on	5.55	−16.11	1.29	0.24	6.14	WLAN
		S1 on S2 off	3.364	−11.11	1.6	0.01	5.38	5G lower band
		S1 on S2 on	2.303	−17.85	1.19	0.08	6.63	4G band

References

1. Mohanta HC, Kouzani AZ, Mandal SK (2019) Reconfigurable antennas and their applications. Univ J Electr Electron Eng 6(4):239–258
2. Shah SAA, Khan MF, Ullah S, Flint JA (2014) Design of a multi-band frequency reconfigurable planar monopole antenna using truncated ground plane for Wi-Fi, WLAN and WiMAX applications. In: 2014 International conference on open source systems and technologies (ICOSST), Dec 2014, pp 151–155
3. Goutham D, Sahay J (2014) A novel multiband frequency reconfigurable antenna for WiMax and Wi-Fi application. In: 2014 International conference on advanced communication control and computing technologies (ICACCCT), May 2014, pp 833–836
4. Boukarkar A, Lin XQ, Jiang Y (2015) A new reconfigurable multi-band monopole antenna for different wireless applications. In: 2015 IEEE international conference on communication software and networks (ICCSN), June 2015, pp 138–142
5. Mun B, Jung C, Park MJ, Lee B (2014) A compact frequency-reconfigurable multiband LTE MIMO antenna for laptop applications. IEEE Antennas Wirel Propag Lett 13:1389–1392
6. Majid HA, Rahim MKA, Hamid MR, Murad NA, Ismail MF (2013) Frequency- reconfigurable microstrip patch-slot antenna. IEEE Antennas Wirel Propag Lett 12:218–220
7. Abdulraheem YI, Oguntala GA, Abdullah AS, Mohammed HJ, Ali RA, Abd-Alhameed RA, Noras JM (2017) Design of frequency reconfigurable multiband compact antenna using two PIN diodes for WLAN/WiMAX applications. IET Microwaves, Antennas & Propagation
8. Ejaz A, Nilavalan R, Abutarboush H (2013) Tunable multiband microstrip antenna for 5GHz WLAN. Glob J Res Eng
9. Yeole DS, Khot UP (2016) Reconfigurable multiband microstrip patch antenna design for wireless communication applications. In: IEEE international conference on recent trends in electronics, information & communication technology (RTEICT), May 2016, pp 633–636
10. Saleem R, Quddus A (2017) Circular slotted reconfigurable multiband patch antenna for wireless applications. In: International applied computational electromagnetics society symposium-Italy (ACES). IEEE, Mar 2017, pp 1–2
11. Kaur P, Asok De SK (2014) Design of a novel reconfigurable fractal antenna for multi-band application
12. Jin G, Deng C, Xu Y, Yang J, Liao S (2020) Differential frequency reconfigurable antenna based on dipoles for sub-6GHz 5G and WLAN applications. IEEE AWPL Antennas Wirel Propag Lett 19:472–476
13. Desai A, Patel R, Upadhyaya T, Kaushal H, Dhasarathan V (2020) Multiband inverted E and U shaped compact antenna for digital broadcasting, wireless, and Sub 6 GHz 5G applications. AEU-Int J Electron Commun 123:153296
14. Sahar NM, Islam MT, Misran N (2015) A reconfigurable multiband antenna for RFID and GPS applications. Elektronika Ir Elektrotechnika 21(6). ISSN 1392-1215

Chapter 47
Optimal Value for Number of Clusters in a Dataset for Clustering Algorithm

Jayashree and T. Shivaprakash

1 Introduction

There is a revolutionary development in the technology of data storage and computing, to manage the large quantity of data from a variety of sources or fields such as medical, social, scientific data, etc. These data contain the most valuable information, and it requires clustering algorithms to form groups or clusters. For conventional clustering algorithm needs prior knowledge [1] of the actual number of clusters. Since clustering is an unsupervised type of operation, it is not easy to detect the number of clusters from large complex high-dimensional datasets [2]. So it is very important and significant to find the number of clusters in the datasets to get an efficient and correct solution of clustering [3] algorithm.

The combination of data mining with machine learning [4, 5], artificial intelligence, deep learning [6, 7], statistics, and the database system [8] provides the techniques for extracting useful information [9, 10] which may be unknown from a large dataset. One of the tasks of data mining [11] is clustering or cluster analysis to extract information from the dataset [12] that is for partition unlabeled data [13] into a number of clusters (groups or categories) [14].

From the aspect of optimization of clustering [15], the main characteristic of clustering is to magnify both the (internal) homogeneity [16] within a cluster and the (external) heterogeneity [17] with different clusters. Specifically, an identical type of data point should belong to the same cluster, while the alternative type of data points should be belonging to different clusters. For measuring similarity between instances in the datasets for clustering tasks, there are a variety of dissimilarity measures [18], and it has got some advantages and disadvantages depending on the type of datasets. For example, some are advisable for text categorization [19],

Jayashree (✉) · T. Shivaprakash
Vijaya Vittala Institute of Technology, Bangalore, Karnataka, India
e-mail: jayashrees08@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_48

while other ones are suitable to image segmentation [20], biomedical imaging [21], document clustering [22], statistical analysis, and network separation [23].

In unsupervised feature extraction with a k-means algorithm [24], fixing the parameter is an open problem to solve. Finding optimal value for cluster number is an important parameter concerning cluster quality and time efficiency [25] of the clustering technique [26, 27]. This paper presents one of the parameters, which is number of cluster value k for clustering algorithm [28] without specifying prior knowledge for k values.

2 Background and Related Work

Hancer and Karaboga [3] introduced a comprehensive survey of different approaches such as traditional, merge-split, and evolutionary methods proposed for the estimation of cluster numbers. The determination of cluster numbers involves three main approaches: traditional, merge-split, and evolutionary computation-based approach [29].

In traditional approaches, all parameters other than cluster number make it fixed and execute clustering algorithm, where there needs to provide a range of numbers for parameter cluster number k. While comparing different traditional approaches, Calinski-Harabasz index was achieved better performance with others. The split and merge approach is based on splitting large, clusters into smaller ones and merge smaller similar clusters considering some user-specific parameters.

Iterative Self-Organizing Data Analysis Technique (ISODATA) [30], Dynamic Optimal Cluster Seeking (DYNOC), maximum variance cluster (MVC) method, Modified Linde-Buzo-Gray (MLBG) methods are few examples of split and merge approach. The evolutionary computation (EC)-based approach provides the solution to find the cluster number starting with some possible number and encode to get some specific and nearly optimal cluster number. There are different encoding schemes like centroid-based encoding, don't care-based encoding, partition-based encoding, etc.

Dinh et al. [31] proposed an algorithm for finding the optimal value for the number of clusters in categorical data clustering by silhouette coefficient. The first step of this algorithm is to find a cluster center followed by find the distance between the center and each element of clusters and then the silhouette method to find the best k.

Slobodan [32] provides a comparison between the silhouette index and the Davies-Bouldin index in Labeling IDS Clusters. Here, IDS stands for the intrusion detection system. IDS uses clustering for determining the good and bad behaviors of the network/host. In that paper, the authors tested and compared the quality of the cluster with two indexes: the Silhouette index and the Davies-Bouldin index, and found that the silhouette index provides a slightly better result than Davies-Bouldin index [33, 34].

2.1 Methods

There are few different methods to calculate index values and by using an index, k value can be estimated. Most of the methods required to point out knee point or elbow point in the graph of index values for optimal value k. Sometimes, this elbow point finding is difficult in some methods, since there is no visible knee point in the graph. Few methods are as follows.

Elbow method Elbow method [35] computes total intra-cluster variation [or total within-cluster sum of square (WSS)] by using common clustering algorithm such as k-means. WSS should be minimum as possible, which indicates compactness of clustering.

Silhouette Coefficient Silhouette coefficient [36, 37] provides the idea about how the objects are classified in cluster output, and it is computed as below formula 1.

$$\text{Silhouette Score} = \frac{(b-a)}{\max(a,b)} \tag{1}$$

where

a is the average within-cluster distance, i.e., the mean of distance between each point within a cluster.
b is the average between-cluster distance, i.e., the mean of distance between all clusters.

Silhouette scores range from -1 to 1, if the is -1 indicates wrong clusters and 1 indicates the number of clusters are well separated, and a higher the value for silhouette coefficient score indicates the model with better-defined clusters. Therefore, the maximum value of the index indicates an optimal solution for k in clustering [38].

Davies-Bouldin Davies-Bouldin (DB) index [39, 40] is based on the finding of the value which represents the quality of partition and that can be computed by estimating the distances between clusters and their dispersions [41]. The DB index [42] can be calculated using formula 2,

$$\text{DB} = \frac{1}{n_c}\sum_{i=1}^{n_c} R_i \tag{2}$$

where

$$R_i = \max_{j=1\ldots nc, i\neq j}(R_{ij}), i = 1\ldots n_c$$

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

$$d_{ij} = d\left(v_i, v_j\right), s_i = \frac{1}{\|c_i\|}\sum_{x\epsilon c_i} d(x, v_i)$$

where

$d(x, y)$ is the euclidean distance between x and y
c_i is cluster i.
v_i is the centroid of cluster c_i
$\|c_i\|$ refers to the norm of c_i

Lower the DB index [43] value, better is the clustering. It also has a drawback [44]. This is dependent both on the data and the algorithm.

2.2 *Calinski-Harabaz Index(CH-Index)*

Calinski-Harabaz index [45] is also called as the variance ratio criterion and it is defined as Eq. 3

$$\mathrm{CH} = \frac{SS_B}{SS_W} * \frac{n-k}{k-1} \tag{3}$$

SS_W is the overall within-cluster variance, which is the same as total with-in sum of squares (Euclidean distance between the two vectors.) $\sum_i^k \sum_{x \in C_i} \parallel x - m_i \parallel^2$ where x is the data point, C_i is the ith cluster, m_i is the centroid of cluster i.

SS_B is the overall between-cluster variance, which can be calculated as the total sum of squares (tss) minus SS_W and(tss) is the squared distance of all the data points from the dataset's centroid. $\sum_i^k n\|m_i - c\|^2$ n is the total number of observations (data points), k is the number of clusters, c is the global centroid.

Advantages include, the score can be computed quickly and the higher value indicates clusters are dense and well separated, which fulfill the standard concept of a cluster.

The limitations of this method are the CH index is generally higher for convex clusters [46] than other non-convex clusters, such as density-based clusters like those obtained through DBSCAN.

Krzanowski and Lai(KL) Krzanowski and Lai (KL) [47–49] index is defined by Eq. 4

$$\mathrm{KL} = \frac{\mathrm{DIFF}_q}{\mathrm{DIFF}_{q+1}} \tag{4}$$

where $\mathrm{DIFF}_q = (q-1)^{\frac{2}{p}} \mathrm{trace}(W_{q-1}) - q^{\frac{2}{p}} \mathrm{trace}(W_q)$

q is the number of clusters
p is the number of variables
W_q is within-group dispersion matrix for data clustered into q clusters

Gap static Gap statistics [50] algorithm adopts the output of any clustering algorithm [51] and compares changes within-cluster variance and expected under some distribution. Gap static can be defined by following Eq. 5

$$\text{Gap}_n(k) = E_n^* \log(W_k) - \log(W_k) \tag{5}$$

where k is the estimated optimal number of clusters,

n is the number of samples
W_k within-cluster sum of squares around the cluster means
E_n^* denotes the expectation under a sample of size n from the reference distribution

Depth difference(DeD) DeD stands for depth difference and its unique way of finding the optimal k value [52]. In this, the depth of each point is calculated by using Mahalanibois depth function. DeD can be defined by using the formula 6

$$\text{DeD} = \text{DW} - \text{DB} \tag{6}$$

where DW is depth within cluster and DB is depth between cluster.

Curve method Basically, the curvature method is as shown in Algorithm 1, to determine the knee point in the evaluation graph, and with an aspect of mathematics, curvature [53] is the amount by which a geometric object deviates from being flat or straight in the case of a line. So the point with the maximum curvature in the graph represents the knee point. The within-cluster variance evaluation graph can be calculated by using following formula 7

$$J(k) = \Sigma_{j=1}^{k} \Sigma_{x_i \epsilon C_j} ||x_i - \overline{x_j}||^2 \tag{7}$$

where C_j is the set of class j samples and $\overline{x_j}$ is the sample mean of class j. For a curve explicitly given as $y = f(x)$, the curvature is computed using the formula 8

$$K = \frac{|y^{"}|}{(1 + y^{'2})^{3/2}} \tag{8}$$

3 Optimal Cluster Number Estimation Algorithm

The proposed optimal cluster number estimation algorithm (OCNE) is as shown in Algorithm 2; the most common clustering algorithm k-means is used to find the within-cluster variance as the evaluation metric to form the graph for evaluation purposes. Let $X = x_1, x_2 \ldots, x_n$ each x_i, $i = 1, 2, \ldots, n$ is the data points of dataset X with n instances and within-cluster variance is defined by using Euclidean distance measure as shown in the formula

$$W = \Sigma_{j=1}^{k} \Sigma_{x_i \varepsilon C_j} ||x_i - \overline{x_j}||^2 \tag{9}$$

where W is total with-in cluster variance, C_j is class j samples, k is the number of clusters that starts from value one, $\overline{x_j}$ is the mean of samples belong to j class.

Algorithm 1 curvature method

Require: dataset X having objects as $x_1, x_2, x_3....x_n$ and k_max
Ensure: index d and k values
  for k=1 to $k_{max} + 1$ **do**
    for t=1 to 20 **do**
      Compute the within-cluster variance using k-means algorithm
    end for
    Get minimum of within-cluster variance
    if $k > 1$ **then**
      Compute the differences in within-cluster variance
    end if
  end for
  for k=2 to k_{max} **do**
    Compute the Curvature index
  end for
  The optimal number k is maximum value of Curvature index
  return k

The next main step in the algorithm is finding the validation index which can be calculated by finding the mean of difference in with-in cluster variance, which is defined as

$$VI(k) = \frac{1}{k} \Sigma_{i=1}^{k} \text{delta}_k \tag{10}$$

where delta_k is the difference in with-in cluster variance. The algorithm stops execution when it reaches the condition $VI(k) < VI(k-1)$ and returns $k-1$ as an optimal k value.

Algorithm 2 Proposed method

Require: dataset X having objects as $x_1, x_2, x_3....x_n$
Ensure: k value
  k=1
  Initialize the validity index VI(k) to value zero
  Initialize scores and delta with value 0
  Do
    k=k+1
    Compute the within-cluster variance using k-means algorithm
    Get minimum of within-cluster variance
    Compute the differences in within-cluster variance and its $delta_k$
    $VI(k) = \frac{1}{k} \Sigma_{i=1}^{k} delta_k$
  Repeat the step from Do untill $VI(k) < VI(k-1)$
  return k-1

3.1 Complexity

The time complexity of the proposed algorithm(optimal cluster number estimation algorithm) is depend on two factors: the complexity of calculating with-in cluster variance which is nothing but k-means algorithm and complexity of computing validation index. The time complexity [54] of k-means algorithm is $O(n^2)$. The time complexity of validation index [55] is the complexity of finding $delta_k$ which is $O(n * \log(n))$ where n is the size of dataset. Finally, the time complexity of the proposed algorithm is $O(n^2) + O(n * \log(n))$ where n is the size of dataset.

Similarly, space complexity depends on k-means algorithm and validation index computation. The space complexity of validation index computation is $O(n)$ where n is the size of dataset. The space complexity of k-means algorithm is $O((m + k) * n)$ where m is the number of objects(samples) and n is the number of attributes considering n-dimensional objects. Therefore, space complexity become $O((m + k)n) + O(n)$.

4 Experiment

Experiments consist of the dataset, different algorithms for comparison such as elbow method, silhouette method, CH index, DB index method, and validation metrics such as relative error measure and k-value comparison table.

4.1 Experimental Settings

Datasets The datasets with the number of variables or attributes, number of samples or instances, and clusters are as shown in Table 1. The datasets include real-world datasets as well as synthetic datasets. The structure of few datasets is shown in Fig. 1.
Evaluation metrics The relative error(RE) for computed k number of clusters is by using the formula 11

$$\text{RE} = \frac{|k_{\text{act}} - k_{\text{est}}|}{k_{\text{act}}} \tag{11}$$

where k_{est} is the number of clusters that is computed and k_{act} is the actual number of clusters in the known dataset. Table 3 shows the relative error of optimal k value calculated with different algorithms. There is the average highest relative error in the experiment is gap static followed with DeD algorithm. The lowest relative error is DB algorithm followed by CH algorithm. The proposed algorithm(optimal cluster number estimation algorithm) provides a moderate level of relative error. It is observed that the algorithm does not depend on the number of instances or a size of

Table 1 List of datasets and its characteristics

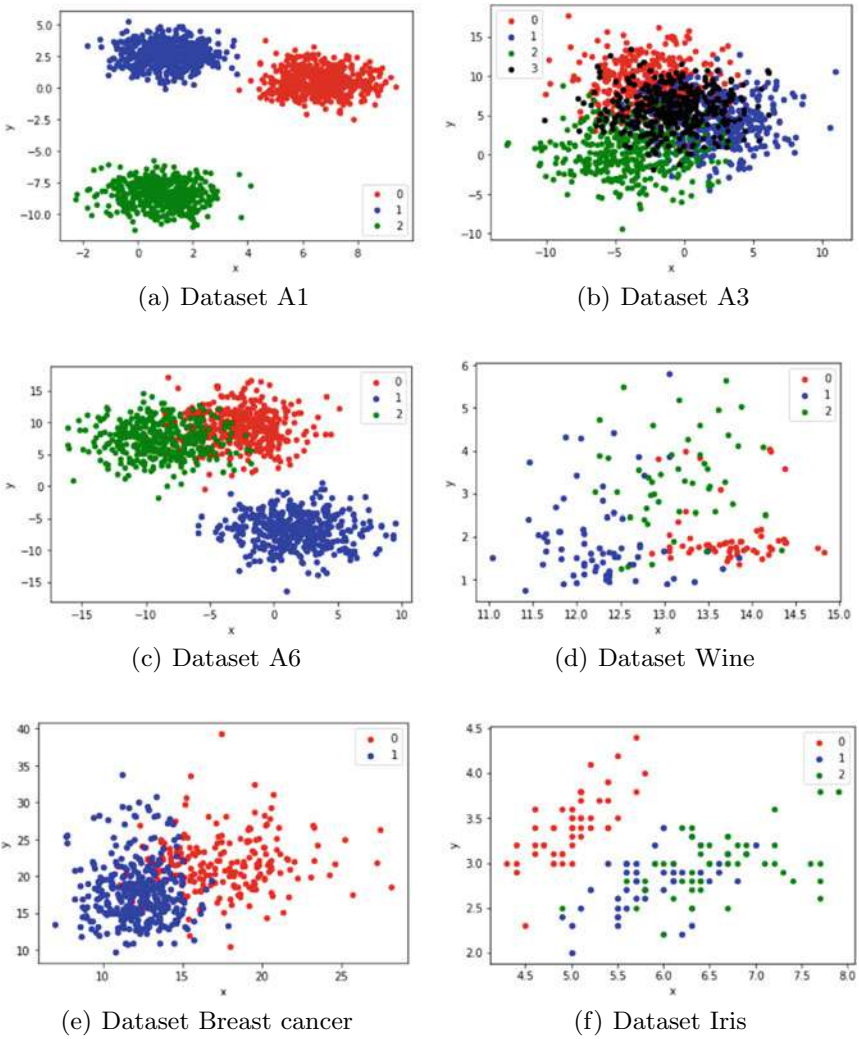
Dataset	No of instances	No of attributes	No of clusters
Wine (WN)	178	13	3
Breast-cancer	286	10	2
Breast-tissue	106	10	4
Ionosphere	351	34	2
Parkinsons	195	23	2
Seeds	210	8	3
Pima	768	9	2
Hill	1212	101	2
Haberman	306	4	2
Heart-va	200	13	2
Musk	476	167	2
Cardiotocography	476	167	10
Wine-quality-red	1599	12	6
Transfusion	748	5	2
Soybean	683	36	19
Gas-drift	13,910	130	6
Iris	150	5	3
A1	1500	2	3
A2	1500	4	3
A3	1500	8	4
A4	1500	8	2
A5	1500	8	3
A6	1200	3	3

dataset, since it is providing approximately correct k values to the dataset which is having a large number of instances such as gas-drift or wine-quality-red dataset.

4.2 Experimental Results

The results are shown in Table 2. It consists of different datasets like wine, breast-cancer, ionosphere, etc., and k_known in the Table is known k values for each dataset. Also, it contains k values of the proposed algorithm for each dataset along with different algorithm k values for comparison purposes.

The proposed method (optimal cluster number estimation algorithm) achieved to provide better k values for cluster number than the other existing algorithms. In the experiment with different datasets, good results which are getting followed with proposed algorithm are curve method, DB method and gap static methods. The



(a) Dataset A1 (b) Dataset A3

(c) Dataset A6 (d) Dataset Wine

(e) Dataset Breast cancer (f) Dataset Iris

Fig. 1 Structure of the different datasets

experiment with real-world datasets such as wine, hill, seeds, transfusion, and iris provides the same k values as known k values.

It is observed in the experiment that the k values getting as result are nearly equal to the known k value or the k values are near to the known k value, in the case of datasets where k values are not the same as known k value. In the experiment, proposed algorithm on cardiotocography dataset provides k value as 7 which is near to the known k value 10. Similarly proposed algorithm with breast-tissue and haberman datasets provide k values as 3 which is near to the value 4 and 2, respectively. In the case of the synthetic dataset also, similar kinds of results were observed, such

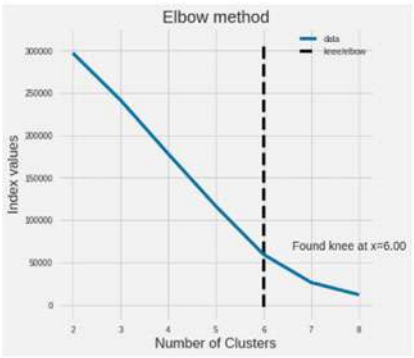
Table 2 Experimental comparison for k values of proposed method OCNE with other different methods on different datasets

Dataset	k_known	CH	DB	SIL	Gap	DeD	Curve	OCNE
Wine(WN)	3	2	2	3	14	2	2	3
Breast-cancer	2	5	3	3	14	3	3	6
Breast-tissue	4	3	4	3	13	2	2	3
Ionosphere	2	3	2	2	14	6	12	7
Parkinsons	2	2	2	3	14	11	3	4
Seeds	3	2	3	2	13	3	3	3
Pima	2	2	4	2	13	19	3	5
Hill	2	2	2	3	14	2	2	2
Haberman	2	2	4	2	7	2	5	3
Heart-va	2	5	2	3	13	4	2	6
Musk	2	5	2	2	14	2	3	7
Cardiotocography	10	5	2	2	14	2	3	7
Wine-quality-red	6	3	4	3	14	2	7	5
Transfusion	2	2	3	4	14	13	4	2
Soybean	19	3	2	3	14	3	2	5
Gas-drift	6	4	2	4	14	2	2	3
Iris	3	4	2	5	14	2	2	3
A1	3	2	2	6	3	5	3	2
A2	3	2	2	2	3	2	3	3
A3	4	2	2	2	4	2	4	3
A4	2	4	2	4	2	2	2	8
A5	3	2	2	2	3	2	3	3
A6	3	4	2	5	3	7	2	3

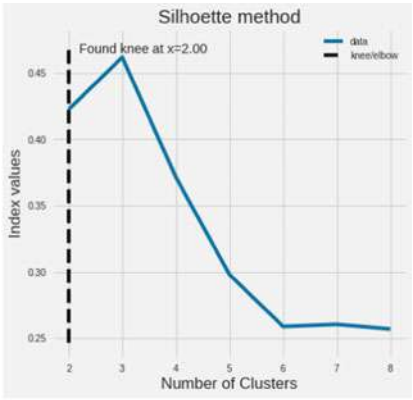
as three datasets out of six provide the same k values as known k values and two datasets provides k values near to known k values.

Optimal cluster numbers of clusters with the graph using few different existing algorithms with seeds dataset are as shown in Fig. 2. Here, the x-axis indicates k values and the y-axis indicates indexes of respective algorithms on seeds dataset. In the proposed OCNE algorithm, the estimated k value is 3 which is shown in Fig. 2e. As per the algorithm, $k = 3$ is an optimal valve, and after that, algorithm does not iterate with the next k values and returns the k value. This signifies that validation index decides the execution of iterations in the algorithm. Here, elbow point or knee points are not considered such as other existing algorithms.

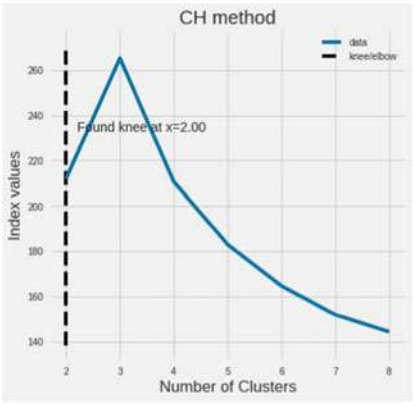
There are other four different algorithms as shown Fig. 2a–d which uses knee point or elbow point to find optimal k value. These algorithms find indexes for each k value and then the user has to find optimal value by finding elbow point. In some datasets, it is a challenge to find knee points in the graph Table 3.



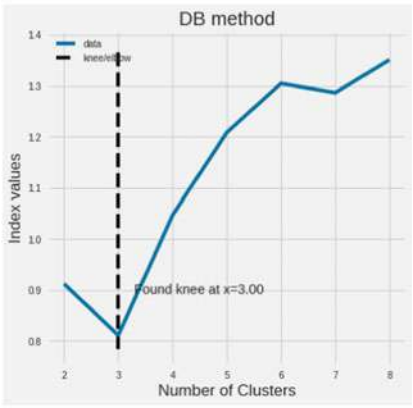
(a) Elbow method



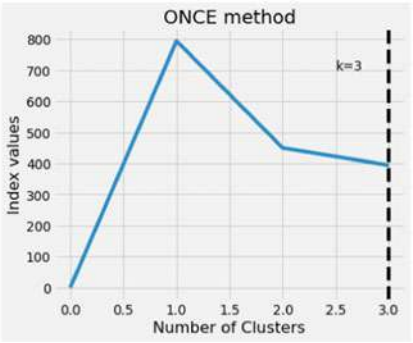
(b) Silhouette method



(c) CALINSKI-HARABAZ method



(d) DAVIES-BOULDIN method



(e) OCNE method

Fig. 2 Different algorithms to find optimal value for number of clusters

Table 3 List of relative errors for k value with respect to a real number of clusters

Dataset	CH	DB	SIL	Gap	DeD	Curve	OCNE
Wine(WN)	0.33	0.33	0	3.66	0.33	0.33	0
Breast-cancer	1.5	0.5	0.5	6	0.5	0.5	2
Breast-tissue	0.5	1	0.5	6	0	0	0.5
Ionosphere	0.5	0	0	6	2	5	2.5
Parkinsons	0	0	0.5	6	4.5	0.5	1
Seeds	0.33	0	0.33	3.33	0	0	0
Pima	0	1	0	5.5	8.5	0.5	1.5
Hill	0	0	0.5	6	0	0	0
Haberman	0	1	0	2.5	0	1.5	0.5
Heart-va	1.5	0	1.5	6	1	0	2
Musk	1.5	0	0	6	0	0.5	2.5
Cardiotocography	0.5	0.8	0.8	0.4	0.8	0.7	0.3
Wine-quality-red	0.5	0.33	0.5	1.33	0.66	0.16	0.16
Transfusion	0	0.5	1	6	5.5	1	0
Soybean	0.84	0.89	0.84	0.26	0.84	0.89	0.73
Gas-drift	0.33	0.66	0.33	1.33	0.66	0.66	0.5
Iris	0.33	0.33	0.66	3.66	0.33	0.33	0
A1	0.33	0.33	1.0	0.0	0.66	0.0	0.33
A2	0.33	0.33	0.33	0.0	0.33	0.0	0.0
A3	0.33	0.33	0.33	0.0	0.33	0.33	0.0
A4	1.0	0.0	1.0	0.0	0.0	0.0	3.0
A5	0.33	0.33	0.33	0.0	0.33	0.0	0.0
A6	0.33	0.33	0.66	0.0	1.33	0.33	0.0

5 Conclusion

The paper presents a novel approach, the optimal cluster number estimation algorithm, to find the optimal value for k number of clusters. This method selects the k value for a number of clusters of a dataset by executing the k-means clustering algorithm over a dataset to compute within-cluster variance and returns the k value on satisfying validation index. This method does not require an extra input parameter, unlike other existing algorithms. The proposed method provides efficient results on finding k value on different real and synthetic datasets. On a comparative study on different datasets on finding k value in the experiment, the proposed algorithm outperforms the other existing algorithms. From the point of view of relative error on k value computation, the proposed algorithm provides moderately good results. In the experiment, it is found that the algorithm does not depend on the number of cluster in dataset or the data size, on computing k value.

In the case of a complex dataset, it is difficult to select the correct or nearly correct value of k for clustering data. The different algorithms provide different values for the same data set on each execution. It is a hard problem to the calculate number of clusters k, because of unknown labels or features in a dataset. Although this method approximately provides good results on different known dataset on executions. The limitation with the algorithm is time efficiency, and the relative error is quite high for some datasets. Future work includes finding efficiency with respect to time and experimenting on different types of datasets. This method uses Euclidean distance to measure the distance between data points and k-means algorithm for clustering; future work can also include other distance measures and other clustering algorithms to get efficient results.

Acknowledgements This material is based upon work supported by the Google Cloud Research Credits program with the award GCP19980904

References

1. Simoncelli EP (2009) Optimal estimation in sensory systems. Cognitive Neurosci IV, 525–535
2. Wang H, Song M (2011) Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. R J 3(2):29
3. Hancer E, Karaboga D (2017) A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. Swarm Evol Comput 32:49–67
4. Angra S, Ahuja S (2017) Machine learning and its applications: a review. In: 2017 International conference on big data analytics and computational intelligence (ICBDAC). IEEE, pp 57–60
5. Louridas P, Ebert C (2013) Embedded analytics and statistics for big data. IEEE Softw 30(6):33–39
6. Shaikh F, Dar P, Srivastava P (2017) Comparison between deep learning and machine learning. https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/
7. Devarajan S (2017) A quick introduction to ai http://www.iamwire.com/2017/07/quick-introduction-ai/155885
8. What is deep learning? how it works, techniques and applications https://in.mathworks.com/discovery/deep-learning.html
9. Artificial intelligence what it is and why it matters https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html
10. TAC (2017) Artificial intelligence hacker noon. https://hackernoon.com/understanding-understanding-an-intro-to-artificial-intelligence-be76c5ec4d2e
11. Gheware S, Kejkar A, Tondare S (2014) Data mining: task, tools, techniques and applications. Int J Adv Res Comput Commun Eng 3(10)
12. Kameshwaran K, Malarvizhi K (2014) Survey on clustering techniques in data mining. Int J Comput Sci Inf Technol 5(2):2272–2276
13. Matatov N, Rokach L, Maimon O (2010) Privacy-preserving data mining: a feature set partitioning approach. Inf Sci 180(14):2696–2720
14. Shen J, Chang SI, Lee ES, Deng Y, Brown SJ (2005) Determination of cluster number in clustering microarray data. Appl Math Comput 169(2):1172–1185
15. Teboulle M (2007) A unified continuous optimization framework for center-based clustering methods. J Mach Learn Res 8(1)
16. Liu Q, Liu W, Tang J, Deng M, Liu Y (2019) Two-stage permutation tests for determining homogeneity within a spatial cluster. Int J Geograph Inf Sci 33(9):1718–1738

17. Zheng S, Hume KA, Able H, Bishop SL, Boyd BA (2020) Exploring developmental and behavioral heterogeneity among preschoolers with ASD: a cluster analysis on principal components. Autism Res 13(5):796–809
18. Shirkhorshidi AS, Aghabozorgi S, Wah TY (2015) A comparison study on similarity and dissimilarity measures in clustering continuous data. PloS One 10(12):e0144059
19. Bekkerman R, El-Yaniv R, Tishby N, Winter Y (2001) On feature distributional clustering for text categorization. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp 146–153
20. Naz S, Majeed H, Irshad H (2010) Image segmentation using fuzzy clustering: a survey. In: 6th international conference on emerging technologies (ICET). IEEE, pp 181–186
21. Xu R, Wunsch DC (2010) Clustering algorithms in biomedical research: a review. IEEE Rev Biomed Eng 3:120–154
22. Zamir O, Etzioni O (1998) Web document clustering: a feasibility demonstration. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, pp 46–54
23. Bandyopadhyay S, Maulik U (2002) An evolutionary technique based on k-means algorithm for optimal clustering in RN. Inf Sci 146(1–4):221–237
24. Suhui W, Ying C, Yanning Z, Yuntao P (2011) Survey on k-means algorithm. Data Anal Knowl Disc 27(5):28–35
25. Hruschka ER, Campello RJ, de Castro LN (2004) Improving the efficiency of a clustering genetic algorithm. In: Ibero-American conference on artificial intelligence. Springer, pp 861–870
26. Shafeeq A, Hareesha K (2012) Dynamic clustering of data with modified k-means algorithm. In: Proceedings of the 2012 conference on information and computer networks pp 221–225
27. Rosenberger C, Chehdi K (2000) Unsupervised clustering method with optimal estimation of the number of clusters: application to image segmentation. In: Proceedings 15th international conference on pattern recognition. ICPR-2000, vol 1. IEEE, pp 656–659
28. Patil NM, Patil DV (2016) A survey on k-means based consensus clustering. Int J Eng Trends Technol 1(3)
29. Aliniya Z, Mirroshandel SA (2019) A novel combinatorial merge-split approach for automatic clustering using imperialist competitive algorithm. Expert Syst Appl 117:243–266
30. Abbas AW, Minallh N, Ahmad N, Abid SAR, Khan MAA (2016) K-means and isodata clustering algorithms for landcover classification using remote sensing. Sindh University Res J-SURJ (Sci Ser), 48(2)
31. Dinh D-T, Fujinami T, Huynh V-N (2019) Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient. In: International symposium on knowledge and systems sciences. Springer, pp 1–17
32. Petrovic S (2006) A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. In: Proceedings of the 11th nordic workshop of secure IT systems. Citeseer, pp 53–64
33. Hamdi A, Monmarché N, Slimane M, Alimi AM (2016) Intelligent hybrid algorithm for unsupervised data clustering problem. In: International conference on hybrid intelligent systems. Springer, pp 442–455
34. Kassambara A (2017) Practical guide to cluster analysis in R: Unsupervised machine learning, vol 1. Sthda
35. Syakur M, Khotimah B, Rochman E, Satoto B (2018) Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In: IOP conference series: materials science and engineering, vol 336, no 1. IOP Publishing, p 012017
36. Zhou HB, Gao JT (2014) Automatic method for determining cluster number based on silhouette coefficient. In: Advanced research on intelligent system, mechanical design engineering and information engineering III, ser. Adv Mater Res vol 951. Trans Tech Publications Ltd., pp 227–230
37. Xia S-X, Li W-C, Zhou Y, Zhang L, Niu Q (2007) Improved k-means clustering algorithm. J Southeast Univ (English Edition) 23(3):435–438

38. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
39. Thomas JCR, Peñas MS, Mora M (2013) New version of davies-bouldin index for clustering validation based on cylindrical distance. In: 32nd International conference of the Chilean Computer Science Society (SCCC). IEEE, pp 49–53
40. Coelho GP, Barbante CC, Boccato L, Attux RR, Oliveira JR, Von Zuben FJ (2012) Automatic feature selection for BCI: an analysis using the davies-bouldin index and extreme learning machines. In: The international joint conference on neural networks (IJCNN). IEEE, pp 1–8
41. Leskovec J, Rajaraman A (2010) Clustering algorithms, CS345a: data mining. Standford University, United State
42. Xiao J, Lu J, Li X (2017) Davies bouldin index based hierarchical initialization k-means. Intell Data Anal 21(6):1327–1338
43. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 2:224–227
44. Gomonov K, Ratner S, Lazanyuk I, Revinova S (2021) Clustering of EU countries by the level of circular economy: an object-oriented approach. Sustainability 13(13):7158
45. Wang Y, Liu Y, Li L, Infield D, Han S (2018) Short-term wind power forecasting based on clustering pre-calculated CFD method. Energies 11(4):854
46. Chen GK, Chi EC, Ranola JMO, Lange K (2015) Convex clustering: an attractive alternative to hierarchical clustering. PLoS Comput Biol 11(5):e1004228
47. Novianti P, Setyorini D, Rafflesia U (2017) K-means cluster analysis in earthquake epicenter clustering. Int J Adv Intell Inf 3(2):81–89
48. Pramono MS, Linuwih S, Lestari W (2014) Hartigan, krzanowski-lai, silhouette, gap statistic's estimate cluster number (k) methods and the implementation on health's sector. IEESE Int J Sci Technol 3(4):1
49. Charrad M, Ghazzali N, Boiteau V, Niknafs A, Charrad MM (2014) Package 'nbclust. J Statist Softw 61:1–36
50. Yan M, Ye K (2007) Determining the number of clusters using the weighted gap statistic. Biometrics 63(4):1031–1037
51. Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. J Roy Statist Soc: Ser B (Statist Methodol) 63(2):411–423
52. Patil C, Baidari I (2019) Estimating the optimal number of clusters k in a dataset using data depth. Data Sci Eng 4(2):132–140
53. Zhang Y, Mańdziuk J, Quek CH, Goh BW (2017) Curvature-based method for determining the number of clusters. Inf Sci 415:414–428
54. Pakhira MK (2014) A linear time-complexity k-means algorithm using cluster shifting. In: International conference on computational intelligence and communication networks, pp 1047–1051 (2014)
55. Arvidsson J (2019) Finding delta difference in large data sets

Chapter 48
Implementation of a New Hybrid Boost Converter for High Gain Enhancement of SOFC System

CH. Siva Kumar and G. Mallesham

1 Introduction

A modern decentralized electrical power system as shown in Fig. 1 consists of conventional energy sources, energy storage systems, different non-conventional energy sources, residential loads, commercial loads, industrial loads, electrical transportation systems road and rail. The large-scale installations of renewable energy sources across the globe to the grid and distribution systems need to meet the peak demands and reduce the utilization of conventional energy sources. In the recent years, the generation of electricity from the renewable energy resources increased a lot. The total installed generating capacity of India as on 31st May 2021 of 383.37 GW out of which 95.7 GW of renewable energy systems contribution and it is expecting to reach 227 GW by 2022 [1, 2]. Besides wind and solar PV, there are certain other non-conventional methods like fuel cells, biomass energy, geothermal energy, tidal energy, etc. [3–6]. Each technique of control circuit has its own advantages and disadvantages. To achieve the higher efficiencies with pulsating currents and poor regulation switched-capacitor-based DC–DC converter place a major role.

The salient features of fuel cells due to increase in advancements and the salient features like efficiency, silent in operation increasing the importance in generation of electrical energy [7]. The fuel cells are classified based on the state of the electrolyte used, temperature, and the power capacity, etc. The most common type of high-temperature fuel cells are molten carbonate fuel cell; solid oxide fuel cell(SOFC). The SOFC operating temperature is 600–1000 °C. Extraction of maximum energy

CH. Siva Kumar (✉) · G. Mallesham
Department of Electrical Engineering, University College of Engineering, Osmania University, Hyderabad, Telangana 500007, India
e-mail: sivakumar.ch@uceou.edu

G. Mallesham
e-mail: drgm@osmania.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_49

Fig. 1 A modern decentralized electrical power system

and utilization of energy from the renewable energy system's changes the fate of the power sector [8, 9].

The high gain DC/DC converters playing a major role in for the development of renewable energy systems. Among them, switched-capacitor structure [10, 11], a dual-switch coupled inductor-based high step-up DC-DC converter is one of the feasible solutions to obtain the high gains [12]. In [13] Allehyani and Ahmed analysed a transformer less single switch high gain DC–DC converter for renewable energy systems. Non-isolated interleaved DC–DC boost converters with voltage multiplier cells is presented [14]. Inspired from all the typologies, a new hybrid boost converter (HBC) incorporating a single operating switch and an inductor element with the help of a bipolar voltage multiplier (BVM) [15].

The complete paper is organized as: modelling of solid oxide fuel cell and fundamental aspects of modelling of hybrid boosting converter are presented in Sects. 2 and 3, respectively. Simulation studies and the final conclusion presented in Sects. 4 and 5, respectively.

2 Renewable Energy Sources Modelling

Simulation packages gives a handy platform to study the performance of the renewable energy sources under different operating conditions, loading conditions before installing in real time [16]. In this work, a simulation model of solid oxide fuel cell is built along with the HBC boost converter in matlab environment.

2.1 Fuel Cell System: Solid Oxide Fuel Cell

Fuel cell technology: greener, clean, and future technology is a alternative for fossil fuels are the future energy suppliers for electrical vehicle charging stations [17]. The electrolyte used in SOFC: Yttrium-stabilized zircon dioxide [18]. Solid oxide fuel cell is often used wide range of applications due to its higher efficiencies at lower operating costs. The first and the smallest bloom energy server produced is the SOFC. From constructional point of view, it is similar to a battery consisting of anode, cathode, and electrolyte [19], and the respective chemical reactions are shown below at cathode:

$$O_2 + 4e^- \rightarrow 2O^{2-} \tag{1}$$

at anode:

$$H_2 + O^{2-} \rightarrow H_2O + 2e^- \tag{2}$$

or

$$CO + O^{2-} \rightarrow CO_2 + 2e^- \tag{3}$$

overall chemical equation:

$$H_2 + O_2 + CO \rightarrow H_{2O} + CO_2 \tag{4}$$

Modelled the SOFC with an assumption: (i) with constant fuel cell operating temperature (ii) ideal fuel cell gasses and the applicability of (iii) Nernst's equation. The dynamic model of SOFC is shown in Fig. 2. The details of SOFC specifications are given in Table 1.

3 Hybrid Boost Converter

The salient features of the hybrid boost converter (HBC): boost structure regulation and gain enhancement due to its arrange of elements [15]. The internal property of

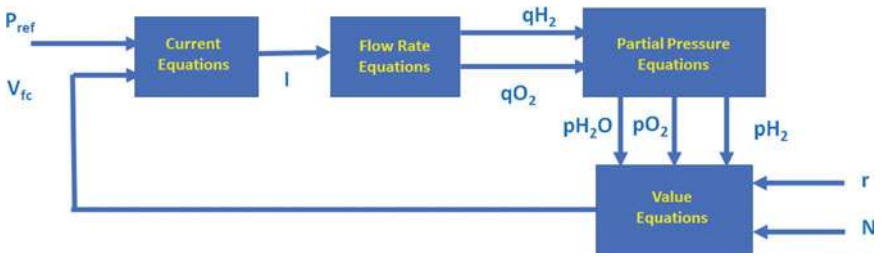


Fig. 2 SOFC dynamic model

Table 1 Details of SOFC specifications

SOFC rated power	100 kW
Absolute temperature	1273 K
Universal gas constant	8314 J/(kmol K)
Ideal standard potential	1.18 V
Number of cells in series in the stack	40
Constant, K_r=N0/4F	0.996×10^{-6} kmol/(s A)
Maximum fuel utilization	0.9
Minimum fuel utilization	0.8
Optimal fuel utilization	0.85
Valve molar constant for water	2.81×10^4 kmol/(s atm)
Valve molar constant for hydrogen	8.43×10^4 kmol/(s atm)
Power factor	1.0
Electrical response time	0.8 s
Fuel processor response time	5 s
Valve molar constant for water	2.81×10^4 kmol/(s atm)
Valve molar constant for hydrogen	8.43×10^4 kmol/(s atm)
Response time for hydrogen flow	26.1 s

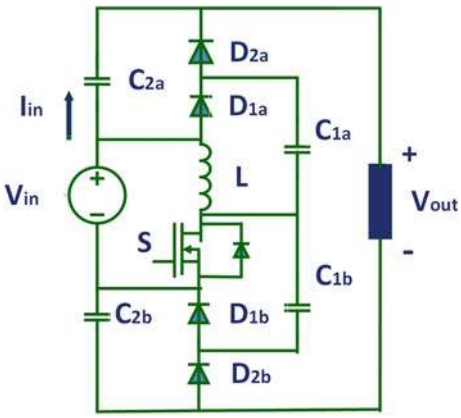


Fig. 3 Second order HBC

the bipolarness for voltage multiples with symmetrical configuration with a inductor element and a operation single switch. The second-order HBC is shown as Fig. 3.

The details of the HBC circuit are shown in Fig. 3. It has two versions of HBC: (i) oddorder HBC (ii) evenorder HBC as shown in Figs. 4 and 5. The higher component utilization is achieved with the even order topology with the help of input source as part of the output voltage of the circuit. Though, both the even and odd order HBCs possesses the similarity in the characteristics and the analysis of the circuit topology and method used. In this work, even order topology has been considered for the work.

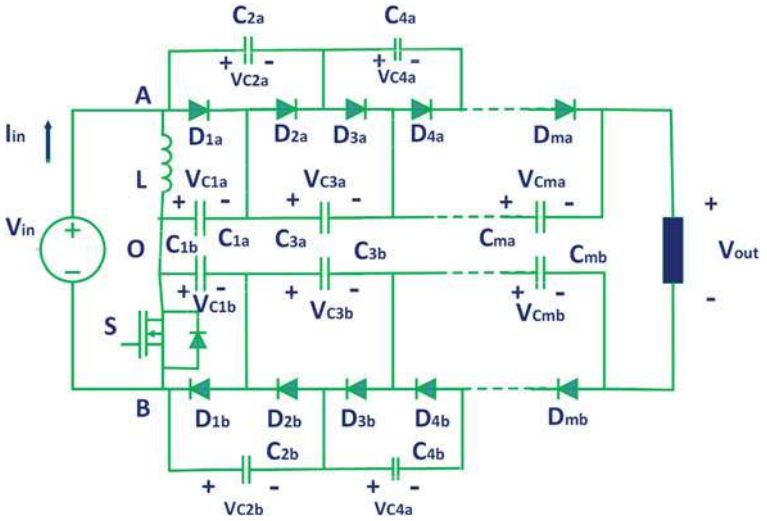


Fig. 4 Odd order HBC

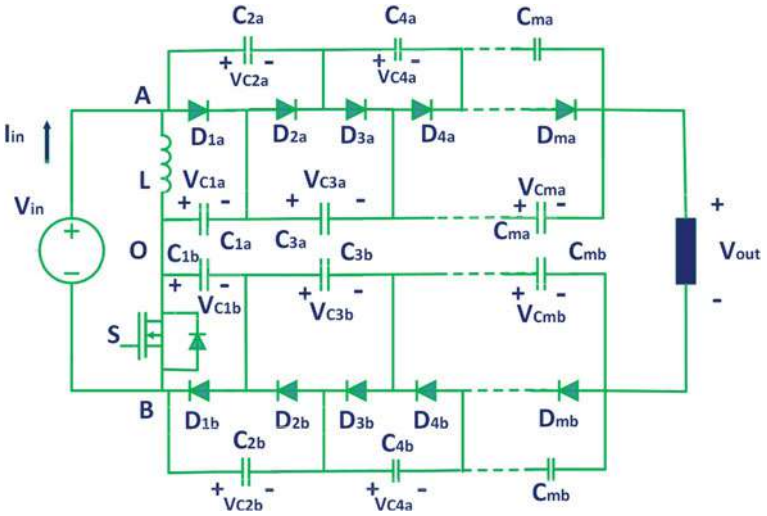
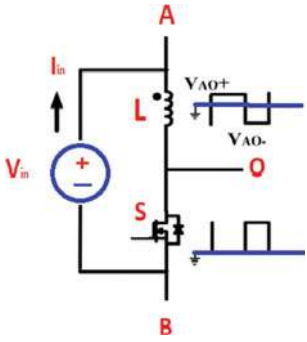


Fig. 5 Even order HBC

Fig. 6 Inductive switch

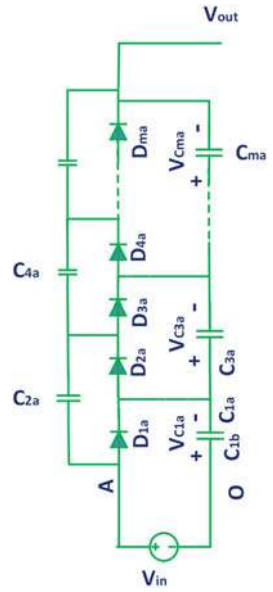


3.1 Inductive Switching Core

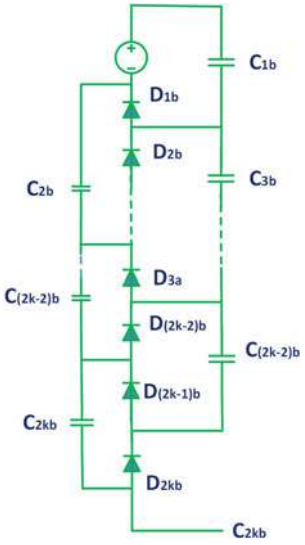
Inductor is used a energy storage system in all the switching patterns/operations in the operation of the boost converters with the condition of releasing the energy to load when it is turns off. The fundamental electrical element used for switching the regulators is an inductor to store the electrical energy in the form of magnetic energy. When the circuit configuration is changed to the on condition, the inductor stores the energy in the form of magnetic energy. When the configuration of the switch is made to off position, the stored energy is fed to the connected load. In a HBC topology, it is the combination of an inductor a switch and: an input source shown as Fig. 6. The complimentary electrical outputs of the configuration are obtained from the ports: AO and OB of the circuit. It is observed the gap between the two level of high voltage and the low voltage of the configurations. But, due to its gap between the configurations, it leads to interleaving operation of the system.

3.2 A Bipolar Voltage Multiplier

The realization of a bipolar voltage is done with the help of the combination of a positive and a negative multiplier branches shown in Fig. 7a and b. Positive multiplier circuit configuration is similar to that of the traditional voltage multiplier. The input to the configuration of the negative multiplier circuit is the point of the input of the cathode terminals of the diodes connected in cascaded mode to generate the negative voltage at the terminals of the anode.



(a) Positive Multiplier



(b) Negative Multiplier

Fig. 7 Multipliers

4 Simulation of Two Level Hybrid Boost Converter: Theoretical Model

The two level HBC is modelled which is shown in Fig. 8. Value of each capacitor used is 100e−6 F. The inductor used in inductive switching core is of value 120e−6 H. The switching frequency is 40 KHz. The input DC voltage is 35V. The simulation is carried out for 0.2 s. The expected output voltage according to equations is 315 V. The obtained voltage is 314.2 V. The graph obtained for this model is shown in Fig. 9 which shows voltage across the inductor; it shows the charging and discharging of the inductor. Figure 10 shows the voltage across the IGBT switch. Figure 11 shows the voltage across the capacitor C3a. The boosted output voltage is shown in the graph of Fig. 12.

4.1 Equivalent Model of Two Level HBC

The equivalent HBC is modelled which is shown in Fig. 13. Value of each capacitors C1a and C1b is (100e−6)/3 F , C2a and C2b is (100e−6)/4 F . The inductor used in inductive switching core is of value 500e−6 H. The switching frequency is 40 KHz. The input DC voltage is 35 V. The simulation is carried out for 0.2 s. The expected output voltage according to equations is 315 V. The obtained voltage is 314.9 V. The output voltage graph obtained for this model is shown in Fig. 14.

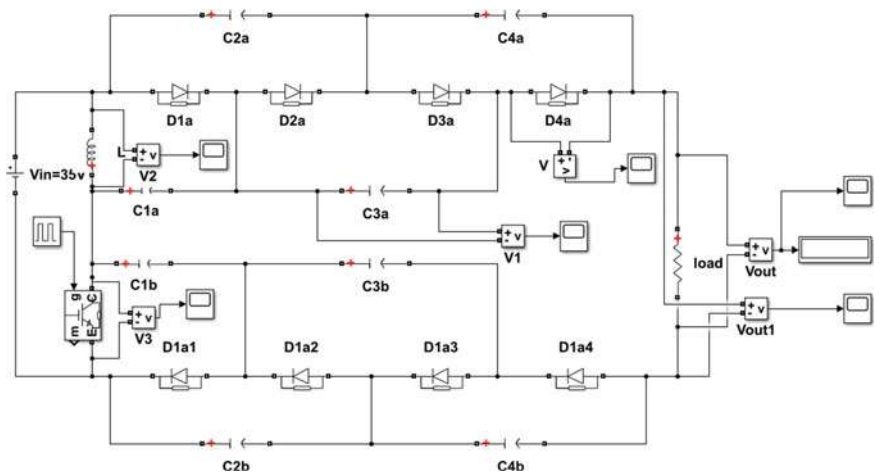


Fig. 8 HBC theoretical model

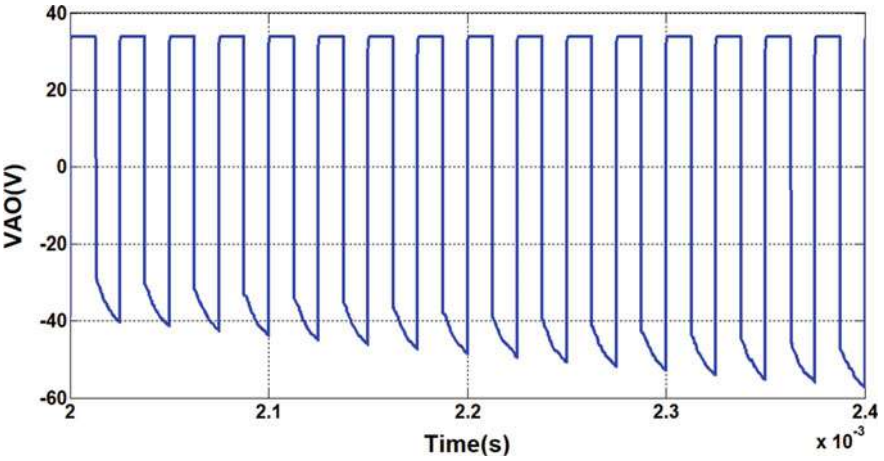


Fig. 9 Voltage across inductor

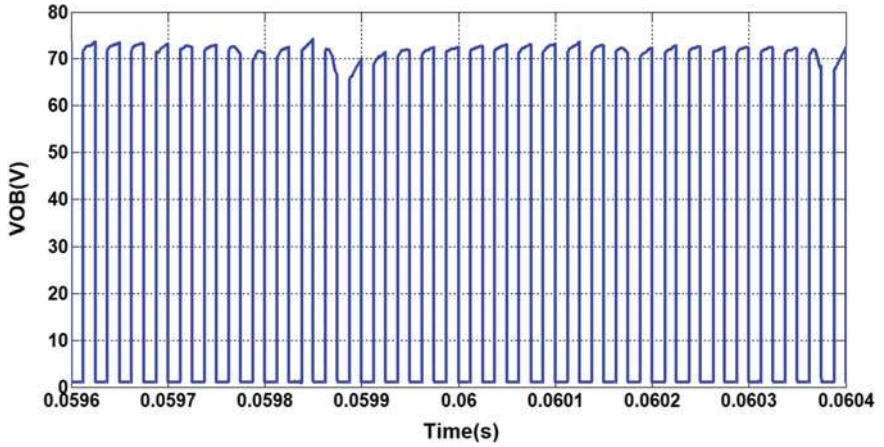


Fig. 10 Voltage across switch (IGBT)

4.2 Simulation of SOFC Connected to Conventional Boost Converter

The SOFC fuel inputs are shown in Fig. 15. SOFC connected to normal boost converter is modelled which is shown in Fig. 16 and shown the SOFC with conventional boost converter. Value of each capacitor used is 100e−6 F. The inductor used in inductive switching core is of value 500e−6 H. The switching frequency is 40 KHz. The output DC voltage of SOFC is 35.18 V; it is given as a input to boost converter. The simulation is carried out for 0.2 s. The obtained voltage is 70.96 V is shown in Fig. 17.

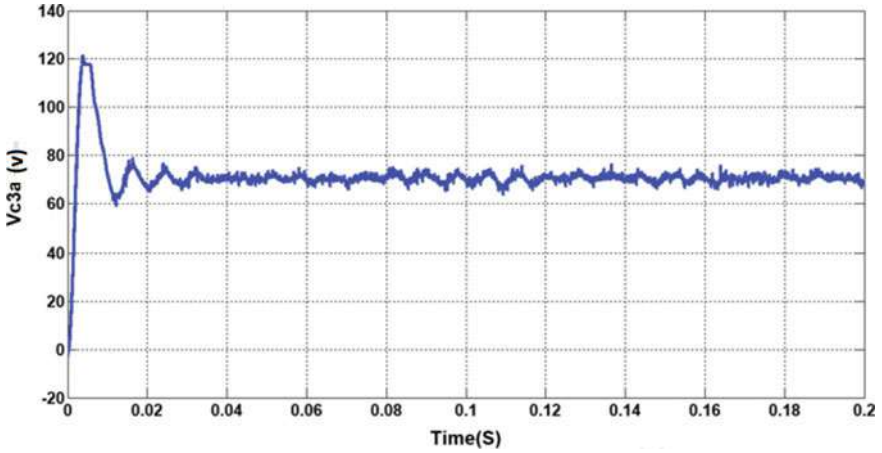


Fig. 11 Voltage across capacitor

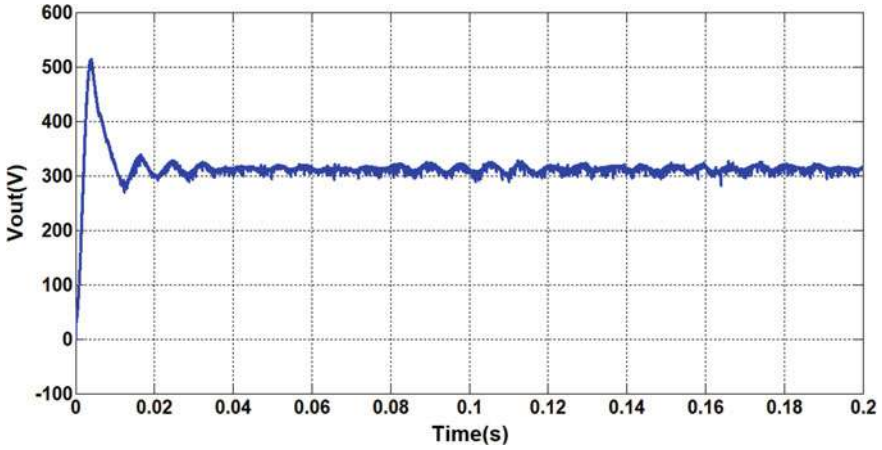


Fig. 12 Output voltage of HBC

4.3 Simulation of HBC with Input from SOFC

SOFC connected to hybrid boosting converter is modelled which is shown in Fig. 18. Value of each capacitors C1a and C1b is (100e−6)/3 F , C2a and C2b is (100e−6)/4 F . The inductor used in inductive switching core is of value 500e−6 H. The switching frequency is 40 KHz. The output DC voltage of SOFC is 35.17 V; it is given as a input to boost converter. The sOFC input to HBC and the output voltage of HBC are shown in Fig. 19. The simulation is carried out for 0.2 s. The expected output voltage according to equations is nine times of the input voltage. The voltage is boosted to 318.3 V.

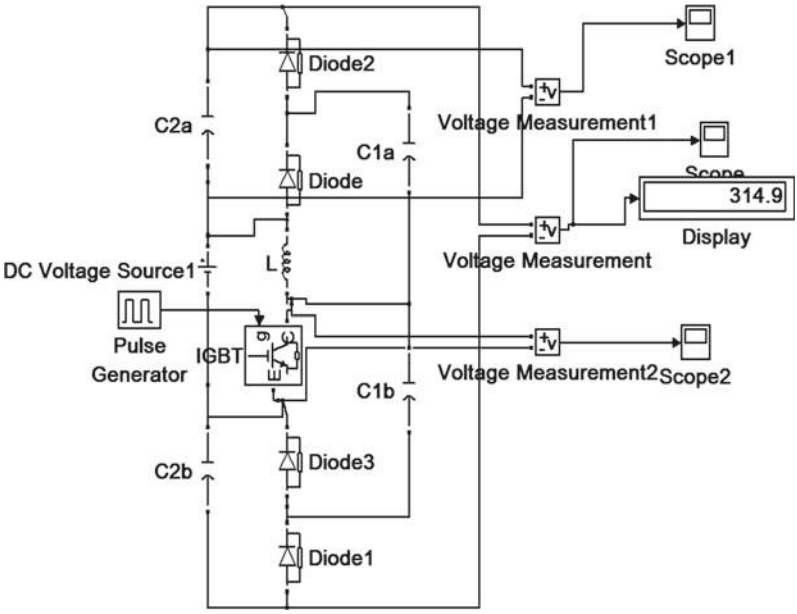


Fig. 13 HBC equivalent model

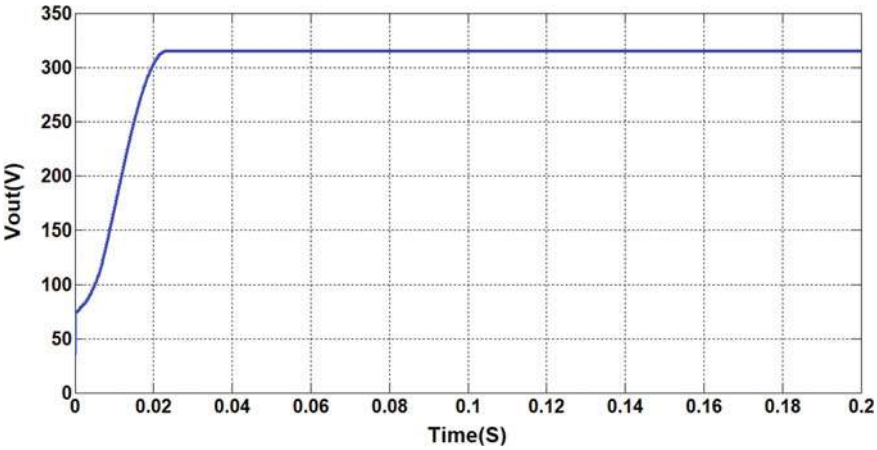


Fig. 14 Output voltage of HBC equivalent model

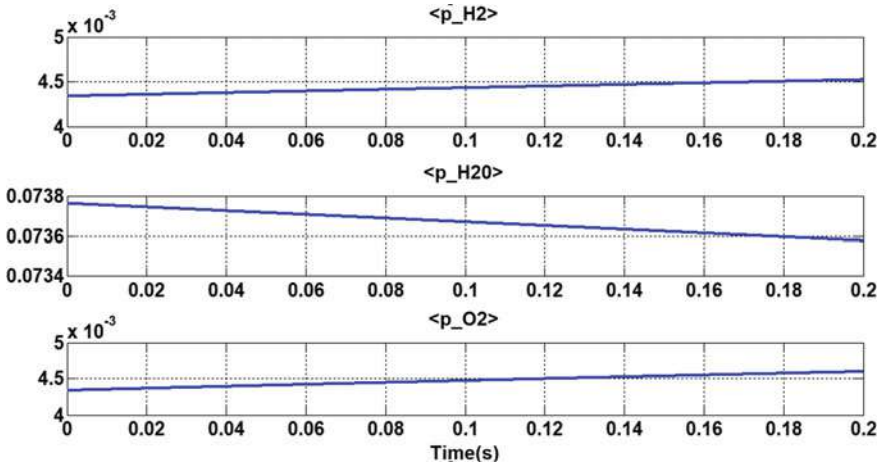


Fig. 15 SOFC: inputs: H_2, H_{2O} and O_2

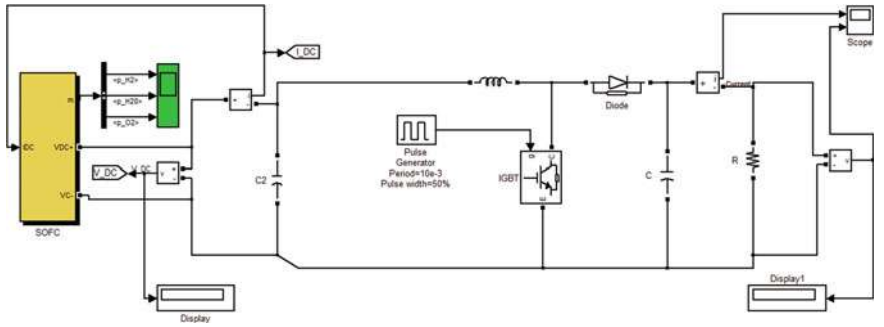


Fig. 16 SOFC with conventional boost converter

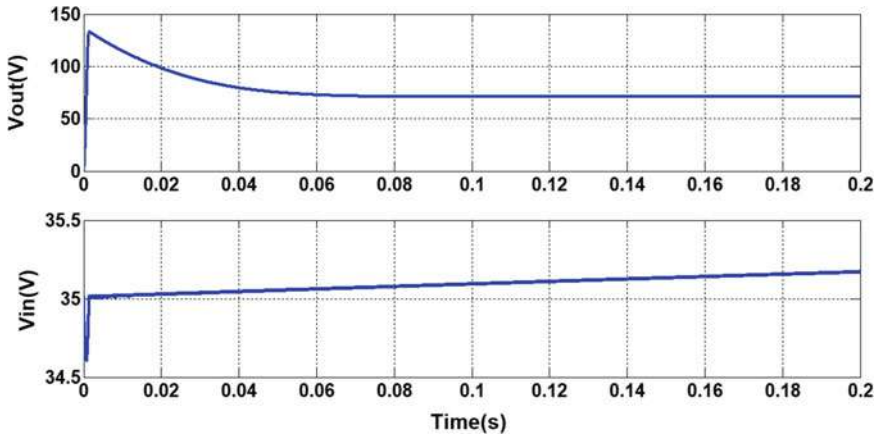


Fig. 17 Input of from SOFC and output voltages of conventional boost converter

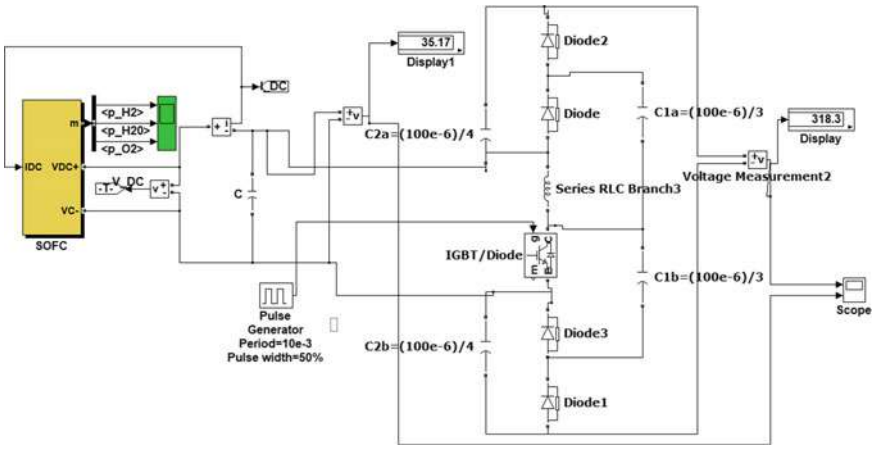


Fig. 18 SOFC with hybrid converter

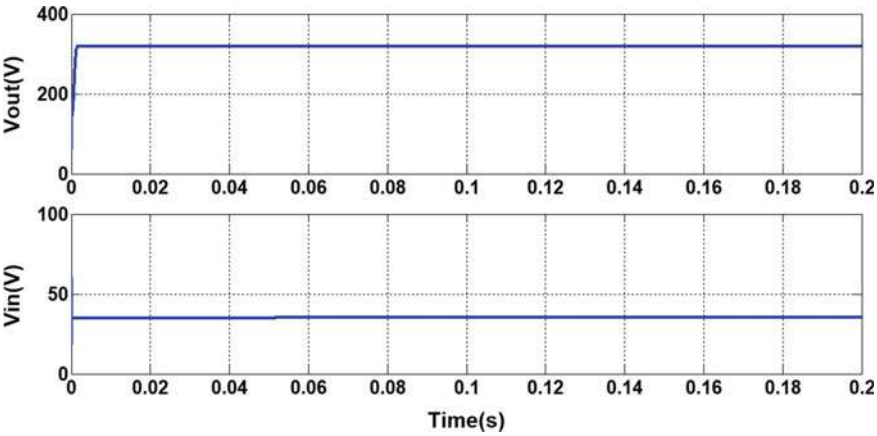


Fig. 19 Output voltage of HBC and the corresponding input form SOFC

5 Conclusion

In this work, discussed the importance of the boost converters in the modern power systems consisting of renewable energy system and different storage systems. Further, the work is concentrated on building the solid oxide fuel cell system. Different case studies have been carried to shown the performance of the a new hybrid boost converter: Two level hybrid boost converter-theoretical model, equivalent to improve the output of the converter, and implementation of a new hybrid boost converter for high gain enhancement of SOFC system and compared the results with the conventional boost converter. The simulation results have shown the performance of the new hybrid boost converter for high gain enhancement.

References

1. India Energy Outlook.: India Energy Outlook 2021. Available: http://www.mospi.nic.in/sites/default/files/reports_and_publication/ES/EnergyStatisticsIndia2021.pdf (2021)
2. Occhipinti Z, Verona R (2020) Kyoto protocol (KP). Climate action, Springer, Springer, pp 605–617
3. Muttaqi KM, Islam MR, Sutanto D (2019) Grids future power distribution grids: integration of renewable energy, energy storage, electric vehicles, superconductor, and magnetic bus. IEEE Trans Appl Supercond 29:605–617
4. Bhatt B, Singh A (2021) Power sector reforms and technology adoption in the Indian electricity distribution sector. Energy 215:118797
5. Verma MK, Mukherjee V, Yadav K, Mukherjee V, Gupta S (2020) Indian power distribution sector reforms: a critical review. Energy Policy 144:111672
6. Ghosh S, Yadav VK, Mukherjee V, Gupta S (2021) Three decades of Indian power-sector reform: a critical assessment. Utilities Policy, Elsevier, pp 101158
7. Muthukumar M, Rengarajan N, Velliyangiri B, Omprakas, Rohit MA, Raja CB, Kartheek U (2021) The development of fuel cell electric vehicles—a review, materials today: proceedings, vol 45, Elsevier, pp 1181–1187
8. Manish Kumar S, Parag N, Amandeep Singh O (2021) Hydrogen fuel and fuel cell technology for 61a cleaner future: 61a review. Environ Sci Pollut Res 1–20
9. NREL.: Ten principles for power sector transformation in emerging economies. Available: https://www.nrel.gov/docs/fy15osti/62611.pdf
10. Starzyk JA, Jan Y-W, Qiu F (2001) A DC-DC charge pump design based on voltage doublers. In: IEEE transactions on circuits and systems i: fundamental theory and applications, vol 48, IEEE, pp 350–359
11. Luo FL, Ye H (2004) Positive output multiple-lift push-pull switched-capacitor Luo-converters. IEEE Trans Ind Electron 51:594–602. IEEE, pp 350–359
12. Rahimi R, Habibi S, Shamsi P, Ferdowsi M (2021) A dual-switch coupled inductor-based high step-Up DC-DC converter for photovoltaic-based renewable energy applications. In: 2021 IEEE Texas Power and Energy Conference (TPEC), IEEE, pp 1–6
13. Allehyani A (2019) Analysis of A transformerless single switch high gain DC–DC converter for renewable energy systems. Arabian J Sci Eng 1–12
14. Alzahrani A, Ferdowsi M, Shamsi P (2021) A family of scalable non-isolated interleaved DC-DC boost converters with voltage multiplier cells. IEEE Access 7:11707–11721
15. Pasha SR, Mallika B (2016) A new hybrid boosting converter for renewable energy applications. Int J Adv Technol Innovat Res 8:2348–2370
16. Singh M, Zappa D, Comini E (2021) Solid oxide fuel cell: decade of progress, future perspectives and challenges. Int J Hydrogen Energy
17. Nikhilesh J, Singh SB (2021) Modeling and simulation of grid-connected wind power plant for electric vehicle charging station with solid oxide fuel cell. In: Latest trends in renewable energy technologies. Springer, pp 1–10
18. Lyu Y, Xie J, Wang D, Wang J (2020) Review of cell performance in solid oxide fuel cells. J Mater Sci 55:7184–7207
19. Bove R, Ubertini S (2008) Modeling solid oxide fuel cells: methods, procedures and techniques. Springer Science & Business Media

Chapter 49
Using AI Chatbots in Education: Recent Advances Challenges and Use Case

Moneerh Aleedy, Eric Atwell, and Souham Meshoul

1 Introduction

Technologies have evolved over the years; what we saw ten years ago as a technological revolution is now becoming ordinary. We are currently in the era of AI, where science is behind every development that simulates human thinking. Some common scientific fields where AI is predominant include computer vision, pattern recognition, natural language processing (NLP), understanding (NLU) and generation (NLG), robotics, and planning to name just a few. Moreover, AI technologies are at the pace of creating revolutions and paradigm shifts in almost all areas such as education, health care, business, engineering, and automotive. A chatbot is an AI technology powered by natural language processing techniques to learn and understand human language. It has become one of the most important tools governments and private sectors use to provide continuous communication to users 24 h a day. Recently, many universities and educational institutions have started to use chatbot technologies to help students in the admission process and academic advising along with the learning process. Faculty members and teachers can use chatbots to follow up with students, respond to inquiries, review students' backgrounds about subjects, and assess homework, among many other tasks. The role of chatbots in education is clear, but their design and use by either teachers or students face many challenges and difficulties.

The contribution of this work is twofold. First and as a review, the progress in developing AI chatbots for education is described, emphasizing up-to-date algorithms used proposed for their design and the challenges related to their imple-

M. Aleedy (✉) · E. Atwell
School of Computing, University of Leeds, Leeds LS2 9JT, UK
e-mail: scmmale@leeds.ac.uk

M. Aleedy · S. Meshoul
Information Technology Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*,
Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_50

mentation. Second, through a use case, we show how an AI chatbot can be leveraged to collect feedback from students regarding a course, namely machine learning in our case, and analyze the results.

This paper will be structured as follows: Sect. 2 reviews chatbots, their categories based on the design approaches, the uses of chatbots in education, and discusses challenges in languages, implementations, and education. In Sect. 3, a use case in using chatbots in education is conducted to support this research. A discussion about the use case is provided in Sect. 4. Finally, Sect. 5 will conclude the overall paper.

2 Literature Review

2.1 Chatbots: Definition and Structure

Generally, chatbots simulate the interaction between humans. They are widely used in education, banking, e-commerce, and business as tools to help customer support [1]. A chatbot can be defined as a conversational tool that allows users to operate computers in a simple natural language that people can understand [2]. Another attempt to define the chatbot describes it as a programmed tool that interacts naturally with the user on a specific topic or subject. This interaction can be by voice or text [3, 4].

Before getting deep into chatbot technology, it is important to understand its structure. The chatbot architecture depends on the domain specified for it, but the basic workflow remains the same. As shown in Fig. 1, a chatbot takes input (text, voice, or both) from the user. Then, the input is passed to the natural language processing (NLP) component in the form of text to learn and understand the user's input. Response constructor uses different algorithms to process the pre-defined Knowledge base, presents a set of suitable responses, and then passes it to the response selector. Response selector uses machine learning and AI algorithms to choose the most appropriate response [5].

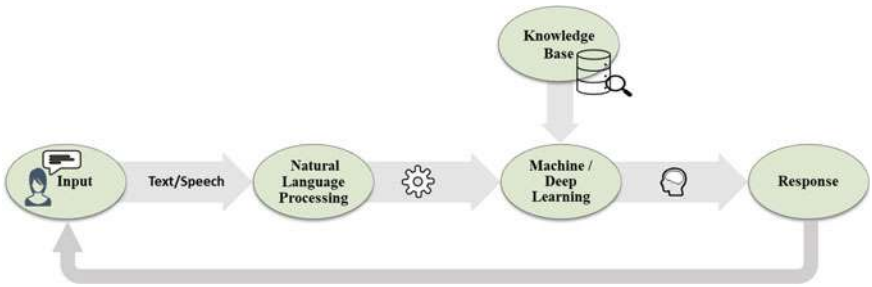


Fig. 1 General chatbot workflow

Chatbots can be classified into two broad categories based on their use. According to [2], chatbots can either be service chatbots or social chatbots. This classification depends on the primary function it performs and the way that its programs operate. Moreover, service-oriented chatbots provide users' services in a friendly and fast way, such as online customer service, banking service, and weather service. On the other hand, social chatbots care more about the emotional side of customers and their satisfaction.

Other researchers classify chatbots based on the knowledge domain, service-provided, goals, and input/responses generated [6]. The knowledge domain defines the chatbot access domains if it is open or close. The open-domain chatbots make conversations on general topics and give meaningful responses. In the closed domain, the chatbots answer questions from a specific domain and fail to respond to other domains' questions. On the other hand, service-based chatbots include chatbots that provide personalized, in-kind, and overlapping services between agents. Moreover, goal-based chatbots include informative, conversational, and specific task chatbots. Finally, the input method and responses generated chatbots accept input, then process it to generate output in natural language, process input based on rules, or use both techniques [6].

As can be seen in Fig. 2, chatbots can be classified in different ways depending on some parameters such as functionality, knowledge domain, service-provided, goals, and input method/generated responses. Therefore, the development of a chatbot can be viewed as a combination of choices related to the aforementioned parameters.

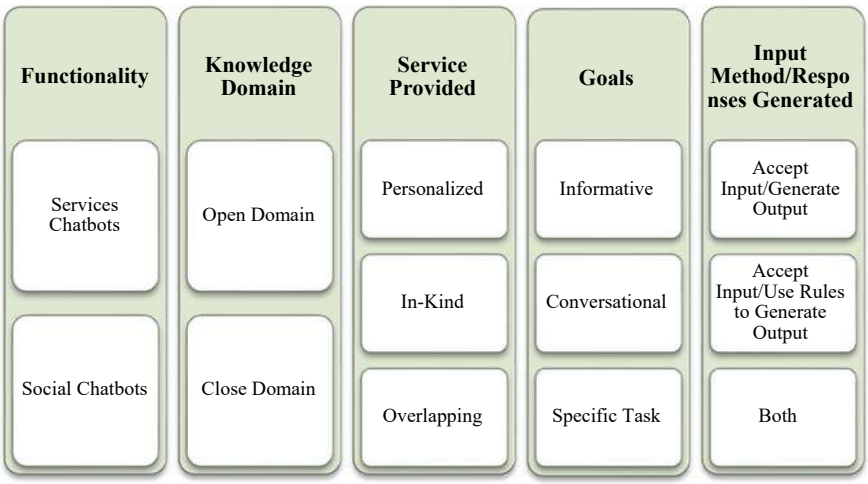


Fig. 2 Chatbots classification

2.2 Approaches for Chatbots' Design

Chatbots can be classified based on the developing approach into three main groups: rule-based, deep learning, and ensemble approaches [5]. Table 1 presents a summary of the used approaches in the design of chatbots.

As it combines the advantages of the three main approaches rule-based, retrieval-based, and generative-based, the ensemble approach has been shown to significantly improve the performance of chatbots [5, 16]. Moreover, specifying the conversation domain and merging different methods with neural networks' ability make the ultimate system robust enough for real-world application [5, 16].

Table 1 Chatbots' design approaches

Approach	Definition	Pros/Cons	Example
Rule-Based Approach	It is a straightforward method with pre-defined questions and answers that match user input to a rule pattern and select a pre-defined answer from a set of responses using pattern matching algorithms. The user's question is decomposed into a set of words and then used to choose the correct output	• Simple • Ideal for closed-domain communications • Not able to answer patterns that don't match pre-defined script	ELIZA, PARRY, ALICE [5, 7–9]
Machine Learning approach	*Retrieval-Based*: select the conversation's proper responses from a database using a neural networks algorithm and reuse it to reply	• Can return incorrect outputs because they are totally based on retrieval of data • Cannot handle unseen queries for which pre-defined responses do not exist	MITSUKU, FIT-EBot [8, 10–12]
	Generative-Based generates proper responses during the conversation; it does not depend on pre-defined questions and answers. Instead, it uses neural network models or deep learning techniques to develop a dialogue with the user	• Do not rely on pre-defined responses • Useful for involving a person in informal open-domain conversations • Not ideal for closed-domain communications • Can generate mistakes • Difficult to be trained	CAiRE [8, 11, 13]
Ensemble approach	A hybrid approach uses rule-based, retrieval-based, or generative-based approaches to respond to user input if there is no match to any rules	• Capable of stimulating conversation on unspecified subjects • Improve the performances of chatbots	AliMe, Medbot [5, 14, 15]

Each of these approaches applies one or more of the following techniques:

- **Pattern Matchers**: Used in traditional chatbots, it answers the user only if there is a pattern (lexical) matching between the user query and set of question–answer stored in its knowledge base. This kind of technique is considered inflexible; it lacks the intuitive ability of humans to see meaning, and the answers are given using a set of pre-defined responses [11].
- **Natural Language Processing (NLP)**: Natural language processing (NLP) is a field of artificial intelligence that studies how computer systems can interpret and control natural language. Information about understanding and using human language is collected to create appropriate technologies for computer systems to manage human language and perform many tasks. Most NLP techniques are based on machine learning. They consist of Natural language understanding (NLU) which develops the task to understand text, and natural language generation (NLG) which presents the responsibility for text generation that is commonly performed by neural networks [17].
- **Machine Learning (ML)**: Algorithms that learn from conversations. They consider the whole dialogue context, not just the current turn, and do not require a pre-defined response for each possible user input. However, it needs an extensive training set, which can be very difficult to provide [17].
- **Deep Learning (DL)**: Chatbots built using artificial neural networks learn everything from data and conversations with humans. It is trained to develop its own opinion on the text and be more effective by using more and more data. Retrieval-based and generative-based are the two major types of bots created using deep learning [5].
- **Reinforcement Learning (RL): Reinforcement learning** is a machine learning framework for learning and continuously improving policies by a computer algorithm through interacting with its environment. Reinforcement learning algorithms that incorporate deep learning (implemented as a neural network) are known as deep reinforcement learning (DRL) [18, 19].

In recent years, the chatbot's technologies powered by combining more than one technique in the field of AI and NLP to accurately mimic the real-life conversations between humans and overcome the limitations of the traditional chatbot.

2.3 The Role of Chatbots in Education

Many researchers studied and evaluated the benefits of using chatbots in educational institutions. They present how the educational agent participates directly in the online discussions, assists the teachers in following the chat discussion, uses natural language processing (NLP) to understand and find the main points in the student discussion, and applies sentiment analysis of the chat responses [20].

A study on Facebook messenger educational chatbots focuses on topics and language used in conversation. The development platform showed that (89%) of the chatbots under investigation use English as their primary communication language. Moreover, about (46%) of chatbots cannot be used to discuss any technical issues or simulate human conversation [3]. Most of the chatbots' topics were learning languages, economic topics, simulating conversation with historical figures, and learning computer science concepts [1].

A chatbot offers an interactive way of learning, similar to the one-to-one interaction between student and teacher. The chatbot provides educational features, including solving individual problems (e.g., current student, parents, graduated). The chatbot has become more important from another perspective, as it saves costs by replacing human assistants and increasing user satisfaction because of the speed response and availability 24 h. From the students' perspective, and by using the chatbot, students can expand their knowledge, exchange ideas, reduce learning loads, and feel more connected, especially the first-year students [21, 22]. Chatbots become suitable for teaching individuals at different levels and abilities because it increases their confidence. From the teachers' perspective, they can use the chatbot to send announcements to students, help correct homework (e.g., identify spelling and grammar mistakes), and track student progress [22].

Moreover, chatbots are used in education to handle frequently asked questions, administrative tasks, student mentoring, motivation, student learning assessments, simulations, training, and providing feedback. For example, in a real case in Cardenal Herrera University, a chatbot acts as a personal assistant to monitor students, predict their behavior, answer administrative questions, and give advice [6].

The most important characteristic of a chatbot is its availability for learners on their mobile devices. It provides an educational method suitable for modern students in terms of speed and convenience, where communication is possible through mobile at any time and place. It also provides immediate feedback and follow-up to students based on their understanding and repetition of information according to their needs. Shy students interact greatly through chatbots, especially learning new languages; it helps them speak fluently and comfortably without judgment because they are sure they are not talking to real people [23].

The educational content is usually huge, and thus students are less interested in starting to learn due to a large number of scientific materials. For example, the database systems course contains many exercises, explanations, examples, and tasks that are often considered difficult for students. However, it is possible to motivate students to start solving them by gradually presenting these tasks, which helps the student search quickly through a chatbot teacher, which can be easily accessed in several ways, including websites or social media [24].

In some higher education institutions in India, an experimental research has been conducted to explore the factors that affect chatbot technology adoption. A sample of 47 students and a series of 10 questions have been used to conduct the study. One of the questions in the study measures the response time by the communication channels, around 61% of the students choose chatbot as the quickest communication

tool, online chat takes second place with 19%, phone calls 8.7%, 6.5% for face-to-face communication, social media and e-mail get 2.2% for each [6].

Another experiment has been conducted on a group of undergraduate students in the university in Hong Kong for the course principles of networks while using a chatbot named Infobot for 15 min. Infobot answers questions related to the educational material, students' grades, academic schedule, and the teacher's office hours. The experiment uses a questionnaire to measure students' satisfaction with the information provided by this chatbot; this information has been previously entered into the database by the teacher. The vast majority of students, nearly 60%, agreed on this chatbot's efficiency and how it helped them get the required information faster and relieve pressure on the teacher by facilitating communication and accelerating learning [25].

In the Faculty of Information Technology of Ho Chi Minh City University of Science (FIT-HCMUS), the increasing demand for learning has put a lot of pressure on this higher education institution. The clear proof of this is that the number of students per teacher increases, which means that each teacher's support for each student has decreased dramatically. This is one of the main reasons that lead to ineffective learning and a high dropout rate. Although many solutions are proposed to solve this problem, most of them cannot be successfully implemented due to financial and organizational difficulties. Scholars and administrators have begun to introduce chatbots into the education sector to face these challenges. 2018 was the year when chatbots were popular in education. One of the chatbot's most significant advantages is its ability to support students individually and with interest. This is especially useful in large-scale learning environments like universities or open online courses (MOOCs) [12].

In this research, we concentrate on the use of chatbots in education institutions. Table 2 shows some education chatbots.

There are numerous public and private universities in Saudi Arabia to teach the growing number of students. Therefore, an increase in the number of students means an increase in the number of inquiries 24 h a day. This requires a sufficient number of employees to achieve the required level of service quality. After investigating the websites of Saudi universities, the results indicated that chatbots are not fully utilized [29].

2.4 Challenges

Many challenges face the effective use of chatbots; this section covers some of them in terms of languages, implementations, and education.

Language Challenges:

Chatbots are developed for many languages such as English, Arabic, and French. However, each language has its own sentence structure, punctuation rules, and the use of spaces, which is a barrier for current chatbots to deal with [11, 30]. For

Table 2 Educational chatbots

Chatbot name	Purpose	Notes
FIT-EBot [12]	• Provide administrative and learning support to students at the Faculty of Information Technology of Ho Chi Minh City University of Science, Vietnam (FIT-HCMUS) on behalf of the academic and administrative staff	• Users can interact directly with the FIT-EBot through their Facebook pages • The data collected from the FIT-HCMUS will better support students compared to other general chatbots
LABEEB: (means Wiseman in Arabic) [26]	• Conversational agents respond to student inquiries in specific courses, objectives, learning outcomes, and academic rules and regulations	• Labeeb successfully receives inquiries in either text (English or Arabic) or speech format (English)
Percy [27]	• Percy is a computer science teaching assistant chatbot that teaches CS 221 course at Stanford University	• If a similar question has already been asked, the chatbot is generally able to retrieve it. If no similar question has been asked, the bot can recognize that and respond, "I don't know." • Percy performs well on policy questions
Python-Bot [28]	• Chatbot for teaching Python programming	• The design of Python-Bot was user-friendly, making it easy for students to use

English chatbots, the authors [11] mentioned several points that affect the efficiency of chatbot conversations; the most important ones include the inability to recognize grammatical errors and similar meanings questions. Moreover, information retrieval from a database is not realistic; two questions may look different in terms of words but have the same meaning, so similarity measures should be used to eliminate differences [11, 31].

On the other hand, the Arabic language chatbots have many linguistic complexities, it consists of several variants that are quite different from each other: Modern Standard Arabic, the official written/read language, and several dialects. Moreover, Arabic writers make very common mistakes in spelling some problematic letters, such as Alf Hamza and Ta Marbouta. Morphological richness of Arabic words are influenced by a large number of features such as gender and numbers. Also, verbs, adjectives, and pronouns are all gender-specific, requiring the chatbot to have two different responses systems—one for male users and one for female users. As in other languages, Arabic has its own set of unique dialogue expressions, for example, while the English greeting expression "good morning" gets the answer "good morning," the Arabic equivalent greeting "صباح الخير" "Morning of Goodness" gets the answer "صباح النور" "Morning of Light" [32, 33, 34]. A collaboration between linguists and computer scientists is required to overcome these challenges.

Implementation Challenges:

Users sometimes tend to initiate domain queries during a conversation with a chatbot and then suddenly move to another domain; this may lead to significantly weak accuracy [10, 11]. Moreover, the performance of a chatbot is dependent on the size and accuracy of the databases, so the higher the database size, better the performance. The semantically similar questions unidentified by the system would be real problems unless the developer included the semantically equivalent words in the terminology section [10, 11, 31]. Moreover, lack of chatbot personality can also push users away from the dialogue, this risk can be reduced by giving the chatbot a name and an avatar. Gender and time recognition are real issues; the chatbot has no sense of time and often replies with the same "Have a good day" or "Good morning," regardless of the time of day [10]. In the research [35], some issues regarding the question answering system (retrieval-base) are found. The research mentioned that the system answer questions related to only a particular domain. The user must follow a specific format when asking questions. They suggest using a sequence-to-sequence model (generative-base) to overcome this issue.

Educational Challenges:

In educational institutions, many challenges are facing the implementation of chatbots. Students feel that they are not interested in speaking with chatbots because the conversations have become monotonous with the long period of use. Therefore, more attention needs to be paid to improving chatbots to be more like chat agents [31]. Furthermore, the chatbot does not understand students' feelings and their satisfaction or anger. Also, it does not give advice or ask questions [11].

3 Use Case

Hubert is an artificial intelligence chatbot that replaces the traditional way of filling surveys by chatting with users. Currently, Hubert is aimed toward three different domains: customer experience, human resources, and education. Depending on the choice, the survey is set up using specific questions about that domain. Hubert consists of two main parts: the chatbot itself and the dashboard. In the dashboard, the user chooses the domain and then customizes the pre-defined questions. Finally, it gives the user options to start talking with the chatbot (copy the link or e-mail the participants). When the participant clicks on the chatbot link, a window with a typical chat layout opens, and Hubert starts the conversation [35].

To get a closer look at the importance of chatbots in education, a practical experiment was performed on Hubert chatbot to get student feedback about a machine learning course. Since it was the middle of the semester, while applying the experiment, we chose the informal course evaluation—during the semester version of Hubert.

This paper focuses on using AI in education, and Hubert provides a great tool to serve this goal. The education domain of the Hubert chatbot allows teachers to get direct feedback from the student. The students' feedback allows the teachers to identify gaps in their teaching efforts and do better. The Huber chatbot makes inquiries like how the course can be improved, what to change, what is functioning admirably, and what is not. The chatbot investigates the conversations, does some analysis, and then sends it to the instructors [36].

Feedback Hubert's website contains several models that provide feedback based on the need of the teacher: stop/start/continue evaluation for mid-term feedback, students' self-reflection—beginning/middle/end of the semester, informal course evaluation—during the semester/at the end of the semester and customized template for the Edubots project.

All participants are postgraduate students (26 female students) from Princess Nourah bint Abdulrahman University in Riyadh, Saudi Arabia. During the lecture, the instructor shared a link with the student. The link opens the Hubert chatbot webpage directly and asks the students if they are ready to start (see Fig. 3).

The Hubert chatbot starts asking students about their opening on the course and what the instructor should do and stop doing. Moreover, Huber encourages students to give some solutions and not participate only with yes and no. If a student does not provide a real answer, Hubert tries to encourage him by asking a follow-up question (see Fig. 4).

In the end, Hubert asks the students to evaluate the experience of talking to it and then terminates the conversation (see Fig. 5).

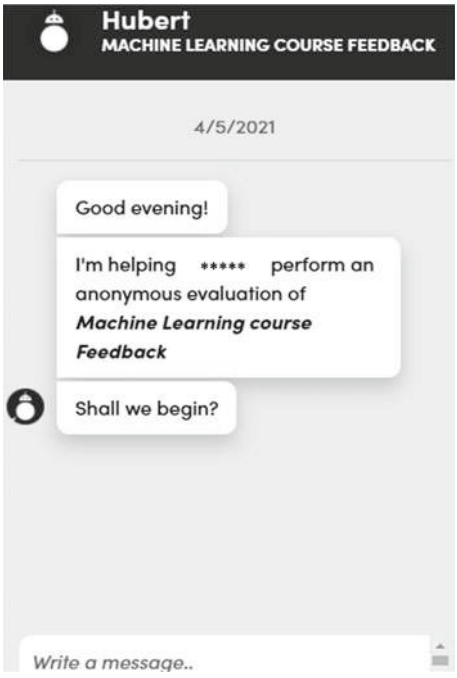


Fig. 3 Start of the conversation

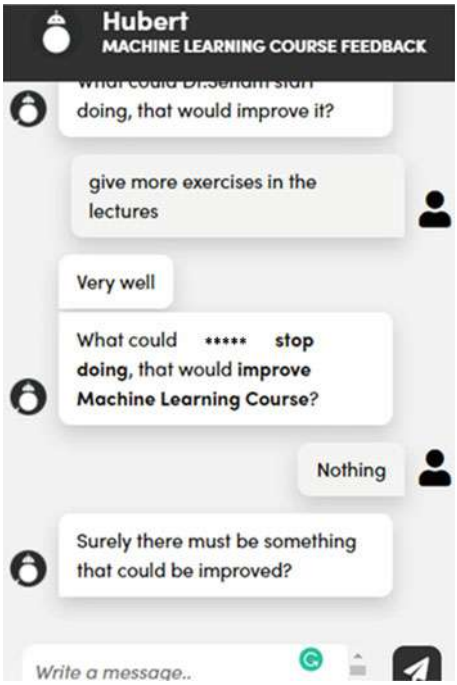


Fig. 4 Follow-up question to encourage student

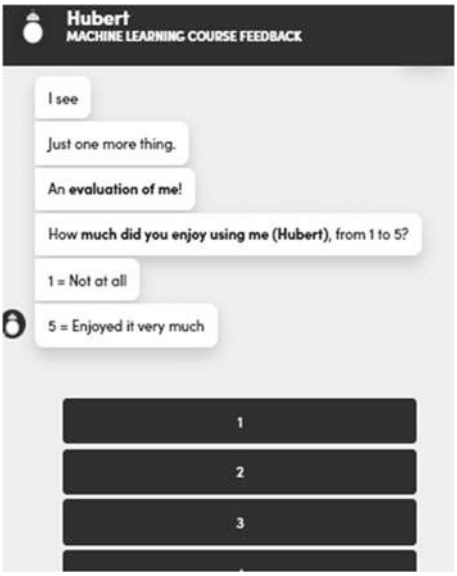


Fig. 5 Evaluating Hubert

On the other hand, the instructor can view all the transcripts with the students, and an analysis of them in the dashboard. The dashboard provides the most discussed topic and sentiment analysis for each student's answer, if positive, negative, or neutral. The instructor can change the sentiment analysis if he sees it incorrect.

The feedback chatbot asks seven main questions as follows:

- What is working well with _____ and should **continue** in the same way?
- What is working well with _____ and should **continue** in the same way?
- What could _____ **stop doing**, that would **improve** _____?
- What class activities or **assignments** help you **learn the most** so far?
- What is your **overall experience** of _____ so far?
- Are there any **other points** you would like to comment on?
- How **much did you enjoy using me (Hubert)**, from 1 to 5? [37]

4 Discussion and Results

Our experiment got 100% responses from the class students; they were excited to talk to the chatbots. Using the dashboard makes it easy to know the strengths and weaknesses of the course and clarify the most important points in students' conversations with the chatbot to develop the course. If the students interact with negative responses such as "nothing" and "no," they are encouraged by the chatbot to participate more by asking a follow-up question, "Surely there must be something that could be improved?".

In our experiment, around 42% of the students responded negatively at first, but the percentage reduced to about 7% after motivating them with another question. At the end of the experiment, students were asked to rate their talking experience with the chatbot. Hubert received a 3.6 out of 5, with 19 students rating 5, 4 students rating 4, and 3 students rating 3 (see Fig. 6).

As a disadvantage, the dashboard needs more improvement as a disadvantage since it may give false results because it counts the incomplete answers. For example, if the student answers some questions, quits the conversation, and then comes back later and answers the same questions, it will count them as two different responses. Moreover, it may also give misleading results because it measures the frequency of words regardless of their meaning. Furthermore, teachers have no control over the responses that appear in the dashboard; they cannot hide or delete any of them. For that reason, the summaries in the dashboard are not very helpful to the teachers.

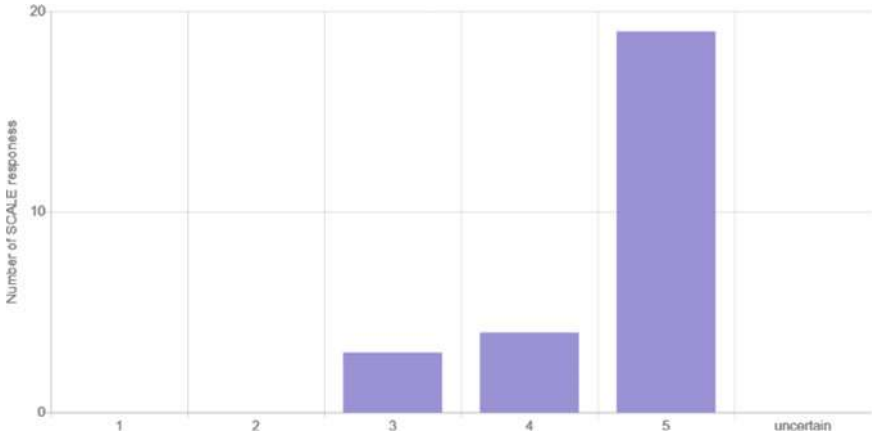


Fig. 6 Hubert evaluation

5 Conclusion

This study explores the existing chatbots' domain of use, language characteristics, and design approaches. The study also focuses on the most important practices of using educational AI chatbots as a parallel educational tool, virtual teaching assistant, trainer to practice English as a second language, and virtual agent to get student feedback. In addition to their ability to serve, some issues are related to registration and administrative problems for students. Educational chatbots are still in their early stages, and many researchers have used recent AI techniques such as in deep learning and reinforcement learning to improve the performance of chatbots. However, there are still major technical, linguistic, and psychological challenges. Recent researches suggested combining more than one technique in the field of AI and NLP to overcome the challenges of the existing chatbots.

References

1. Verleger M, Pembridge J (2018) A pilot study integrating an AI-driven chatbot in an introductory programming course. In: Proceedings—frontiers in education conference, FIE, 2019, Oct 2018
2. Duncker D (2020) Chatting with chatbots: sign making in text-based human-computer interaction. Sign Syst Stud 48(1):79–100
3. Smutny P, Schreiberova P (2020) Chatbots for learning: a review of educational chatbots for the facebook messenger. Comput Educ 151:103862
4. Miklosik A, Evans N, Qureshi AMA (2021) The use of chatbots in digital business transformation: a systematic literature review. IEEE Access 9:106530–106539

5. Singh S, Thakur HK (2020) Survey of various AI chatbots based on technology used. In: ICRITO 2020—IEEE 8th international conference on reliability, infocom technologies and optimization (trends and future directions), pp 1074–1079
6. Sandu N, Gide E (2019) Adoption of AI-chatbots to enhance student learning experience in higher education in india. In: 2019 18th International conference on information technology based higher education and training, ITHET 2019
7. Weizenbaum J (1966) ELIZA-A computer program for the study of natural language communication between man and machine. Commun ACM 9(1):36–45
8. Nuruzzaman M, Hussain OK (2020) IntelliBot: a dialogue-based chatbot for the insurance industry. Knowledge-Based Syst 196:105810
9. Yeung Shum H, Dong He X, Li D (2018) From Eliza to XiaoIce: challenges and opportunities with social chatbots. Frontiers Inform Technol Electron Eng. Zhejiang University 19(1):10–26, 01 Jan 2018
10. Wu Y, Wu W, Xing C, Xu C, Li Z, Zhou M (2019) A sequential matching framework for multi-turn response selection in retrieval-based chatbots. Comput Linguist 45(1):163–197
11. Nuruzzaman M, Hussain OK (2018) A survey on chatbot implementation in customer service industry through deep neural networks. In: Proceedings—2018 IEEE 15th international conference on e-business engineering, ICEBE 2018, pp 54–61
12. Hien HT, Cuong PN, Nam LNH, Nhung HLTK, Thang LD (2018) Intelligent assistants in higher-education environments: the FIT-EBOt, a chatbot for administrative and learning support. In: ACM international conference proceeding series, pp 69–76
13. Lin Z et al (2019) CAiRE: an empathetic neural chatbot, Jul 2019
14. Qiu M et al (2017) AliMe chat: a sequence to sequence and rerank based chatbot engine. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: short papers), vol 2, pp 498–503
15. Bharti U, Bajaj D, Batra H, Lalit S, Lalit S, Gangwani A (2020) Medbot: conversational artificial intelligence powered chatbot for delivering tele-health after COVID-19, pp 870–875
16. Tammewar A, Pamecha M, Jain C, Nagvenkar A, Modi K (2017) Production ready chatbots: generate if not retrieve. arXiv
17. Adamopoulou E, Moussiades L (2020) Chatbots: history, technology, and applications. Mach Learn Appl 2:100006
18. Palasundram K, Mohd Sharef N, Kasmiran KA, Azman A (2020) Enhancements to the sequence-to-sequence-based natural answer generation models. IEEE Access 8:45738–45752
19. Rajamalli Keerthana R, Fathima G, Florence L (2021) Evaluating the performance of various deep reinforcement learning algorithms for a conversational chatbot, pp 1–8, Jun 2021
20. Ferreira-Mello R, André M, Pinheiro A, Costa E, Romero C (2019) Text mining in education. Wiley Interdiscip Rev Data Min Knowl Discov 9(6)
21. Chen LE, Cheng SY, Heh J-S (2021) Chatbot : a question answering system for student, pp 345–346, Aug 2021
22. Palasundram K, Sharef NM, Nasharuddin NA, Kasmiran KA, Azman A (2019) Sequence to sequence model performance for education chatbot. Int J Emerg Technol Learn 14(24):56–68
23. Dokukina I, Gumanova J (2020) The rise of chatbots-new personal assistants in foreign language learning. Procedia Comput Sci 169:542–546
24. Ondáš S, Pleva M, Hládek D (2019) How chatbots can be involved in the education process. In: 17th International, ieeexplore.ieee.org
25. Lee LK, Fung YC, Pun YW, Wong KK, Yu MTY, Wu NI (2020) Using a multiplatform chatbot as an online tutor in a university course. In: Proceedings—2020 international symposium on educational technology, ISET 2020, pp 53–56
26. Almurtadha Y (2019) LABEEB: intelligent conversational agent approach to enhance course teaching and allied learning outcomes attainment. J Appl Comput Sci Math 13(1):9–12
27. Chopra S, Gianforte R, Sholar J Meet Percy: the CS 221 teaching assistant chatbot. ACM Trans Graph 1(1)
28. Carlander-Reuterfelt D, Carrera A, Iglesias CA, Araque O, Sanchez Rada JFS, Munoz S (2020) JAICOB: a data science chatbot. IEEE Access 8:180672–180680

29. Almurayh A (2021) The challenges of using arabic chatbot in Saudi universities. IAENG Int J Comput Sci 48(1):190–201
30. Ayanouz S, Abdelhakim BA, Benhmed M (2020) A smart chatbot architecture based NLP and machine learning for health care assistance. In: ACM international conference proceeding security
31. Niranjan M, Saipreethy MS, Kumar TG (2012) An intelligent question answering conversational agent using Naïve Bayesian classifier. In: Proceedings—2012 IEEE international conference on technology enhanced education, ICTEE 2012
32. Ali DA, Habash N (2016) Botta: an Arabic dialect chatbot, pp 208–212
33. Al-Ghadhban D, Al-Twairesh N (2020) Nabiha: an Arabic dialect chatbot. Int J Adv Comput Sci Appl 11(3):452–459
34. AlHumoud S, Al Wazrah A, Aldamegh W (2018) Arabic chatbots: a survey. Int J Adv Comput Sci Appl 9(8):535–541
35. Houtsma MJ (2020) Perceived AI performance and intended future use in AI-based applications
36. Raj U, Jain R (2019) Methodologies in chat bot learning. SSRN Electron J
37. Hubert+1—Add more to your team [Online]. Available: https://hubert.ai/. Accessed 07 Apr 2021

Chapter 50
Automatic Segmentation and Classification of Brain Tumours on Pre-operative and Post-operative MRI Sample Using Deep Learning

K. V. Shiny and N. Sugitha

1 Introduction

In simple words, the irregular growth of cells or tissue in or around the brain is called brain tumour. In our human body, normally cells grow, get damaged, and die. Unfortunately, this is not the case in certain cases. Due to the start of a cancerous tumour(s), this cycle is disrupted. The tumour cells grow exponentially and do not perish, unlike healthy cells. New cells do not form, and the old cells remain in the body which leads to the building up of extra cells into a mass of tissues known as tumour. This may occur in any of the stages of human life but are different from each other. The majority of primary central nervous system (CNS) cancers are brain tumours, which account for 85–90% of all cases. The 5-year survival rate indicates how many patients live for at least 5 years after a tumour is found. People with a malignant brain or CNS tumour have a 36% 5-year survival rate. After ten years, the survival rate is estimated to be around 31%. Hence, an effective method for detection and classification of tumours is indispensable for ensuring better quality of life for the patients. Tumours are classified into two types, namely malignant (non-cancerous) and benign (cancerous) [1]. The exact reason for the brain tumour is not yet determined.

K. V. Shiny (✉)
Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, India
e-mail: kvshiny7@gmail.com

N. Sugitha
Department of Information Technology, Noorul Islam Centre for Higher Education, Kumaracoil, India

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_51

1.1 Need for Automatic Detection

MRI is a type of imaging that is used to assess the presence of cancers in brain pictures. MRI is a type of imaging that doctors use to diagnose and treat patients all over the world. MRI visualises organs, bones, and biological structures using magnetic fields, pulses, and a computer [2]. Automatic segmentation of brain tumours is critical in the medical industry, because it allows for more precise and reliable disease identification.

Despite the fact that automatic tumour segmentation is growing more common, automatic brain tumour segmentation is not a clinical practise. The purpose of the work is to locate tumour spots on a brain MRI and estimate the tumour region using symmetry analysis and a fully automated method employing a digital image processing technique [3]. The primary concern in identifying the tumour is its identification, due to the tumour fluctuating depths and locations. It also relies on the observer's ability to distinguish between tumour cells' location, length, and shape. For glioma diagnosis, T1-weighted, T2-weighted, and fluid attenuation inversion recovery (FLAIR) MRI data are frequently used [4].

1.2 Need for Segmentation in Pre- and Post-operational MRI Samples

Pre-operative images are the MRI image of a patient taken before surgery. The previous research has focussed on pre-surgery baseline scans, but this study takes a different strategy, segmenting gliomas in both pre- and post-surgery scans. The task of segmenting tumours on pre-operative baseline imaging differs greatly from that of segmenting residual and recurrent malignancies [5]. Due to the presence of surgically-forced cavities and non-neoplastic contrast enhancements on cavity boundaries, this segmentation method is extremely difficult. Manual activation of at least one seed site where the tumour began was required by state-of-the-art technologies. Seed spots cannot be visible in many post-operative images unless there is an obvious recurring tumour [6]. Finally, because the tumour bulk has been eliminated, the mass impact in many post-operative scans has been considerably decreased. As a result, post-operative scan segmentation no longer requires mass effect models. To address these challenges, a system that directly addresses the segmentation of gliomas in post-operative scans will be developed.

The following sections make up the remainder of this work: Sect. 2 describes the traditional brain tumour detection methodologies that have been used in the literature, as well as the challenges that have been encountered, which have inspired the development of the current proposed technique. Section 3 shows the proposed approach for detecting brain tumours using improved deep CNN. Section 4 compares the suggested technique's outcomes to those of alternative alternatives, and Sect. 5 concludes.

1.3 Research Objectives

- In MRI images, low- and high-grade gliomas are separated, and different deep learning algorithms are being investigated.
- To develop a hybrid system that focusses on pre-operative glioma segmentation whilst also including a component for post-operative scan segmentation.
- To design a hybrid model with deep learning and ML to classify the presence of meningioma, thereby giving an early diagnosis to reduce the mortality rate.

2 Literature Review

A timely and accurate identification of a brain tumour can improve the patient's chances of survival. Accurate diagnosis of various tumour types, on the other hand, is a difficult undertaking. As a result, utilising an optimization technique to create a capable tumour representation is a critical step in facilitating effective brain tumour identification [7]. The limits of a few existing brain tumour detection methods are evaluated here, and the limitations of each method served as inspiration for the development of a new brain tumour detection strategy.

Convolution neural networks (CNNs) were utilised to identify tumour locations by Bhargavi and Mani. To reduce noise and artefacts from MR pictures, the pre-processing phase was utilised [8]. The texture features were recovered using a GLCM with morphological operation after decomposing the pixel with a discrete wavelet. The tumour region was detected using a modified convolution neural community classifier [9]. It should be mentioned that the tumour determination is quick and precise. However, the approach was unable to estimate tumour depth, size, or location.

Adaptive convex region contour, created by Pandiselvi and Maheswaran, is a method for detecting tumour regions using MR images (ACRC) [10]. To evaluate if the region is aberrant or normal, SVM was used to organise the slice. When the SVM results are obtained, the aberrant slices are taken into account when the segmentation process is used [11]. For modelling 3D reconstruction, the rapid mode image matching (RMIM) technique was used. The MR imaging scan was used to evaluate the tumour regions using the 3D model. The method enhanced outcomes by detecting the tumour zone with great efficiency, although computations are time intensive.

Sultan et al. (2019) used publicly available datasets to construct a deep learning network that employs CNN to classify tumours. The model was used to classify cancers such as pituitary tumours, gliomas, and meningiomas. These tumours were utilised to distinguish between glioma levels [12]. Furthermore, two dropout layers were used to prevent overfitting difficulties in the softmax layer, which was used to forecast the tumour location. Although the approach enhanced accuracy, it was not suitable for large datasets.

Swati et al. (2019) employed a block-wise fine-tuning strategy based on pre-trained deep CNN to categorise tumours. Using an enlarged benchmark dataset, the

approach investigated the T1-weighted contrast [13]. It was a more ubiquitous procedure that required less pre-processing and was more exact. Although the technology beats the previous methods, it did not incorporate normal CE-MR pictures of the brain to differentiate cancers. Reviewed sources summarised in Table 1.

Table 1 Overview of the reviewed sources

Title	Pre-processing	Segmentation + classification	Feature extraction	Remarks
Segmentation of magnetic resonance imaging brain-based Techniques: a review of tumours	NA	• Deformable model • Machine learning-based segmentation • Atlas-based segmentation • Pixel-based segmentation • Region-based segmentation • Edge-based segmentation	NA	• Did not consider grading • One of the most widely used segmentation techniques is the ANN technique
Brain tumour detection approaches: a review	• Gaussian filter • Contour let transform • Median filter • Anisotropic diffusion filter	Manual segmentation Region-based method Threshold clustering	• Statistical • Topological feature	There is no such thing as a single method that is both appropriate and efficient
Using template-based K means and an improved fuzzy C-means clustering algorithm, automatic human brain tumour detection in MRI images	• Median filter • Operation based on histograms • Based on an adaptive histogram operation	The template-based K means and enhanced fuzzy C means (TKFCM) algorithm is a template-based K mean and improved fuzzy C-mean algorithm	NA	• Detects human brain cancers in a matter of seconds • Applicable for tumour core only
Greedy snake model and fuzzy C-means optimization for automatic brain tumour segmentation from MRI	• Median filter + morphological reconstruction	Snake contour detection + fuzzy C-means clustering	NA	• Identified AROI • Not applicable for multimodel segmentation

(continued)

Table 1 (continued)

Title	Pre-processing	Segmentation + classification	Feature extraction	Remarks
Using a deep convolutional neural network and the SVM algorithm, an intelligent diagnosis method for brain MRI tumour segmentation has been developed	• Morphological operations	• Fully automatic heterogeneous segmentation using SVM	• SVM, linear regression model	• Pre-operative MRI segmentation only
Deep learning-based fully automated brain resection cavity delineation for radiation target volume characterisation in glioblastoma patients		• Fully-convolutional densely connected architecture	• NA	• Manual contours by RT experts done in post-operative MRI
Using radiological scans, context aware deep learning was used to segment brain tumours, classify subtypes, and predict survival		Context encoding-based convolutional neural network	SVM, linear regression model	Pre-operative segmentation only
GLISTR boost: glioma segmentation using multimodal MRI segmentation, registration, and biophysical tumour growth modelling in combination with gradient boosting machines	Smallest univalue segment assimilating nucleus	Glioma image segmentation and registration (GLISTR) software		Semi-automatic, pre-operative segmentation only

3 Methodology

The flowchart of the proposed system for automatic segmentation and categorisation of brain tumours is shown in Fig. 1. Pre-processing should be done before applying segmentation algorithms since raw MRI images typically contain artefacts and non-brain regions that affect segmentation quality. The pre-operative MRI scans are supplied into this programmed for pre-processing, which includes filtering and region of interest (ROI) extraction [14]. Segmentation is performed on the retrieved images. To produce candidate brain tumour regions, the region growing algorithm is used. For further segmentation, the modified Bhattacharya distance is utilised. Segmentation is followed by feature extraction where it is done by histogram equalisation (HE) and GLCM [15]. HE is the easiest to use and effective of the numerous enhancement procedures. In HE, all of the pixels are spread uniformly, resulting in a more intense image than the original. The HE and GLCM are used to extract texture-based features and different edge regions. After feature extraction, in the classification step, the newly implemented Bir cat optimization algorithm is applied which is a deep learning concept that uses deep belief network for training the neural network. This will not only give the classified tumour tissues but also classify the

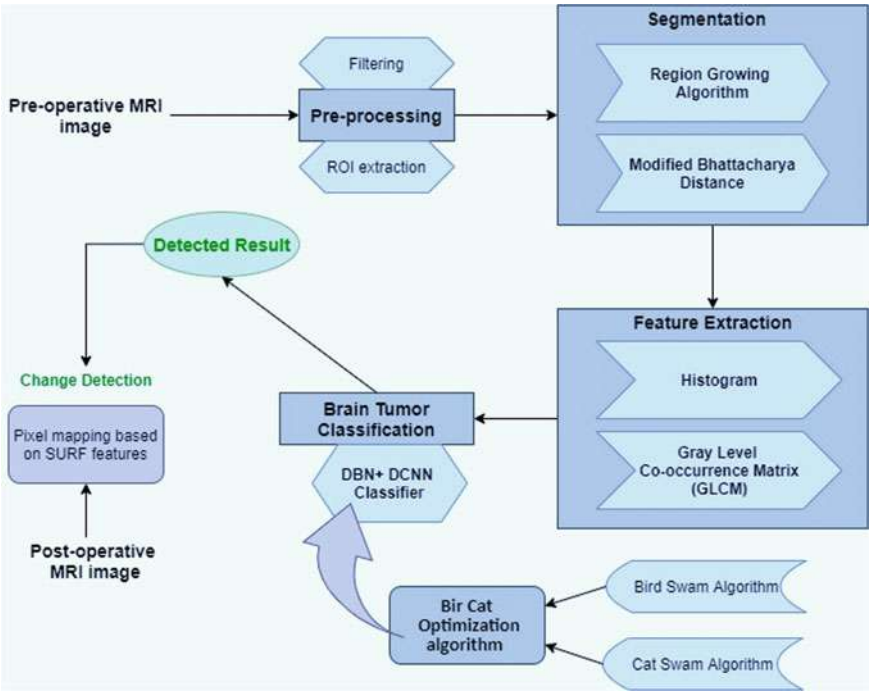


Fig. 1 Proposed block diagram for automatic segmentation and classification of brain tumour

various categories of abnormalities in brain tumour. Further, post-operative segmentation and classification are done which include all the above image processing steps that were done for pre-operative MRI [16].

3.1 Data Acquisition

The virtual skeleton database has pre- and post-operative multimodal MRI scans of patients with gliomas (LGGs, HGGs) as training pictures for the BRATS 2016 competition. Pre-operative baseline training scans from the BRATS 2013 challenge (10 LGGs and 20 HGGs), 44 LGG and 112 HGG pre-operative baseline scans from the National Institutes of Health (NIH) Cancer Imaging Archive (TCIA), and 88 HGG post-operative NIH-TCIA scans from the BRATS 2016 challenge are all included in the dataset [17].

27 patients' longitudinal observations at various time intervals are documented in the post-operative scans. The addition of these longitudinal scans will allow experienced neuroradiologists to classify these tumours as "progressing," "stable disease," or "shrinking" based on volumetric segmentation labelling. The pre-operative and post-operative testing groups were separated by ocular confirmation of a surgically imposed cavity [18].

Each patient's data include T1-weighted, T2-weighted, and T2 fluid-attenuated inversion recovery (FLAIR) MRI volumes, both native and contrast-enhanced (CE) [19]. Finally, the training set's ground truth (GT) segmentations were provided. The BRATS 2013 data were manually labelled, whereas the NIH-TCIA data were automatically annotated with a mix of expert annotations. The GT segmentations include the enhancing component of the tumour (ET), the tumour core (TC), which is defined as the union of necrotic, non-enhancing and enhancing components of the tumour, and the overall tumour (WT), which is defined as the TC plus the peritumoral edematous area [20].

3.2 Pre-processing

Because it smooths and normalises MRI images, pre-processing is essential in digital image processing. The impact of black portions in the limits of brain pictures is minimised when pre-processing is done [21]. Image enhancing functions such as noise reduction, de-noising, filtering, and skull removal are performed in this step. Using a pre-processing approach, it improves the image quality. The input MRI image is sent through a pre-processing stage in which undesired sounds and external disturbances are removed utilising filtering and ROI extraction in the proposed study [22]. Sample input image for brain MRI is shown in Fig. 2.

This is a tough task due to the tumour's irregular size, position, and shape in the brain, as well as the unequal distribution of classes in the dataset. Duplicate

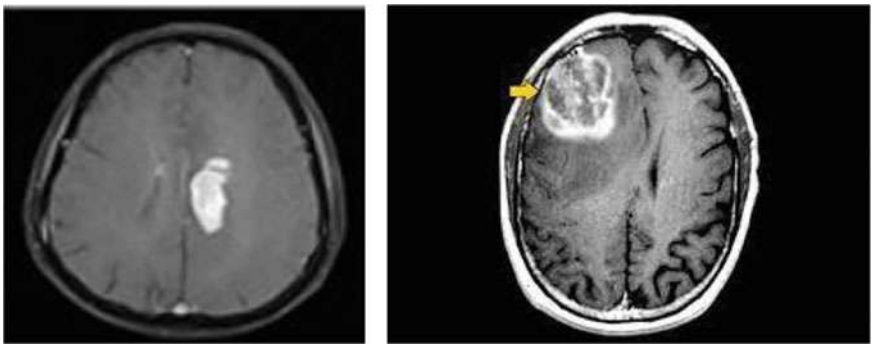


Fig. 2 Sample input image for brain MRI

information is eliminated from photos, and a region of interest (ROI) is extracted to solve these issues. The steps followed in pre-processing are

- For additional refinement, the original image is upgraded to sharpened images.
- The database's MR scanned images are transformed to greyscale images with a size of 255255.
- These images are processed to eliminate noise, and hence, visual quality of noise images is not satisfied.
- The high quality of the image is maintained, made noise free, and operated by a high-pass filter for edge detection and sharpening.

Following the application of three noise reduction methods, such as median filter, Gaussian filter, and low pass filter, the pre-processing stage includes image enhancement and skull removal, as well as selecting the best method of noise reduction based on different parameters such as MSE and peak signal to noise ratio (PSNR) [23]. Modern MRI and CT scans have extremely little noise, but due to the temperature influence on the image, it is important to eliminate noise and other artefacts in order to get quality information. Three different noise reduction strategies were used in this study: median filter, Gaussian filter, and low pass filter [24]. All three algorithms were used for image extraction, and the method that reduced precise noise was chosen based on two error metrics: peak signal to noise ratio (PSNR) and mean square error (MSE) (PSNR). PSNR is a parameter that determines how well an image is reconstructed. For picture compression, it requires both original and noisy data as input.

$$\begin{aligned} \text{MSE} &= \text{sum}(\text{sum}((\text{Input Image} - \text{Reconstructed Image}).\wedge 2)) \\ &\quad /(\text{rows} * \text{columns}) \end{aligned} \tag{1}$$

$$\text{PSNR} = 10 * \log 10\,(256 * 256/\text{MSE}) \tag{2}$$

The Gaussian filtering approach smooths out the noise in MRI pictures without sacrificing vital data in the edges and structure of the images. A Gaussian filter is a linear spatial filter that is used to reduce the picture sample's high-frequency components [25]. To achieve Gaussian smoothing, a Gaussian function is applied to an input image:

$$G(F) = G_{\sigma}(a, b) * I(a, b) \tag{3}$$

$$G\sigma(a, b) = \left(1/2\pi\sigma^2\right) e\left(-a^2 + b^2\right)/2\sigma^2 \tag{4}$$

where

I (a, b)—input image, $G\sigma$ (a, b)—Gaussian smoothing filter deviation σ, a, b represent the spatial coordinates.

* Represents the convolution operator.

The gradient operator is used to smooth the images and compute the edges in the sample image. The image is then suppressed using a Gaussian filter.

$$G(F) = \nabla(G\sigma(a, b) * I(a, b) \tag{5}$$

where

∇ represents gradient operator to estimate the directional changes in intensity values.

Some parts of an image are extracted based on predetermined criteria such as intensity information and edges in an image, amongst others. The first stage in this technique is to create an initial seed point, which will be used to extract regions with similar intensity values that are associated to that seed point. Seeds are pixels or groups of pixels that pertain to the region of interest [26]. The second stage examines homogenous pixels in tiny neighbouring regions and adds selected pixels to the increasing region based on the results. The previous process is repeated until the increasing region is depleted of pixels. Finally, all of the pixels contributed to the increasing zone are used to detect the item. However, this method has drawbacks in that each region of interest must be extracted using seed points, which necessitates manual intervention. Region of interest depicted in Fig. 3. Another disadvantage of region growth is that it is susceptible to noise, and its reliance on homogeneity requirements may result in the extraction of non-interesting regions [27].

3.3 Segmentation

In the diagnosis of a brain tumour, pinpointing the exact site is critical, since it aids in determining the tumour's shape and size. Image segmentation is crucial in brain tumour detection strategies. The process of segmenting separate tumour locations

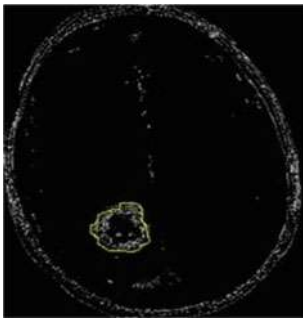


Fig. 3 Region of interest (ROI)

from an MRI scanned sample image is known as segmentation. Because the borders define each homogenous region or collection of contours, the segmentation results are provided by the picture label. Segmentation is the process of classifying MRI data into specific tissue types, as well as identifying and describing specific anatomical features [28]. To remove a tumour from a brain MRI, a variety of image segmentation approaches are applied. To offer exact information, GM, WM, CSF, and other soft brain structures are segmented. Manual segmentation and automatic segmentation are the two types of segmentation procedures [29]. Manual segmentation is time-consuming and requires experience or specialist knowledge of the human brain. Automatic segmentation, on the other hand, relies only on a histogram based on pixel intensity [30].

Three tissue types are identified in MRI scans of the brain:

- White matter (WM)
- Grey matter (GM)
- Cerebrospinal fluid (CSF)

The term "segmentation" refers to the process of breaking photographs into meaningful sections and separating valuable objects from the backdrop. Segmentation is performed in this study utilising the region growing algorithm and then the modified Battacharya distance [31].

Region Growing Algorithm

Region growth is a method of image segmentation that divides pixels in a large image into smaller regions or subregions based on specified parameters. The procedure of segmenting by region is relatively simple and noise resistant. It is often referred to as a pixel-based image segmentation technique because it involves selecting starting seed points from the image [32]. By observing neighbouring pixels, this method of segmentation determines which region initial seed points belong to. Repeat until the image has been separated into pieces (refer Fig. 4).

Implementation steps are as follows:

Step 1: choose a set of starting seed pixels that will allow us to accurately depict the desired region;

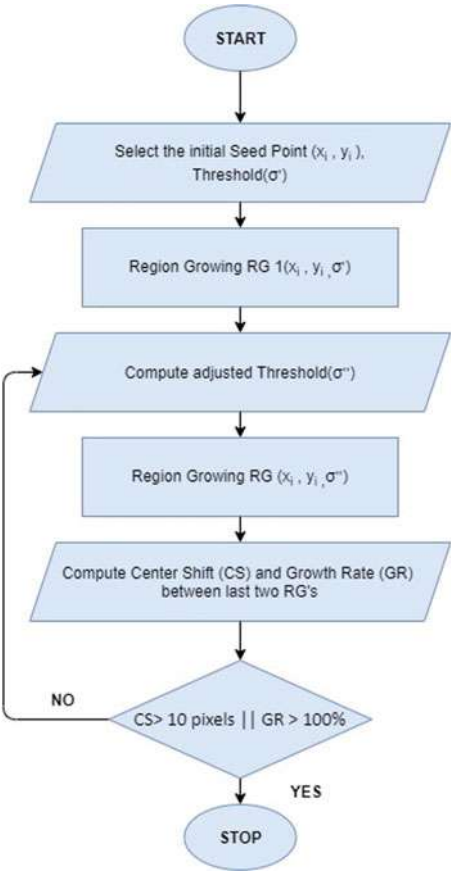


Fig. 4 Flowchart of region growing algorithm

Step 2: Select the criteria for including nearby pixels in the growing process.

Step 3: Choose conditions that will bring the growing process to a halt.

The primary principle behind region growth is to group similar pixels together to form an area. Choose a seed point in the image to be divided as the growth starting point, then choose a neighbouring pixel with comparable characteristics and meets the growth requirements, and lastly merge the pixels that meet the requirements with the seed point region. A new seed point is built using the newly merged pixels. Until there are no more pixels to combine, the cycle of seeking and merging continues indefinitely. The pixel similarity function is now denoted by the letter F. The region in k steps is denoted by Rk, and the regional characteristic function of RK is denoted by $F(\mathrm{RK})$. Similarly, the letters $s(i, j)$ stand for the current pixel, and the eigenvector of $s(i, j)$ is $F(s)$. By comparing the normal distance between two points $F(\mathrm{RK})$ and $F(s)$ and determining if they cross a specific threshold, the region's growth can be established. If $|F(\mathrm{RK}) - F(s)| T$, the pixel s is either merged $\mathrm{RK} + 1 = \mathrm{RK} + s$ or skipped, and the next adjacent pixel is searched. The formula for determining region

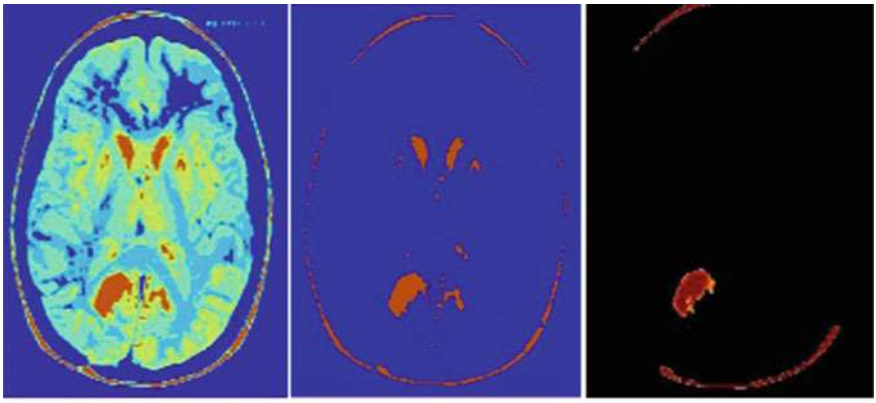


Fig. 5 Segmentation output result after applying region growing algorithm

growth is simple. The criterion is significant because the region is growing based on it. It is normal for certain pixels to not be merged into the zone due to improper threshold selection, resulting in holes. Traditional thresholding focusses mostly on grey-level data, with little regard for spatial data. Output result for segmentation after applying region growing algorithm is shown in Fig. 5.

Modified Bhattacharyya Distance

Following the completion of the region growing procedure, further segmentation is performed using the modified Bhattacharya distance. Using the modified Bhattacharya distance, the difference between the pixel values of the initial seed point and the neighbouring pixels is determined first:

$$B = \frac{1}{4}\ln\left(\frac{1}{4}\left(\frac{\eta_u^2}{\eta_v^2} + \frac{\eta_v^2}{\eta_u^2}\right)\right) + \frac{1}{4}\left(\frac{(\beta_u - \beta_v)^2}{\eta_u^2 + \eta_v^2}\right) + \frac{1}{4}\left(\frac{\partial_u - \partial_v}{\eta_u^2 + \eta_v^2}\right) \tag{6}$$

where ∂u signifies the skewness of seed point, ∂v is the skewness of neighbouring pixel, $\eta 2$ denotes the variance, and the mean is indicated as β.

Finally, the result obtained from the Bhattacharya distance is

$$S^n = \left\{S_1^n, S_2^n, \ldots\ldots, S_i^n, \ldots, S_w^n\right\} \tag{7}$$

where ith segment of nth image is denoted as S_1^n and w represents total number of obtained segments.

Deformable model with Fuzzy C-Mean clustering (DMFCM)

When used in conjunction with image processing, a deformation-based segmentation tool can help overcome some of the drawbacks of manual slide editing and traditional

image processing approaches. Fuzzy C also means that the cluster models give a concise analytical explanation of the object's shape. The functional shape of the image's subject determines the contour form (x, y)

$$\varepsilon(v) = s(v) + p(v) \tag{8}$$

where

$$s(v) = \int_{o}^{1} w_1(s)\left|\frac{\partial v}{ds}\right|^2 + w_2(s)\left|\frac{\partial v}{ds}\right|^2 \tag{9}$$

$$p(v) = \int_{0}^{1} p(v(s)) \tag{10}$$

The Euler–Lagrange equation must be satisfied by the contour $v(s)$ that minimises the energy (v), according to the calculus of variations.

$$\frac{-\partial}{\partial s}\left(w_1\frac{\partial v}{\partial s}\right) + \frac{\partial^2}{\partial s^2} + \left(w_1\frac{\partial^2 v}{\partial s^2}\right) + \nabla p(v(s,t)) = 0 \tag{11}$$

For a snake having both internal and external energy, the Lagrange equations of motion are

$$-\mu\frac{\partial^2}{\partial t^2} + \gamma\frac{\partial v}{\partial t}\frac{\partial}{\partial s}\left(w_1\frac{\partial}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2\frac{\partial^2 v}{\partial s^2}\right) = -\nabla p(v(s,t)) \tag{12}$$

It is considered to be in equilibrium when the internal and external pressures are balanced, and the shape comes to rest. (i.e $.\partial\frac{v}{\partial}t = \partial 2\frac{v}{\partial}t2 = 0$), which yields the equilibrium condition (12).

The following are steps for the fuzzy C-means clustering algorithm.

- Initialise j clusters randomly with cluster centres as $c_j = \{c_1, c_2, \ldots c_k\}$.Calculate each pixel's fuzzy membership with every centre given by Eq. 13

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_j - c_j\|}{c_i - c_k}\right)} \tag{13}$$

- Where m is a fuzziness measure, u_{ij} is an ith pixel membership with the jth cluster fuzzy cluster centres update, as shown in Eq. 14

$$c_j = \frac{\sum_{i=1}^{n}\mu_{ij}^m * x_i}{\sum_{i=1}^{n}\mu_{ij}}^m \tag{14}$$

Minimise the J objective function indicated in Eq. 15

$$J(u, v) = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^{m} * \left\| x_j - c_j \right\|^2 \tag{15}$$

Deformable clustering is applied to basic images with smooth area borders in fuzzy C models. The segmentation, visualisation, monitoring, and quantification of brain structures have all been done using this approach [33]. They have been used to detect structures in the brain.

3.4 Feature Extraction

Feature extraction in the edge region using histogram:

The frequency and directionality of brightness variations in picture data are depicted by an edge histogram. The edge information extraction process is divided into the following stages:

- A single image can be broken into 44 sub-images.
- Each sub-picture is then broken into a manageable number of non-overlapping image chunks.
- Each image block has five sorts of edges: vertical, horizontal, 45* diagonal, 135* diagonal, and non-directional.
- The relative frequency of occurrence of five types of edges in the relevant sub-images is shown in the histogram for each sub-image. Normalised sub-images confirm and organise the sub-images of the picture blocks. The binary representations are represented by the normalised bin value.

Grey-Level Co-occurrence Matrix (GLCM)

In an image, the GLCM assesses the second-order statistical information of adjacent pixels (GLCM). The image computes the combined probability density function (PDF) of grey-level pairs. The following equation can be used to express it:

$$\text{PDF} = P\,\mu\,(m, n)\,(m, n = 0, 1, 2 \ldots\ldots Z - 1)$$

Where

m, n is the Grey level of two pixels, Z is grey image dimensions, P is the position relation of two pixels and PDF is Probability density function. Normally, the direction (θ) is 1350, 900, 00, 450, and distance (D) is 2, 1 are used for calculation. Using the GLCM matrix, grey-level images can be extracted from texture features. Cluster prominence, homogeneity p, energy, contrast, correlation, entropy, correlation m, correlation p, cluster shade, autocorrelation, homogeneity m, maximum probability, and dissimilarity are some of the textural features. The features are extracted and analysed using various orientations and distances from the segmented images.

Energy:

Energy is expressed as the repetition of pixel pairs of an image,

$$L1 = \sum m = 0\ N - 1 \sum L - 1\ n = 0\ \ P2\mu\ (m,\ n) \tag{16}$$

The local variations are presented in the images, and they are measured by the contrast. If the image has large variations, the contrast value is also high. It represented the two-dimensionality frequency histogram. Separate pixel pairs are allocated to each basis of the specific predefined displacement vector [34]. The complexity of images is indicated, and the probability of co-occurrences values is based on the non-uniformity images

$$L4 = \ -\sum m = 0\ N - 1 \sum L - 1\ n = 0\ \ P2\mu\ (m,\ n)\ \log\ (p\ \mu\ (m,\ n)) \tag{17}$$

The constant energy is inversely proportional to the contrast of homogeneity.

Contrast:

The contrast of a picture is used to measure the image's local alterations. It represents textural sensitivity as well as changes in intensity. The intensity contrast between the pixel and its neighbour determines the return. The contrast between the constant images is nil. It indicates that there is a lot of local variances, and the contrast has a lot of higher values. When there is a lot of grey scale difference, the contrast gets a lot bigger, but the texture gets a lot coarser. In the case of a tiny contrast value, there will be an intense texture development as follows:

$$\text{contrast} = \sum_{j=0}^{Z_g - 1} j^2 \sum |m\ -\ n| = P_d(m, n) \tag{18}$$

Correlation:

It assesses the image sample's grey tone linear dependencies. The value of each feature varies from −1 to 1. It denotes a negative or positive association, respectively. When the photos are horizontally texturized, the correlation in 0° is stronger than in the other directions.

$$C = \frac{\sum_{m=1}^{Z_g} \sum_{n=1}^{Z_g} (m - \mu_m) P_d(m, n)}{\sigma_m \sigma_a} \tag{19}$$

where μ_m, μ_n—mean, C—correlation, Σ_m, σ_n—standard deviations of P_d (m) and P_d (n).

Cluster shades (Cs):

The cluster shade is the skewness or lack of symmetry of the matrix. The image is asymmetric when the cluster shade is large, but symmetric when the cluster shade is small.

$$C_s = \sum_{x=0}^{Z-1} \sum_{y=0}^{Z-1} (x + y - \mu_a - \mu_b)^3 p(x, y) \tag{20}$$

where

C_s—cluster shades, x, y—values of two grey levels, μ_a μ_b—mean, $p(x, y)$—probability of grey level.

The GLCM is a statistical texture analysis method that takes into consideration pixel spatial correlations. The GLCM functions calculate the frequency with which specified pairs of pixels in an image appear in a specific spatial relationship, generate a GLCM, and extract statistical measures from this matrix to characterise the texture of an image. The grey comatrix function in MATLAB estimates the frequency with which an intensity (grey-level) value I appears in a given spatial link to a pixel with a value j, resulting in a grey-level co-occurrence matrix (GLCM) [35]. The pixel of interest and the pixel to its immediate right (horizontally adjacent) are the default spatial relationship between the two pixels, but you can change it. The total number of times the pixel with value I appears in the spatial relationship to a pixel with value j in the input image is represented by each element $I\ j$) in the output GLCM.

A GLCM matrix has the same number of rows and columns as the image's grey levels, G. $P\ I\ j|\ x, y$) in the matrix represents the relative frequency separated by a pixel distance (x, y). The second-order probability values for changes between grey levels I and j at distance d at a given angle are stored in the matrix element $P\ (i, j|d,)$. G is the number of grey levels employed, and P as well as the x and y means of Px and Py are all obtained from GLCM. The ith entry of $P(i, j)$ is formed by adding the rows: $Px(i)$.

$$Px(i) = \sum_{j=0}^{G-1} P(i, j) \tag{21}$$

And

$$Py(j) = \sum_{i=0}^{G-1} P(i, j) \tag{22}$$

$$x = \sum_{i=0}^{G-1} i\, Px(i) \tag{23}$$

And

$$y = \sum_{j=0}^{G-1} j P y(j) \tag{24}$$

3.5 *Classification*

During the classification, more precision and information from each class are gathered. The ultimate result is influenced by the user's choice of classes. Deep belief network (DBN) and deep convolutional neural network (DCN) are used to classify the data (DCN). CNNs employ a multilayer perceptron variation that is supposed to need little pre-processing. Pooling CNNs, convolutional CNNs, and employing DBN for classification are two types of CNNs. Known kernels are used to engage the existing layer's feature maps, which are then combined with the activation function to determine the feature map's outcomes at a convolutional layer [36]. The newly implemented Bir cat optimization technique is used in the classification stage, which is a deep learning concept that uses a deep belief network to train the neural network.

Cat Swarm Optimization (CSO)

In 2006, Taiwanese academics Chu et al. created CSO, a heuristic global optimization approach. CSO was created by imitating a cat's behaviour [37]. Cats have been seen to spend the majority of their time observing their surroundings rather than hunting. They alternate between travelling slowly and remaining immobile in the lead up to the hunt. The searching mode is what it is called. The tracing mode is a distinct submode. The CSO technique relies on the cooperation of these two governments to identify the optimum answer. Each cat moves at its own pace and in its own territory. The number of cats in a cat group that go into seeking mode and how many go into tracing mode is determined by MR. The cat's mode is indicated by the flag. A fitness value FS, which indicates the cat's ideal position as well as its accommodation to the fitness function, is the best answer. The algorithm's phases are as follows:

Algorithm-Cat Swarm Optimization

- Step 1: D-dimensional co-ordinate values are assigned to each cat in the initial population.
- Step 2: Set the rate at which the random placements in each dimension are created.
- Step 3: The population is randomly divided into seeking and tracing modes using the mixture ratio MR.
- Step 4: Do the necessary position update on the cat based on the cat's flag bit.
- Step 5: Evaluate and track the importance of each cat's fitness function and keep the most fit cat.
- Step 6: If the conditions are met, the algorithm will complete; otherwise, it will return to step three.

Bird Swarm Algorithm

Theory of the Bird Swarm Algorithm: The following idealised principles can be used to simplify the aforementioned social behaviours.

- Each bird has the capacity to go from alert to foraging mode. A stochastic process is used to model a bird's decision to forage or remain vigilant.
- Each bird can record and update their previous best experience with the food patch right away, as well as the swarms' prior best experience, whilst foraging. Food hunting can also benefit from this information. The swarm's social information is quickly distributed.
- Each bird would try to fly towards the swarm's centre whilst remaining attentive. This behaviour may be influenced by swarm competition-induced interference. Birds with larger reserves are more likely to be discovered near the swarm's centre than birds with smaller stores.
- Birds would frequently migrate from one location to another. When flying to a new site, birds regularly switch between producing and scrounging. A producer is the bird with the most reserves, whereas the bird with the fewest reserves is a scrounger. Other birds with an intermediate reserve would choose between being a producer or a scrounger at random.
- Producers are constantly on the lookout for new sources of food. Scroungers would randomly follow a producer in search of food.

Deep belief networks

A deep belief network is a deep learning network made up of numerous RBMs that are stacked on top of each other [37]. A stochastic recurrent neural network with a visible unit layer (v) and a binary hidden unit layer (b) is referred to as an RBM (h). In their joint configuration, the total energy of the visible and hidden units (v, h) is given by

$$E(v,h) = -\sum_{i,j} v_i h_j W_{ij} - \sum_i v_i a_i - \sum_j h_j b_j \tag{25}$$

The visible layer indices are I, the hidden layer indices are j, and the weight relationship between the visible and jth hidden unit is $w_{i,j}$. Additionally, v_i and h_j denote the states of the ith visible and jth hidden unit, respectively, whilst a_i and b_j denote the visible and hidden layer biases.

To abstract hierarchical qualities generated from the raw input data, the training technique is greedy layer-by-layer, with weight fine-tuning. DBN was designed to duplicate a perceived distribution throughout the input and concealment layers' space by using direct connections between lower layer nodes and indirect connections between upper layer nodes. Layer-wise training is also carried out at the same time, with weight parameters changing via contrastive convergence to produce a balanced estimate of learning probability (CD). To learn abstract features that are durable and invariant to transformation, noise, and other influences, the conditional probability distribution of the input samples is established. Figure 6 shows a model for deep belief network model.

Classification using DCNN

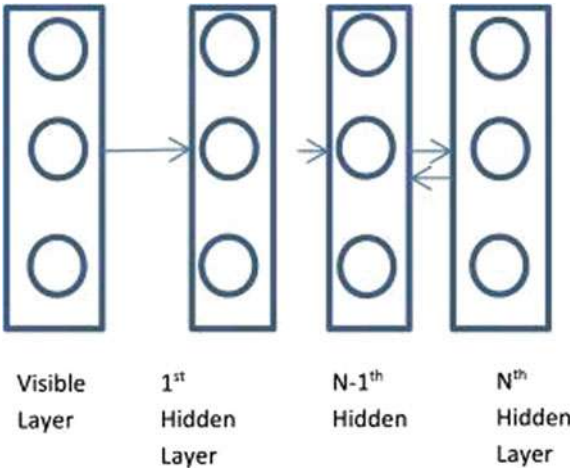


Fig. 6 Deep belief network model

This section discusses how the suggested method may be used to detect brain tumours, and how the feature vector can be used to do so. The proposed training strategy is used to train the classifier, which uses the deep convolution neural network (DCNN) to categorise the recovered features [38] as shown in Fig. 7. The proposed system's purpose is to use features extracted from the input image to detect malignant regions. The method can handle a wide range of optimization problems and provide effective solutions. In addition, HHO can deal with issues including deceptive optima, local optimal solutions, and multimodality concerns in the search space.

Convolutional layers: Using the feature vector obtained from the segmented objects, the convolutional layer produces features and extracts patterns. To generate feature maps, trainable weights are utilised to connect the neurons, which are then

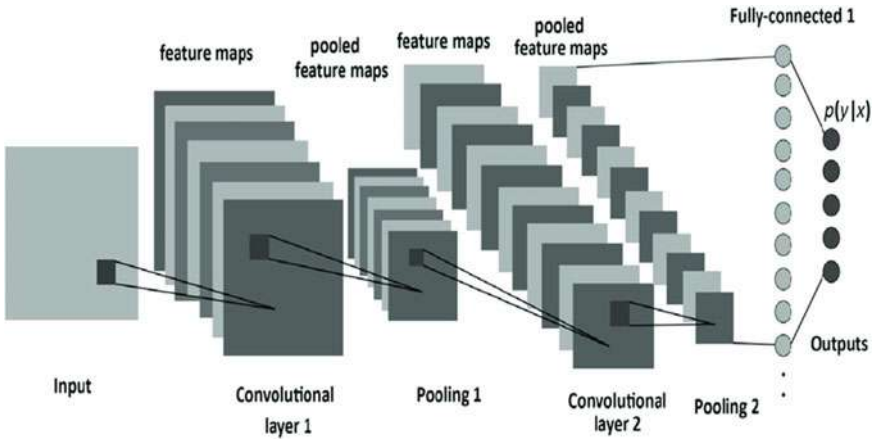


Fig. 7 Architecture of DCNN

matched with the inputs. The output is fed into a non-linear activation function, obviating the necessity for complicated functional mappings between input and response variables. The convolution layer receives the segmented image created by the sparking process, and the number of convolution layers in deep CNN is

$$T = \{T1,\ T2;\ \ldots\ldots T_{h,}\ \ldots\ldots T_i\}$$

where I is the total number of convolution layers in deep CNN and Th is the hth convolution layer. The output is given by the units at (p, w).

$$\left(T_j^h\right)_{p,w} = \left(v_j^h\right)_{p,w} + \sum_{p=1}^{w_1^{p-1}} \sum_{z=-r_1^u}^{r_1^u} \sum_{s=-r_2^u}^{r_2^u} \left(X_{f,p}^u\right)_{z,s} * \left(T_j^{h-1}\right)_{p+z,w+s} \tag{26}$$

where * defines the conv operator, which permits pattern extraction from the outputs of conv layers that are close by, and $\left(T_p^{u-1}\right)_{m+z,r+s}$ represents maps with fixed features, and $W_1{}^{p-1}$ shows how many feature maps there are in total, an $\left(X_{f,p}^u\right)_{z,s}$.

Rectified Linear Unit (ReLU) and Pooling (POOL) Layer: The activation function for testing accuracy and efficacy is known as ReLU. The ReLU layer is crucial because it allows deep CNN to cope with larger networks faster. When the feature maps are fed into the ReLU layer, the output is as follows:

$T_f^u = \text{fun}\left(T_f^{u-1}\right)$. Where T_f^u denotes the input, T_f^{u-1} denotes the output, and fun () denotes the uth layer's activation function.

Fully connected layers: To begin object categorisation, the fully connected layers accept the patterns obtained from the pooling and conv layers as input. The output of the fully connected layers is as follows:

$$S_f^u = Z\left(a_f^u\right) \text{with } a_f^u = \sum_{p=1}^{w_1^{p-1}} \sum_{m=1}^{w_2^{p-1}} \sum_{n=1}^{b_3^{j-1}} \left(V_{f,p,m,n}^u\right) \cdot \left(T_f^{u-1}\right)_{m,n} \tag{27}$$

where $V^u{}_{f,p,m,n}$ indicate the weight connecting (m, n) in pth feature map of layer $u-1$ and fth unit in layer u.

4 Evaluation

The overlap between the recommended strategy's tumour labels and the GT in three locations, namely WT, TC, and ET, was investigated qualitatively and quantitatively using the Dice coefficient to evaluate the method's segmentation performance on the training data [39]. Figure 1 displays the GT and segmentation labels obtained using

the suggested method from four patients' pre-operative multimodal baseline scans (two HGGs and two LGGs).

On the Brats and SimBrats datasets, the proposed deep CNN produced the following outcomes: (a) original images, (b) segmented images, (c) loop applied images, and (d) ground truth. The characteristics are removed, and the LOOP image (c) is shown, with white patches indicating the malignant location to be diagnosed (d).

In order to further test the efficacy of the proposed technique, the Dice coefficient was utilised to quantify the per-voxel overlap between the GT and the suggested segmentation. The distribution of cross-validated Dice coefficient scores in the training set for each region of interest (WT, TC, and ET) when only the generating phase was employed, as well as when the entire approach was used, is shown in Fig. 8. The results include all pre-operative baseline scans, pre-operative baseline HGGs, pre-operative baseline LGGs, and post-operative baseline HGGs.

When the discriminative step was used, the mean and median Dice coefficient values for all tissue labels and datasets improved significantly. Furthermore, the ET label segmentation results show a significant difference between LGGs and HGGs, with the former producing less consistent and inferior outcomes. Segmenting

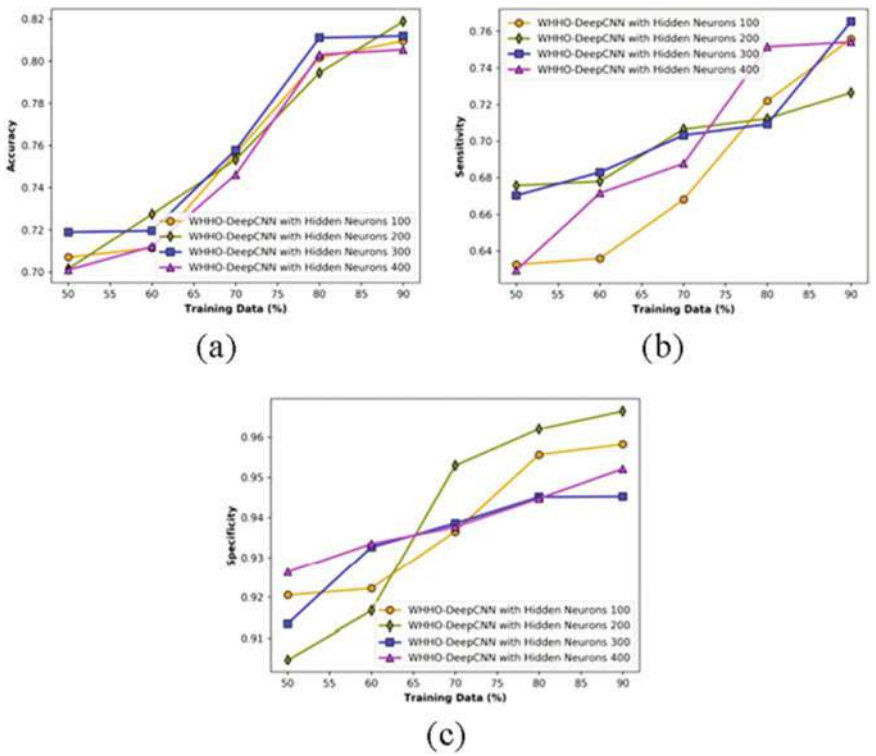


Fig. 8 Methods analysis employing a dataset in terms of **a** accuracy, **b** sensitivity, **c** specificity

enhancing tumours is more difficult since LGGs have a unique pathophysiological nature that might manifest as small, ill-defined areas of enhancement, or no enhancement at all. The LGGs' WT label segmentation, however, is comparable to the HGGs' [40] and depicted in Fig. 9.

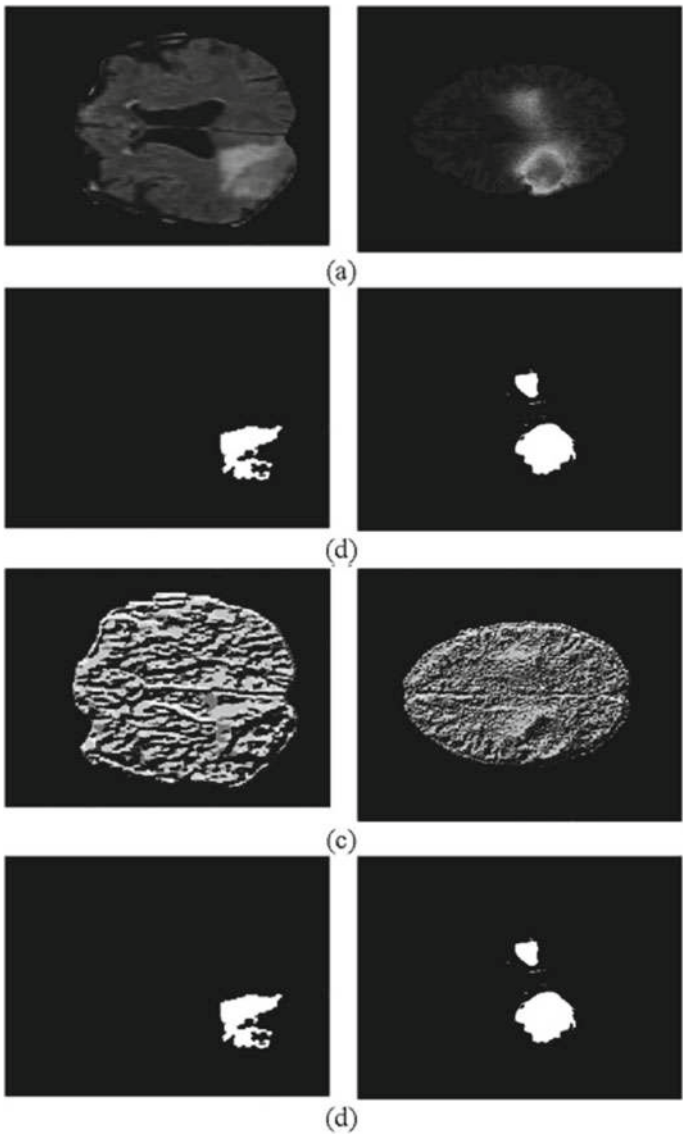


Fig. 9 Segmented output image for the given MRI image based on its ground truth: **a** original images, **b** segmented image, **c** loop applied image, **d** ground truth

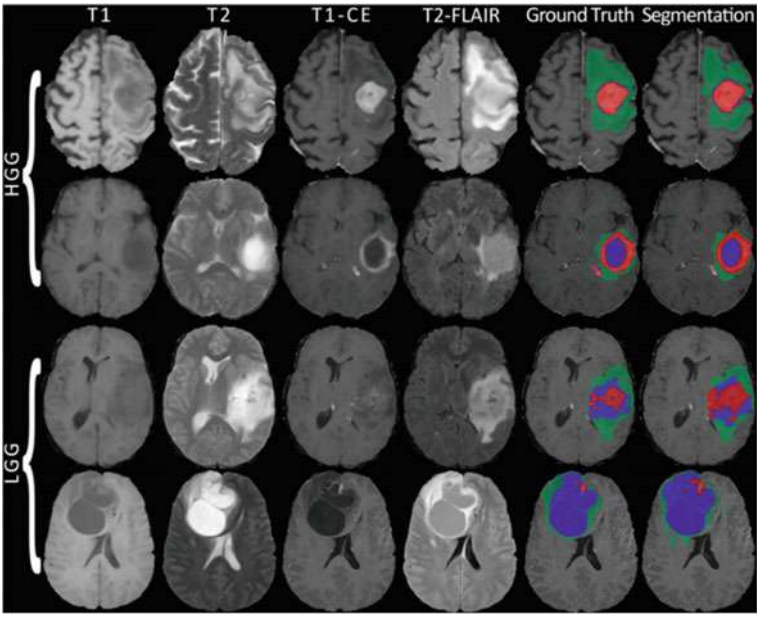


Fig. 10 Pre-operative baseline multimodal scans

4.1 Pre- and Post-operative Multimodal MRI Segmentation of Gliomas

As an example, the segmentation findings for four pre-operative baseline multimodal scans and four post-operative baseline multimodal scans are shown in Figs. 10 and 11, respectively. Edoema, enhancing tumour, and the union of necrotic and non-enhancing sections of the tumour are shown by green, red, and blue masks, respectively.

Figure 12: For each phase of the proposed approach, each tissue label, and various data groups, distributions of dice coefficients amongst patients are shown for each phase of the proposed method. The generating-only stage (G) and the whole generative-discriminative approach are depicted in green and yellow boxes, respectively (GD). The mean and median values are represented by the dot and line inside each box

4.2 Confusion Matrix and Validation Metrics

A "confusion matrix" also describes the classification model's performance (Refer Table 2). The elements of the uncertainty matrix, as follows, are

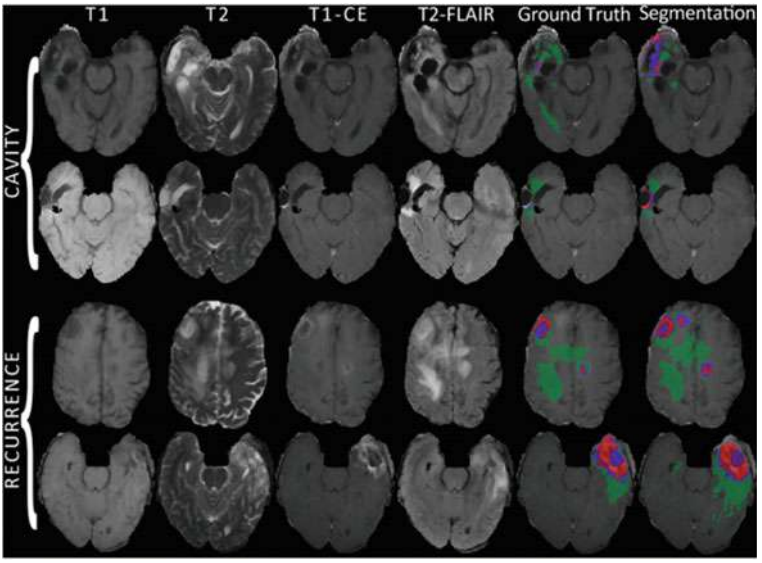


Fig. 11 Post-operative multimodal scans

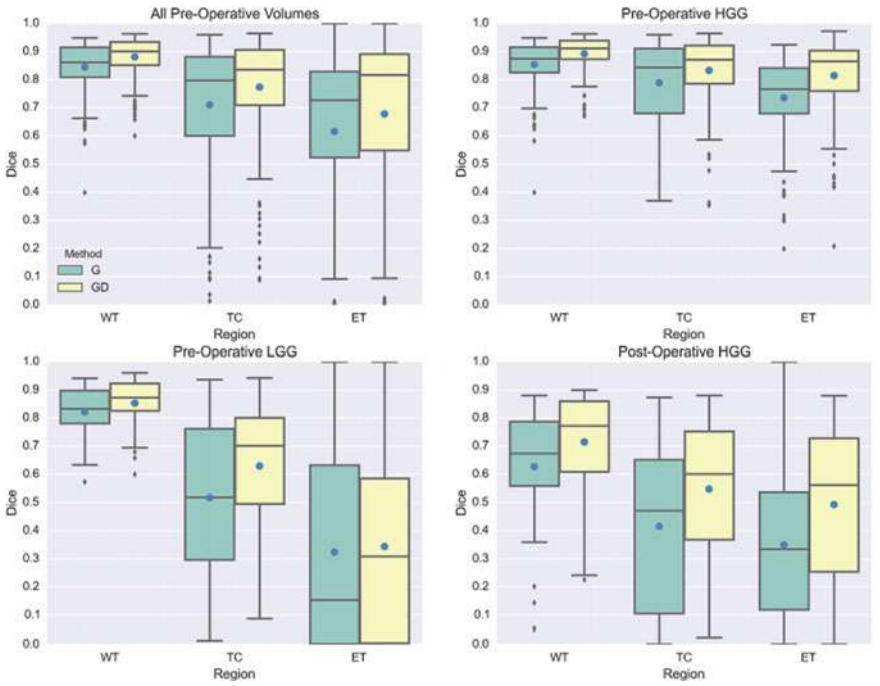


Fig. 12 Dice coefficient distributions across patients

Table 2 Performance metrics

Pre-processing	PSNR $= 10\log10\left(\frac{\text{MAX}^2}{\text{MSE}}\right)$ (28)
	$\text{MSE} = \sum_{i=1}^{K} p_i(x_i - t)^2$ (29)
	$\text{SSIM} = \frac{(x\mu_y + l_1)(2\sigma_{xy})l_2}{\left(x^2+\mu_y^2+l_1\right)\left(\sigma_x^2+\sigma_y^2+l_2\right)}$ (30)
Segmentation	Jaccard Index $= J(A, B) = \frac{S(A\cap B)}{S(A\cup B)}$ (31)
	Dice Overlap Index (DOI) $= D(A, B) = 2 \times \frac{A\cap B}{A+B}$ (32)
	Similarity Index SI $= \frac{2\times\text{Truepositive}}{2\times\text{truepositive} + \text{falsepositive} + \text{falsenegative}}$ (33)
	Absolute Volume Measurement Error (AVME) $= \left(\frac{V_{\text{automatic}}}{V_{\text{manual}}} - 1\right)$ (34)
	Figure of Merit$(\varepsilon) = 1 - (\varepsilon) = 1 - V_{\text{manual}} - V_{\text{automatic}} \vee \frac{\sum_{i=1}^{n} {\mu_{ij}}^m}{V_{\text{manual}}}$(35)
Classification	Sensitivity $= \frac{\text{TP}}{\text{TP}+\text{FN}} \times 100\%$ (36)
	Specificity $= \frac{\text{TN}}{\text{TN}+\text{FP}} \times 100\%$ (37)
	Accuracy $= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \times 100\%$ (38)
	Precision $= \frac{\text{TP}}{\text{TP}+\text{FP}} \times 100\%$ (39)
	$F-$ measure $= \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$ (40)

- TP (True Positives):
  No. of benign/malignant MR images that are marked as benign/ malignant images.
- TN (True Negatives):
  Non-benign/non-malignant MR images are known as non-benign/non-malignant images.
- FP (False Positives):
  No. non-benign/non-malignant photos that are marked as benign/malignant.
- (FN) False Negatives:
  Number of benign/malignant MR images that are marked as non-benign/non-malignant

The results as shown in the Table 3, the table portrays the comparative study of developed DBN + DCNN. The proposed model of DBN + DCNN shows that 0.957 accuracy, 0.967 specificity, and 0.918 sensitivity. From these results, the proposed model works better when compared with DBN-bird swam algorithm and DBN-Bir cat algorithm.

Table 3 Comparison of algorithms

Algorithms	Accuracy	Sensitivity	Specificity	*Image I*
DBN + DCNN	0.957	0.9673	0.918	
DBN-bird swam	0.907	0.9081	0.805	
DBN–Bir cat	0.921	0.9173	0.908	
DBN + DCNN	0.9678	0.9234	0.9679	*Image 2*
DBN-bird swam	0.9874	0.9234	0.9756	
DBN–Bir cat	0.9112	0.9156	0.9567	
DBN + DCNN	0.9234	0.9689	0.9867	*Image 3*
DBN-bird swam	0.9456	0.9112	0.9856	
DBN–Bir cat	0.9567	0.9134	0.9756	
DBN + DCNN	0.9589	0.956	0.8969	*Image 4*
DBN-bird swam	0.9073	0.9789	0.9645	
DBN–Bir cat	0.9231	0.9178	0.9786	
DBN + DCNN	0.945	0.9456	0.9856	*Image 5*
DBN-bird swam	0.9287	0.9782	0.9667	
DBN–Bir cat	0.9786	0.9785	0.9786	

5 Conclusion

This paper proposes using MR pictures to detect brain tumours using a completely automated deep CNN. The deep CNN is trained using the suggested Bir cat optimization algorithm. The Bir cat algorithm is the result of combining the bird swarm and cat swarm algorithms. This will classify tumour tissues as well as the many sorts of tissues or anomalies in the brain. On each input brain MRI image, segmentation is conducted. The relevant pixel of tumour regions aids in better segmentation results. Furthermore, statistical and texture features are used to detect brain tumours effectively. Because of the pre-processing step, the method can be tweaked to improve image quality, even when employing low-quality photographs. Other methods were outperformed by the proposed Bir cat deep CNN. Other brain tumour datasets will be used in the future to compute the efficiency of the suggested technique. In order to increase the efficiency of current approaches, advanced optimization methodologies will be researched. For both pre-operative baseline HGGs and LGGs, the generative-discriminative technique reliably segregated the whole tumour and tumour core.

The contrast between HGGs and LGGs is notably obvious in the expanding tumour zone, with LGG segmentation accuracy being lower and less constant. This discrepancy in performance implies that segmenting enhancing tumours in LGGs is significantly more challenging than segmenting enhancing tumours in HGGs. This could be due to the fact that the former has a distinct pathogenic phenotype with

small, ill-defined enhancement zones or no enhancement at all, whereas HGGs have well-defined enhancement zones.

Because of their more sophisticated character and fewer training samples, the segmentation accuracy of the post-operative HGGs was lower than that of the pre-operative baseline HGGs in all three areas of interest. Over the entire dataset, the suggested method produced high-quality segmentations for the overall tumour, tumour core, and enhancing tumour.

References

1. Deepa AR (2019) A comprehensive review and analysis on mri based brain tumor segmentation, Int J Sci Technol Res 8(10)
2. Wu W (2020) An intelligent diagnosis method of brain MRI tumor segmentation using deep convolutional neural network and SVM algorithm. Comput Math Methods Med 2020, Article ID 6789306
3. Zeng K (2016) Segmentation of Gliomas in pre-operative and post-operative multimodal magnetic resonance imaging volumes based on a hybrid generative-discriminative framework. In: Crimi A et al. (eds): BrainLes 2016. Springer International Publishing AG 2016, LNCS 10154, pp 184–194. https://doi.org/10.1007/978-3-319-55524-9 18
4. Magadza T (2021) Deep learning for brain tumor segmentation: a survey of state-of-the-art. Imaging 7(19).https://doi.org/10.3390/jimaging7020019
5. Statistics adapted from the American Cancer Society's publication (2021) Cancer Facts & Figures
6. Kulkarni SM (2020) A framework for brain tumor segmentation and classification using deep learning algorithm. (IJACSA) Int J Adv Comput Sci Appl 11(8)
7. Gupta A Applying deep learning approach for brain tumor detection. In: Materials today: proceedings. https://doi.org/10.1016/j.matpr.2020.10.063
8. Abd-EllahA MK review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. Magn Reson Imaging. https://doi.org/10.1016/j.mri.2019.05.028
9. Lathera M (2019) Investigating brain tumor segmentation and detection techniques. In: The international conference on computational intelligence and data science (ICCIDS 2019). https://doi.org/10.1016/j.procs.2020.03.189
10. Sethuram Rao G (2018) Brain tumor detection approaches: a review. In: International conference on smart systems and inventive technology (ICSSIT 2018). IEEE Xplore Part Number: CFP18P17-ART; ISBN: 978-1-5386-5873-4
11. Alam MS (2019) Automatic human brain tumor detection in MRI image using template-based K means and improved fuzzy C means clustering algorithm. Big Data Cogn Comput 3(27). https://doi.org/10.3390/bdcc3020027
12. Nimeesha KM, Gowda RM (2013) Brain tumour segmentation using K-Means and fuzzy C-Means clustering algorithm. Int J Comput Sci Inform Technol Res Excellence 3(2)
13. Aswathy S, Devadhas GG, Kumar SS (2017) MRI brain tumor segmentation using genetic algorithm with SVM classifier. J Electron Commun Eng, e-ISSN, 2278–2834
14. Aswathy S, Devadhas GG, Kumar SS (2015) Quick detection of brain tumor using a combination of EM and level set method. Indian J Sci Technol 8(34)
15. Aswathy S, Devadhas GG, Kumar SS (2019) Brain tumor detection and segmentation using a wrapper based genetic algorithm for optimized feature set. Cluster Comput 22(6):13369–13380
16. Aswathy S, Devadhas GG, Kumar SS (2020) A tumour segmentation approach from FLAIR MRI brain images using SVM and genetic algorithm. Int J Biomed Eng Technol 33(4):386–397

17. An Meir EG et al Exciting new advances in neuro-oncology: the avenue to a cure for malignant glioma. CA. Cancer J Clin 60(3):166–193
18. Bakas S et al Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci Data 4(1)
19. Khosravanian A, Rahmanimanesh M, Keshavarzi P, Mozaffari S Fast level set method for glioma brain tumor segmentation based on superpixel fuzzy clustering and lattice boltzmann method. Comput Methods Programs Biomed 198
20. Tang Z, Ahmad S, Yap PT, Shen D Multi-atlas segmentation of MR tumor brain images using low-rank based image recovery. IEEE Trans Med Imaging 37(10):2224–2235
21. Bakas S Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. Cancer Imag Arch
22. Ramli NM, Hussain MA, Jan BM, Abdullah B Online composition prediction of a debutanizer column using artificial neural network. Iran J Chem Chem Eng 36(2):153–174
23. Kamari E, Hajizadeh AA, Kamali MR Experimental investigation and estimation of light hydrocarbons gas-liquid equilibrium ratio in gas condensate reservoirs through artificial neural networks. Iran J Chem Chem Eng 39(6):163–172
24. Ganjkhanlou Y et al Application of image analysis in the characterization of electrospun nanofibers. Iran J Chem Chem Eng 33(2):37–45
25. Chen G, Li Q, Shi F, Rekik I, Pan Z RFDCR: Automated brain lesion segmentation using cascaded random forests with dense conditional random fields. Neuroimage 211:116620
26. Jalalifar A, Soliman H, Ruschin M, Sahgal A, Sadeghi-Naini A (2020) A brain tumor segmentation framework based on outlier detection using one-class support vector machine. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS, Jul 2020, vol 2020-July, pp 1067–1070
27. Torabi Dashti H, Masoudi-Nejad A, Zare F Finding exact and solo LTR-retrotransposons in biological sequences using SVM. Iran J Chem Chem Eng 31(2):111–116
28. Mitra V et al (2017) Robust features in deep-learning-based speech recognition. In: New Era for robust speech recognition. Springer International Publishing, pp 187–217
29. Zhou C, Ding C, Wang X, Lu Z, Tao D One-pass multi-task networks with cross-task guided attention for brain tumor segmentation. IEEE Trans Image Process 29:4516–4529
30. Badrinarayanan V, Kendall A, Cipolla R SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
31. Ali MJ et al Enhancing breast pectoral muscle segmentation performance by using skip connections in fully convolutional network. Int J Imaging Syst Technol
32. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J UNet++: redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans Med Imaging 39(6):1856–1867
33. Siddiqi MH, Ali R, Khan AM, Park YT, Lee S (2015) Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields. IEEE Trans Image Process. 24(4):1386–1398. https://doi.org/10.1109/TIP.2015.2405346
34. Marcheggiani D, Täckström O, Esuli A, Sebastiani F (2014) Hierarchical multi-label conditional random fields for aspect-oriented opinion mining. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol 8416 LNCS, pp 273–285
35. Ranjbarzadeh R, Saadi SB Automated liver and tumor segmentation based on concave and convex points using fuzzy c-means and mean shift clustering. Meas J Int Meas Confed
36. Karimi N, Ranjbarzadeh Kondrood R, Alizadeh T An intelligent system for quality measurement of golden bleached raisins using two comparative machine learning algorithms. Meas J Int Meas Confed 107:68–76
37. . Ravindran R, Soman KP (2009) Berkeley wavelet transform based image watermarking. In: Proceedings of the international conference on advances in recent technologies in communication and computing (ARTCom '09). IEEE, Kerala, India, Oct 2009, pp 357–359
38. Alwan IM, Jamel EM (2015) Digital image watermarking using Arnold scrambling and Berkeley wavelet transform. AlKhwarizmi Eng J 12:124–133
39. Haralick RM, Shanmugam K, Dinstein I (1973) Textural features for image classification. IEEE Trans Syst Man Cybern 3(6):610–621

Chapter 51
A GPU-Accelerated Neural Network Approach for the Diagnosis of Heart Disease in Clinical Medicine

D. Sasikala, B. Gopi, V. Sujatha, and Ravindrakumar Selvaraj

1 Introduction

In cardiology, the heart rate can be measured as the heartbeats/min. Generally, a human body absorbs the oxygen and expels the CO_2. Hence, the heart rate of a human body varied during physical exercise or sleep. Medical professionals continuously monitor this heart rate to diagnosis the health condition of a human body. This monitoring process is also held by individuals who are interested to maintain their proper health conditions. In cardiovascular system (CVS), heartbeat rate is an essential parameter. Normally, the healthy human being has the heart rate of about 72 bpm. The heartbeat slowly increased during exercise. After completion of exercise, the heart rate gradually decreased. The human body fitness is measured by this heart rate at which the pulse goes back to the normal. Bradycardia is one of the heart diseases occurred when the heart rate is lower than the normal rate. Another type of heart cardiac called tachycardia which is occurred when the heart rate is higher than the normal rate [1]. In recent years, there are many portable electronic devices developed for monitoring the ECG signals, and also these devices are capable of classifying the ECG signals. So, this type of portable device detects the heartbeat abnormalities as more as accurate [2]. The direct response of the heart's activity is generally known as the electrocardiogram (ECG) signals. These ECG signals are simultaneously monitored and recorded by using the electrocardiography [3]. The

D. Sasikala · B. Gopi
Department of ECE, Muthayammal Engineering College, Rasipuram, Tamilnadu, India

V. Sujatha
Department of ECE, Shree Sathyam College of Engineering and Technology, Sankari, Tamilnadu, India

R. Selvaraj (✉)
Department of Biomedical Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamilnadu, India
e-mail: gsravindrakumar7@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_52

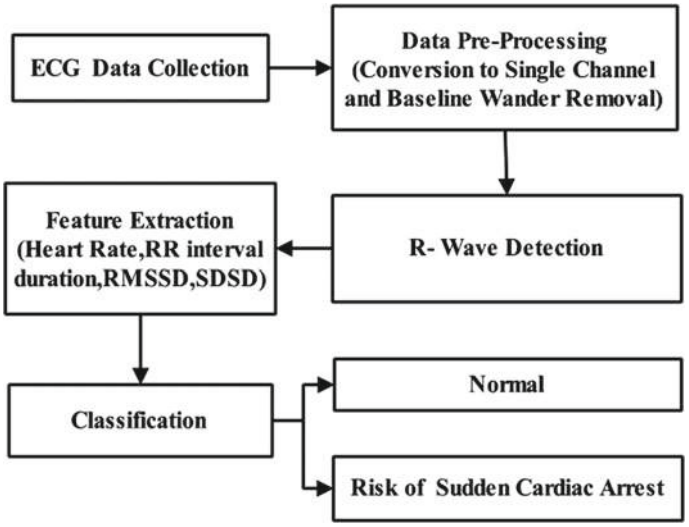


Fig. 1 General block diagram of heartbeat detection

cardiac deaths are stated from the ECG signal pattern. The treatment procedure of the different heart diseases is completely not identical to another. So, we must continuously analyze the ECG signal to automatically detect the arrhythmia type. The noises should be removed [4]. In traditional approach, the entire process of the arrhythmia detection can be broadly categorized into two stages, (that is) the initial stage is featuring extraction, and the final stage is the classification. This can be illustrated in Fig. 1. In literature for fetal preprocessing, extraction, and classification, several works have been carried out [5, 6].

1.1 Preprocessing

In most of biomedical signal analysis, preprocessing step is performed for data integration, data reduction, cleaning of data, and transferring the data. This step is purely knowledge discovery process (that is) the real-time biomedical signals are noisy in nature as well as inconsistent and incomplete. Hence, the preprocessing step is used as solution for above issues. The missing values of the ECG signals are filled by the data cleaning. At the meantime, it also performs the smoothening of the noise. This is done by outlier findings and performing a proper correction of the ECG data irregularity [7].

1.2 Feature Extraction

Feature extraction step is performed after the complete elimination of noise. In feature extraction, the important fiducial points of the ECG signal have been detected. The commonly extracted features from the ECG signal are listed as Q and S waves detection, P and T waves/peaks detection, R peaks detection, onset and offset points of P wave, onset and offset points of QRS complex, onset and offset points of T wave [8, 9].

1.3 Classification

The classification stage is used to classify the final heartbeat with the help of classifier. In ECG signal classification, there are different classifiers employed based on the extracted features. The re-sampling takes place after the completion of the segmentation process. The segmentation process consists of single or multiple classifiers. Based on the input feature vector, the classifier chooses the needed classes. As a result of the classifier, the ECG signals are categorized into abnormal and normal beats [10]. In this paper, Sect. 2 describes the previous literature survey, Sect. 3 represents the various NN architectures, Sect. 4 describes the proposed method of arrhythmia classification, Sect. 5 involves the experimental result, and Sect. 6 ends with the corresponding conclusion.

2 Literature Survey

An empirical mode decomposition (EMD)-based heartbeat measurement as well as enhancement method was introduced in literature [11]. The presented EMD algorithm is implemented in the ECG signal preprocessing stage. The fetal extraction from ECG signal was implemented [12]. It is processed in the real-time FECG signals. Here, the multi-scale DWT is used for feature extraction of the ECG signals. Another method of ECG feature extraction was developed [13]. They exploited the fast correlation-based feature selection method, and it is used remove the redundant features. This is implemented to produce the improved quality for the classification of heart diseases. They also performed variety of classification algorithms such as KNN, SVM, random forest, Naïve Bayes, and multilayer perceptron ANN.

The particle swarm optimization (PSO) combined with ant colony optimization (ACO) is employed to optimize the mentioned classifiers. Genetic algorithm-based heart disease classification was presented [14]. This genetic algorithm is combined with the classifier to perform the classification process. The main objective of the genetic algorithm is to discover the high-level prediction and its accuracy rate, high

interestingness values. The novel approach of ECG feature extraction and classification was presented in past [15]. In this method, the feature extraction consists of three important blocks. They are (i) discretization of incoming data, (ii) algorithm for feature extraction, and (iii) heuristic rough set reduction algorithm for reduction of the feature. In classifier, it employed to achieve the highest level of accuracy based on the jackknife cross-validation scheme. Similarly, there are nine types of classification algorithm evaluated in [16]. They discussed the overall function and performance of the classifiers such as SVM, KNN, Naïve Bayesian neural network, decision tree, and ANN. For heart disease detection, the support vector machine classifier is performed based on the apriori algorithm.

3 Background Methodology

A cardiovascular disease (CVD) is one of the critical diseases, and its management is to be a global imperative. The simultaneous evaluation of the cardiac condition is done in the certain scenarios like ICU monitoring. Hence, there are lot of automatic cardiac detection and alert generations developed [17]. In the recent decades, the cardiac arrhythmia has been detected with the help of deep learning methods [18].

3.1 Recurrent Neural Network (RNN)

Generally, the RNN is the improvised pattern of the feed forward neural network. The FNN is fully constructed with the lot of feedback loops. These loops produce the cyclic graph. Hence, the repeated loops are considered as the short-term memory, and they are utilized to stock up and recover the previous information in a particular time period. The temporary task performed in RNN has been successfully executed in the loop-based environment. Instead using multilayer perceptrons (MLPs), the recurrent neural network can easily perform the temporal sequences in the variety of length. Along with the time-steps, this neural network entirely shares its parameters. This sharing mechanism of the RNN is used to reduce the handling of the unseen strings of the arbitrary length. In other words, the RNN can significantly learn the dynamic temporal behaviors of the both input and output strings in the arbitrary length. For long-lasting artificial intelligence tasks, the RNN performs very extensively for modeling of languages, speech recognition applications, and machine translation, etc., [19, 20]. Equations (1)– (3) illustrate the mathematical formulation of the RNN. $w_{hx} \in R^{kxd}$, $w_{hh} \in R^{kxk}$, and $b \in R^{k}$.

$$H = Rd\mathrm{x}Rk \rightarrow Rk \tag{1}$$

$$H(x, h) = f(w_{xh}x + w_{hh}h + b) \tag{2}$$

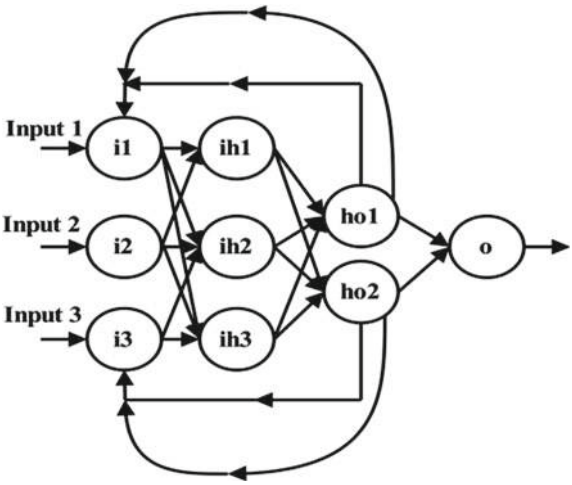


Fig. 2 RNN with feedback connections

$$w_{hx} \in R^{kxd}, w_{hh} \in R^{kxk} \text{ and } b \in R^{k} \tag{3}$$

where the input vector is represented as xt and the weight matrices are denoted as w_{xh}, w_{hh}. The RNN has the hidden layer which has the inbuilt nonlinear activation function and is denoted as f. For this nonlinear function f, sigmoid function (σ) is applied in the element wise. The short-term memory of the RNN's network is mentioned as H parameter. The RNN with the feedback loops is clearly illustrated in Fig. 2.

3.2 Artificial Neural Network (ANN)

Basically, ANN is one of commonly used mathematical model for the many data processing applications. It includes the N number of elements or units which are generally known as neurons. These neurons are arranged to form the layers. The weights and biases are used to interconnect the one layer with another layer. ANN is the most widely used NN approach in different applications. Multilayer perceptrons (MLPs) are a network which contains the N number of the neurons which are separated in the different layers. In these layers, some of the layers are allotted for the input and for output. One or more hidden layers are also available in the ANN [19] (Fig. 3).

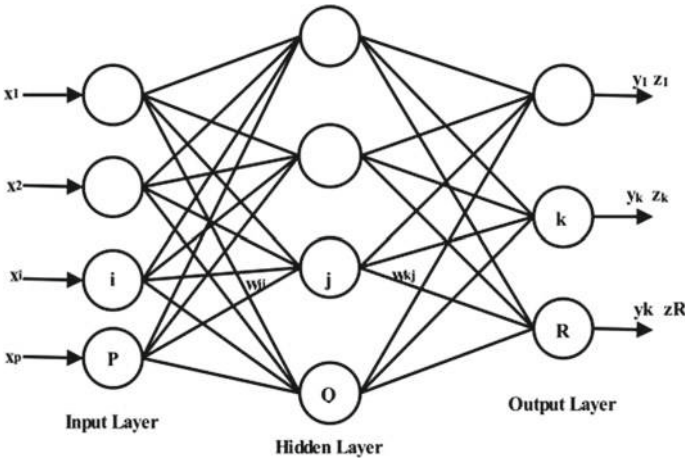


Fig. 3 Artificial neural network architecture

3.3 Convolutional Neural Network (CNN)

CNN is a unique type of the multilayer perceptrons (MLPs). Three major layers are utilized to build the CNN. They are known as pooling layer, fully connected layer, and convolutional layer. CNN accepts the time series of 1D-input data, and this 1D data is arranged in the sequential time instants order. The convolutional neural network contains the rectified linear units (ReLUs) which are generally considered as the nonlinear activation function. INPUT-CONV-POOL-FC architecture employed in the CNN where convolutional layer (CONV) acts as the primary building block. The 1D-input ECG data is represented by a input vector $x = (x_1, x_2, \ldots, x_n - 1, x_n, c_l)$. Here, x_n belongs to the features which include the time series of the both normal and abnormal ECG data, and cl belongs to the normal and abnormal class label. The 1D CNN is designed from the feature map which is denoted as the fm. It can be achieved by application of CNN in the 1D-input data with the filter w ($w \in$ Rfd, f—features from the input data and it is used to produce the output). The next block of the CNN is the pooling layer which accepts the output feature from the previous block. Here, the new set of feature map is represented as fm, and the feature f is acquired from the following Eq. (4).

$$\mathrm{hl}_i^{\mathrm{fm}} = \tan h\,(w^{\mathrm{fm}} x_{i:i+f-1} + b) \tag{4}$$

In Eq. (4), the term hl denotes the filter which is used to denote the every feature set f in the 1D-input data which is represented as ($x_{1:f}$,, $x_{2:f+1}$,…,$x_{n\text{-}f+1}$). This is employed for generating the feature map (that is) hl = (hl_1, hl_2,… $\mathrm{hl}_{n\text{-}f+1}$). Here, b belongs to the R (bias term), and hl belongs to the $R_{n\text{-}f+1}$.

As already said, the pooling layer accepts the output from the previous convolutional layer. ReLU activation function included in the convolutional layer which

uses maximum of (0, X) to every input. The term *X* denotes the ReLU. The down sampling operation performed in the pooling layer. In each and every feature map (i.e., hl = max {hl}), the max-pooling process is applied. Hence, the most significant features are attained from the max-pooling operation. Due to this, the highest value of the features also selected. These selected features are given as input to the fully connected layer. The fully connected layer consists of softmax function which produces the class probability distribution. Hence, the FC layer in the CNN calculates the final class output.

4 Proposed Methodology

The function of the human heart is directed by the electrical system. It continuously stimulates the movement of the blood through contractions. So, the irregularity of this electrical stimulations produces the disorder called cardiac dysrhythmia or arrhythmia. This cardiac arrhythmia causes the cardiac arrest. But, some of the arrhythmias are less harmful. In the United States, there are 500,000/year deaths caused by the arrhythmia. This death rate can be reduced to 25% by employing the early detection of arrhythmia and followed by proper treatments. The proposed block diagram of the arrhythmia detection is shown in Fig. 4.

The ECG arrhythmia classification is done by employing the GPU-accelerated neural network. The proposed method contains two important functional phased called feature extraction and classification phases. In the first phase, there are two main stages involved such as preprocessing stage and processing stage. The initial preprocessing stage is used for improving the classification accuracy of the neural network. Then only, it will give more accurate features. Initially, the raw 1D ECG signals are obtained from the ECG electrodes which are placed in the human body. The acquired ECG signals are corrupted from the baseline noise. Baseline wander also presented due to factors such as biological or instrument sources (respiration, skin resistance, and amplifiers thermal drift). These factors are under in low-frequency noise sources. In our proposed method, the noise present in the ECG signal is removed by using discrete wavelet transform. The DWT is constructed by employing the low-pass and high-pass filters. The smoothing response of the filter is given in the below Eq. (5).

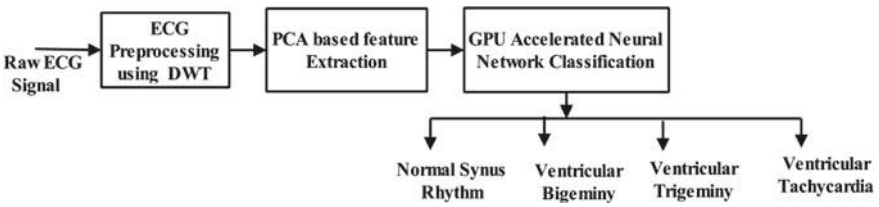


Fig. 4 Proposed block diagram for ECG arrhythmic classification

$$y(i) = (1/2N + 1)(x(i + N) + x(i + N - 1) + \cdots + x\,(i - N)) \tag{5}$$

In above Eq. (5), the $y(i)$ denoted as the smoothed value for the i-th x—input data, the number of neighborhood data is denoted as N, and the span is represented as $2N + 1$. For feature extraction, we employed principle component analysis (PCA) which is commonly used feature extraction algorithm in many signal processing applications. At first, the PCA algorithm computes the eigenvectors of a covariance matrix. The eigenvectors have the highest eigenvalues, and they are used to separate the 1D ECG data into the new subspace. These subspaces are in unity as well as it contains fewer dimensions. The PCA algorithm generally modifies the covariance matrix of N number of features into the new dataset. The extracted dataset contains the fewer features which is less than the N. In this way, the reduction of N number of feature is performed by PCA, and also it constructs the new feature set. The classification phase performs the both binary and multi-class classification based on the multilayer GPU-accelerated neural network. The output from the each neuron is known as the sigmoid function, and it is expressed in Eq. (6).

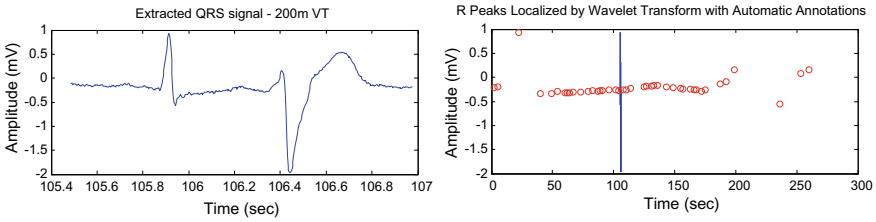
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}, x = \begin{bmatrix} x_0 \\ x_1 \\ . \\ x_n \end{bmatrix} \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ . \\ \theta_n \end{bmatrix} \tag{6}$$

In classification stage, the cost of the NN is reduced by employing the stochastic gradient descent and back propagation algorithms. There are huge number of hyper-parameters involved in the NN. The major part of this project is to offer the effects of the learning rate, size and depth of the hidden layer, depth, and regularization parameters. Hence, we briefly analyzed the performance of the neurons for each hyperparameter. If the number of neurons is raised, then there will be reduction in the performance. This is applicable to both binary and multi-class classifications. So, we employed 100 neurons in each layer. The increasing in the hidden layer numbers will drastically reduce the performance of the NN. Then only, we choose two hidden layers for our proposed method.

5 Experimental Results

5.1 Database

We collect the ECG database from the MIT-BIH arrhythmia database [21]. It contains the 25 complete records and 48 records with the reference annotations. These are freely accessible in PhysioNet's inception. Out of 48, the remaining 23 files are available for MIT-BIH arrhythmia database. It contains the digital recording of 360 samples/second/channel. It has the resolution of 11 bit/ 10 mV range. Each record



(a) (b)

Fig. 5 **a** QRS extraction, **b** *R* peak extraction of ECG by single feature extraction method

is annotated by the multiple cardiologists. The computer-based readable reference annotation is obtained after the resolving of many disagreements. In this database, each beat includes the approximately 110,000 annotations.

5.2 Feature Extraction Phase

The incoming ECG data is preprocessed to remove the unwanted baseline noises. After the preprocessing stage, feature extraction method was performed by means of extracting single and multi-features of the tested ECG database. Figure 5a, b shows the single QRS and R peak extraction of the ECG data. Figure 6a–f shows the multiple QRS extraction, and multiple *R* peak extraction is shown in Fig. 7a–f.

5.3 Classification Phase

Proposed classification phase employs GPU-accelerated neural network. The architecture of the NN is shown in Fig. 8. Table 1 shows the classification of arrhythmia for different samples. The NN performed under *N* type of feature vectors. NN performance was validated by kappa statistic, MSE, RMSE, relative absolute error, and root relative squared error. These parameters are given in Table 2.

The NN performance curve is shown in Fig. 9 which involves the two hidden layers and two phases such as training and testing. The best outcome is also plotted in the Fig. 9 which produces the relationship between the epoch of 542 samples and its corresponding mean square error.

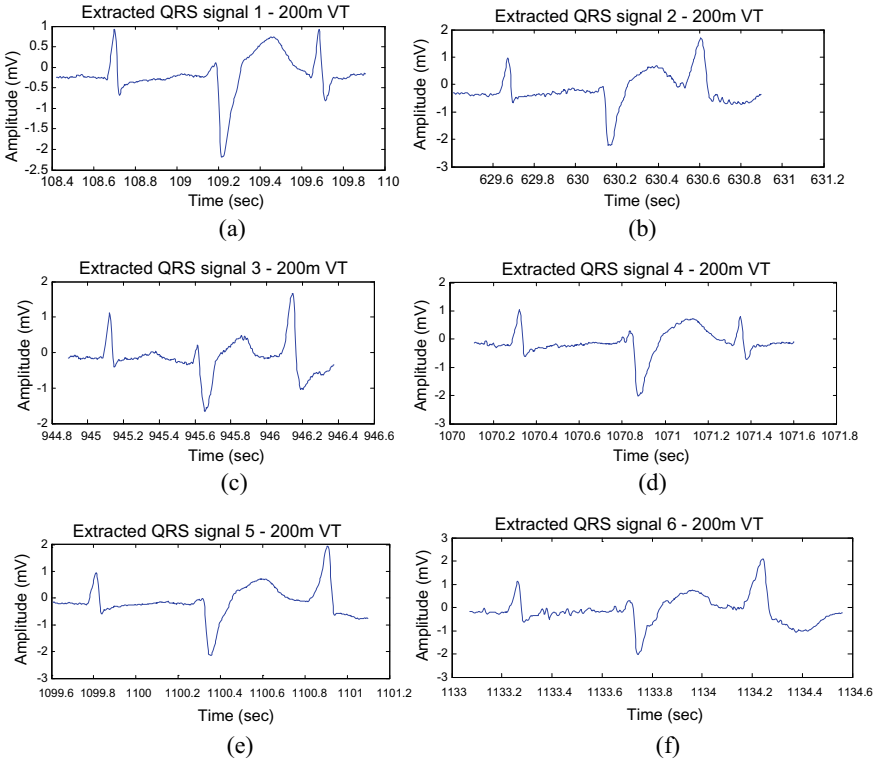


Fig. 6 **a–f** QRS extraction of different ECG signal

6 Conclusion

In this paper, we proposed GPU-accelerated neural network for ECG arrhythmia classification. The proposed classification algorithm performed in the real-time MIT-BIH arrhythmia database. Our proposed method involves two main phases such as feature extraction and classification phase. In feature extraction phase, both single and multiple QRS and R peaks are detected by using the PCA. Arrhythmia type can be classified in the classification phase. The classifier outperforms the convention classifiers such as ANN, RNN, and CNN and can be verified by measuring MSE and RMSE parameters.

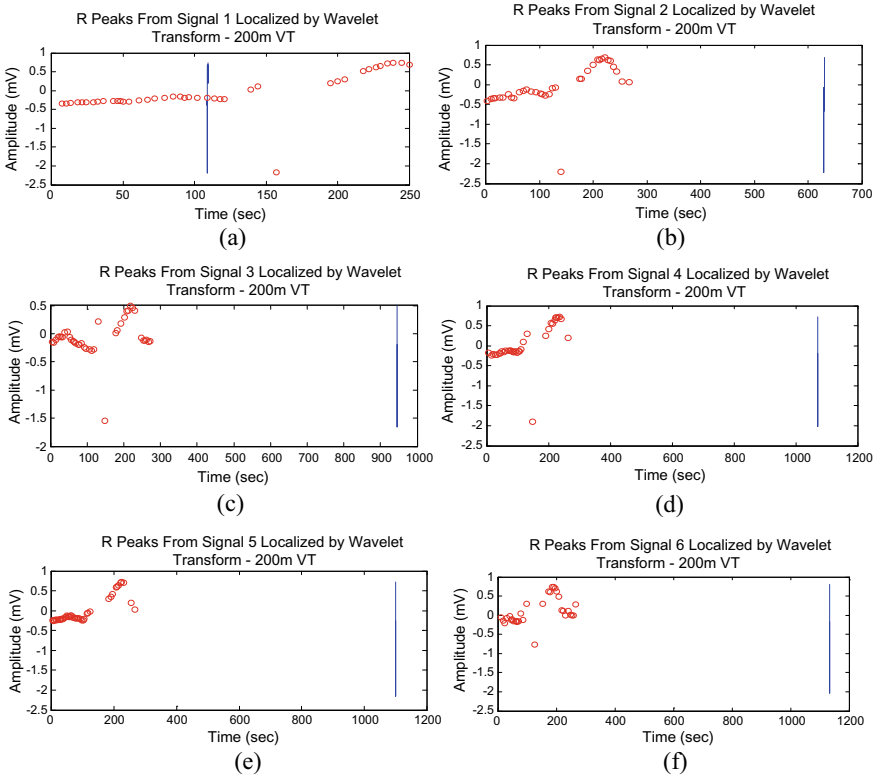


Fig. 7 **a–f** *R* peak extraction of different ECG signal

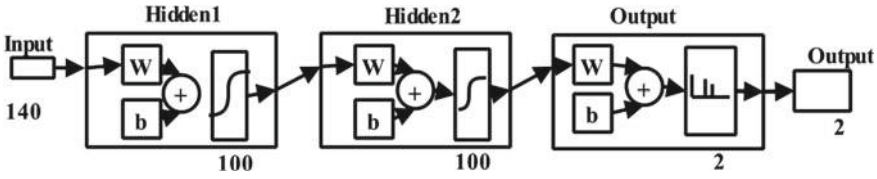


Fig. 8 Neural network model for classification phase

Table 1 Classification of the cardiac arrhythmia

ECG samples in MIT-BIH arrhythmia database	Cardiac type
103	Normal Synus Rhythm
106	Ventricular Bigeminy
119	Ventricular Trigeminy
200	Ventricular Tachycardia
208	Ventricular Trigeminy
210	Ventricular Bigeminy
215	Ventricular Tachycardia
233	Normal Synus Rhythm

Table 2 Performance measures of the NN

Parameter	Value
Accuracy	96.1702%
Kappa statistic	0.8038
Mean absolute error	0.0239
Root mean squared error	0.1234
Relative absolute error	24.3767%
Root relative squared error	56.0492%

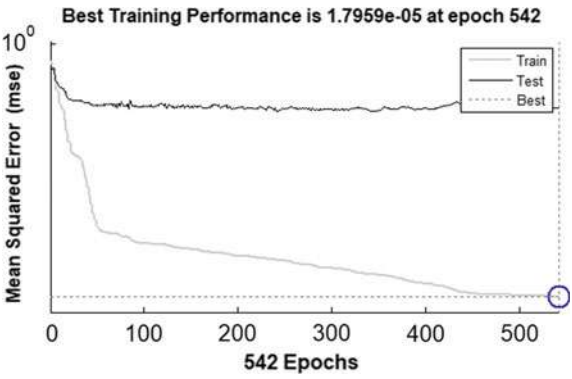


Fig. 9 Performance curve of the NN

References

1. Zeba K, Patil LS, Gowda SR, Varsha R, Shobha Chandra K (2018) Real time heart attack and heart rate monitoring android application. Int J Comput Sci Mobile Comput 07(04):115–124
2. Al-Masri E, Momin M (2018) Detecting heart rate variability using millimeter-wave radar technology. In: 2018 IEEE international conference on big data (big data), Seattle, WA, USA, pp 5282–5284
3. Ravindrakumar S, Bommannaraja K (2010) Fetal ECG extraction and enhancement in prenatal monitoring—review and implementation issues. In: IEEE: Trendz in Information Sciences & Computing(TISC2010) Conference Proceedings

4. Ravindrakumar S, Bommannaraja K (2016) A multi-stage adaptive singular value decomposition approach for fetal ECG signal extraction in multichannel input system for prenatal health monitoring. Asian J Inf Technol
5. Ravindrakumar S, Bommannaraja K (2015) Certain investigation on denoising the multichannel abdominal ECG signal using various adaptive noise suppression techniques. Aust J Basic Appl Sci 9(5):372–380
6. Haque A (2015) Cardiac dysrhythmia detection with GPU-accelerated neural networks. In: Conference proceedings, pp 1–5
7. Bhuvaneswari Amma NG (2012) Cardiovascular disease prediction system using genetic algorithm and neural network. In: 2012 International conference on computing, communication and applications, Dindigul, Tamilnadu, pp 1–5
8. Chandra, Sharma S, Singh A, Kumar G 2018) Feature extraction of ECG signal. J Med Eng Technol 42(04):306–316
9. Nithya D, Ravindrakumar S (2012) Detection of cardiovascular abnormalities using peak detection and adaptive thresholding: a synthetic and real time approach. In: 2012 international conference on computing, communication and applications, Dindigul, Tamilnadu, pp 1–5
10. Houssein EH, Kilany M, Hassanien AE, Snasel V (2016) A two-stage feature extraction approach for ECG signals. In: Proceedings of the third international Afro-European conference for industrial advancement—AECIA, pp 299–310
11. Norouzi N, Aarabi P (2013) Multi-channel heart-beat detection. In: 2013 IEEE global conference on signal and information processing, Austin, TX, pp 739–742
12. Desai KD, Sankhe MS (2012) A real-time fetal ECG feature extraction using multiscale discrete wavelet transform. In: 2012 5th international conference on BioMedical engineering and informatics, Chongqing, pp 407–412
13. Khourdifi Y, Bahaj M (2019) Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization. Int J Intelligent Eng Syst 12(1):242–252
14. Akhil jabbar M, Chandra P, Deekshatulu BL (2012) Heart disease prediction system using associative classification and genetic algorithm. In: International Conference on Emerging Trends in Electrical, Electronics and Communication Technologies-ICECIT, pp 1–11
15. Liu X, Wang X, Su Q, Zhang M, Zhu Y, Wang Q, Wang Q (2017) A hybrid classification system for heart disease diagnosis based on the RFRS method. In: Computational and mathematical methods in medicine, pp 1–11
16. Sowmiya C, Sumitra P (2017) Analytical study of heart disease diagnosis using classification techniques. In: 2017 IEEE international conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Srivilliputhur, pp 1–5
17. Oksuz I, Ruijsink B, Puyol-Antón E, Clough JR, Cruz G, Bustin A, Prieto C, Botnar R, Rueckert D, Schnabel JA, King AP (2019) Automatic CNN-based detection of cardiac MR motion artefacts using k-space data augmentation and curriculum learning. Med Image Anal 55:136–147
18. Bai M, Xu Y, Wang L, Wei Z (2018) A deep multi-scale convolutional neural network for classifying heartbeats. In: 2018 11th international congress on image and signal processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, pp 1–5
19. Jadhav SM, Nalbalwar SL, Ghatol AA (2012) Artificial neural network models based cardiac arrhythmia disease diagnosis from ECG signal data. Int J Comput Appl 44(15):8–13
20. Swapna G, Soman KP, Vinayakumar R (2018) Automated detection of cardiac arrhythmia using deep learning techniques. In: International Conference on Computational Intelligence and Data Science (ICCIDS 2018), pp 1192–1201
21. Moody GB, Mark RG (2001) The impact of the MIT-BIH Arrhythmia database. IEEE Eng Med Biol 20(3):45–50 (PMID: 11446209)

Chapter 52
Spatial Analysis of Forest Health Dynamics Through Google Earth Engine Cloud in Similipal Tiger Reserve, Odisha, India

Kishore C. Swain, Chiranjit Singha, and Sanjay Kumar Swain

1 Introduction

Remote sensing-based satellite technologies are widely accepted and verified approaches for forest ecology management of their temporal capability [1]. Google Earth Engine (GEE) cloud could be the superior guide for monitoring the forest ecosystem dynamics in a time-bound manner [2], with higher spatial resolution. Geospatial domain GEE is a web-based API with big remote sensing data storage ability for parallel computing without any processing fees [3–5]. Long-term Landsat satellites archive data sets are continuously used to monitor the forest health dynamics such as forest growth, forest degradation, climate change impact, and time series plant phenology [6–8]. Harmonic analysis decomposes a time dependent periodic phenomenon in to a series of sinusoidal wave functions, which has ample applications beyond mathematics. The harmonic optimization model based on Landsat data is found suitable for assessing the forest type, forest change, and recovery pattern through satellite-derived green proxies [9, 10]. Mathematical harmonics regression was used to show the phenology trend of year-to-year noise intrinsic inside the time series spectral images [11]. Wilson et al. [12] further used Landsat derived tasseled cap transformation (TCT) time series to correlate with the forest land cover dynamics through Fourier Harmonic regression in the USA during the years 2009–2013. Pasquarella et al. [13] reported that time series harmonic regression reduces noise in spectral response from the seasonal forest behavior. Harmonics functions are very useful for cloud-free Landsat NDVI time series data to effectively predict the seasonal forest phenology patterns [14]. Long-term time series NDVIs through harmonics functions are a good predictor for tree canopy and croplands

K. C. Swain · C. Singha (✉) · S. K. Swain
Department of Agricultural Engineering, Institute of Agriculture, Visva-Bharati, Sriniketan, West Bengal 731236, India
e-mail: singha.chiranjit@gmail.com

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_53

cover mapping [15, 16]. Harmonic metrics quantifying the phenological dynamics of the tree/grass coexistence and competition [17].

NDVI has a good predictor for analysis of the relationship between precipitation distribution and vegetation productivity. Fernández-Manso [18] showed a clear dependency between the precipitation level and NDVI in conifers forest through climate change analysis in Spain. Clinton et al. [19] investigated global phenology responses with the precipitation variability using the autocorrelation method in calendar years of 2008–2011. AVHRR or MODIS-derived NDVI products have been great potential for identified affected vegetation phenology behavior by climate change [20–22]. The autocovariance of a time series lagged collection is very useful for understanding and predicting forest health dynamics [23]. Hong et al. [24] computed the lagged collection of the Enhanced Vegetation Index (EVI) for cumulative precipitation estimation from earth observation using artificial neural networks (ANN PERSIANN) gridded rainfall to inspect the interface between precipitation and NDVI for predicting ecological process impact of an exotic species. In this study, we applied an automated structure for normalizing Landsat 7/8 imagery, of annual cloud-free composites of spectral indices across the Landsat archive. Harmonic model was executed with a python API library in the GEE cloud computing platform for forest change study. It carries out a spatial analysis of linear modeling of smoothing, interpolation, and phenology patterns.

The purpose of this study is to create a forest attribute through long-term time series data of stacked Landsat images to analyze the variation of forest cover. Phenology response to climatic variables such as NDVI and precipitation were measured by the covariance and correlation analysis. Multitemporal data described the per pixel basis seasonality phenological variation trend of the three different forest categories such as dense, moderately dense, and open forest in Similipal Tiger Reserve forest, Odisha, India during 2000-2019.

2 Materials and Methods

2.1 Study Area

Similipal Tiger Reserve is a large protected area situated in the Mayurbhanj district of Odisha, India (20° 17′–22° 34′ North latitude and 85° 40′–87° 10′ East longitude) located between the Central Indian forests and the Northern Eastern Ghats (Fig. 1). UNESCO (2009) stated that the STR region included as World Network of Biosphere Reserves being a key biodiversity hotspot. The total study area was covered by 2707 km^2 with average height of 559.31 m from MSL. This region experiences a tropical monsoonal-type climate. Summers season temperatures are around 40 °C, whereas the winter temperatures are as low as 14 °C. The maximum rainfall occurs from June to October in the range of 2000 mm. Similipal regions have an outstanding rich floral and faunal diversity. Sal is the dominant tree species in this region with tropical moist

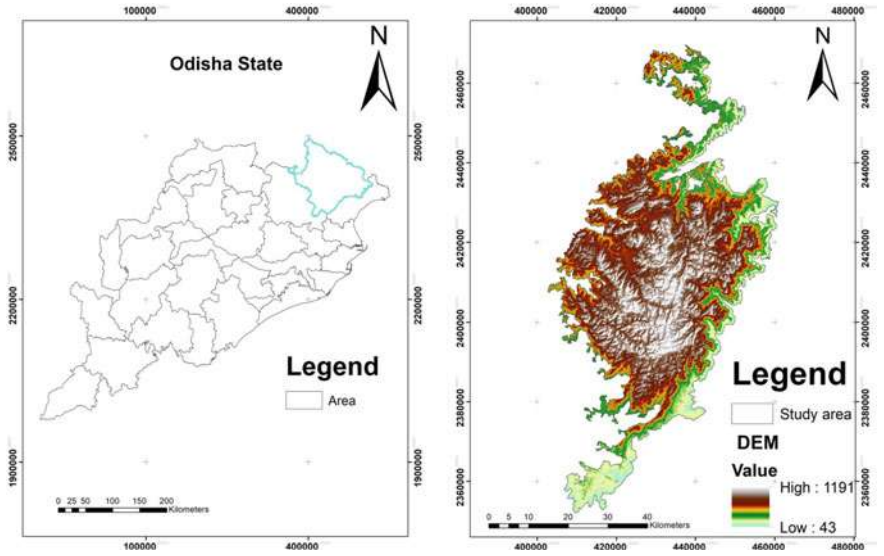


Fig. 1 Study area location

deciduous and semi-evergreen forest. The Government of India assessed that nearly 274 km^2 of dense forest lost over 20 years. According to FAO reports, the annual rate of deforestation is 1.577% yr^{-1} in south Asia region.

2.2 Landsat Image Processing

All Landsat 7/8 Tier 1 surface reflectance images (<20% cloud cover) were acquired from Google Earth Engine (GEE) data catalog for the entire Similipal forest area during 2000–2019. Quality Assessment (QA) Band was used for the cloud masking of all Landsat images because most (60–75%) images had cloud cover in May to November which is partially reduced to 30–40% during from December to April in the entire region. We retrieved a total of 1056 Landsat images to extract the phenological characteristics of the three different forest classes such as dense, moderate, and open forests. Vegetation distribution map was developed per pixel NDVI data through GEE Python API (Eq. 1).

$$\text{NDVI} = (\text{NIR} - R)/(\text{NIR} + R) \tag{1}$$

where NDVI—Normalized Difference Vegetation Index, R—Red band, NIR—Near Infrared.

2.3 Harmonic Regression from Landsat Time Series

Google Earth Engine (GEE) workflow is very useful to regulate the harmonic regression coefficients with the NDVI time series data. This variable is derived by stacked of multitemporal raster through chronologically ordered images. The fast-computing GEE platform was used to extract the spectral characteristics of different forest land-cover from the harmonic regression model. Harmonic regression algorithms estimated NDVI fitted and original time series values by Landsat collections from 2001 to 2019. According to Shumway and Stoffer [25] mathematical Harmonic model is used to decompose a multitemporal, stationary indicator into a sequence of specific cosine or sine waves, individually considered by a definite amplitude and phase. The harmonic function is calculated by the linear regression reducer to get the phase and amplitude of the NDVI time series [26]. Temporal behavior of the pixel in the NDVI profile describing by the linear model (Eq. 2). This model is represented by reducing stationarity and detrending of the original NDVI time series data.

$$\mathrm{pt} = \beta 0 + \beta 1\mathrm{t} + \mathrm{et} \tag{2}$$

where pt is NDVI at time t, et is a random error, β0 and β1 use for better understanding of each pixel with coefficients images.

Then Harmonic curve was assessed through reducer and median function in NDVI profiles corresponding to each forest land cover pattern [25], (Eqs. 3 and 4).

$$A\cos(2\pi\omega\mathrm{t} - \varphi) = \beta 2\cos(2\pi\omega\mathrm{t}) + \beta 3\sin(2\pi\omega\mathrm{t}) \tag{3}$$

$\beta 2 = A\cos(\varphi)$
$\beta 3 = A\sin(\varphi)$
$A =$ amplitude $= (\beta 22 + \beta 32)$½
$\varphi =$ phase $= a\tan(\beta 3/\beta 2)$
$\omega =$ angular frequency.

Amplitude and phase of this linearize harmonic function fit by the following linear model with ($\omega = 1$):

$$\begin{aligned} \mathrm{pt} &= \mathrm{NDVIt} \\ &= \beta 0 + \beta 1\mathrm{t} + A\cos(2\pi\omega\mathrm{t} - \varphi) + \mathrm{et} \\ &= \beta 0 + \beta 1\mathrm{t} + \beta 2\cos(2\pi\omega\mathrm{t}) + \beta 3\sin(2\pi\omega\mathrm{t}) + \mathrm{et} \end{aligned} \tag{4}$$

Ordinary least squares regression setup for fitting the model in one cycle per unit time ($\omega = 1$) with third and fourth terms (Eq. 4). Hue saturation value (HSV) transformation algorithm presented the phase and amplitude of the image collection for converting the bands between 0 and 1 (HSV to RGB).

2.4 Autocovariance and Autocorrelation

The autocovariance of a time series specifying the lagged collection of the detrended Landsat data. Lag value acquired from the same image at the time $t - l$, where l is the lag. The autocovariance reducer assumes a set of one-dimensional arrays as input. The autocorrelation is very useful to link the preceding values to the recent values within a specified time difference (in days). Earth Engine could recall the temporal cadence of Landsat images stacked into one image per time point that computes through autocovariance and map functions [25], (Eqs. 5 and 6).

$$\text{AutoCovariance} = \text{Cov}(\text{pt},\ \text{pt} - \text{l}) = E\big[(\text{pt} - \text{E}[\text{pt}])(\text{pt} - \text{l} - E[\text{pt} - \text{l}])\big] \quad (5)$$

$$\begin{aligned}\text{AutoCorrelation} &= \text{Corr}(\text{pt},\ \text{pt} - \text{l}) \\ &= \text{Cov}(\text{pt},\ \text{pt} - \text{l})/(\ \text{SD}(\text{pt})\text{SD}(\text{pt} - \text{l})\ )\end{aligned} \quad (6)$$

t = time
pt = NDVIt = pixel value at time t
pt − l = NDVIt − l = pixel value at time $t - l$

2.5 Cross-Covariance and Cross-Correlation

Forest phenology pattern is a key indicator of climate-biosphere relationships. Cross-covariance and cross-correlation are essential for predicting climate change impact on the forest ecology [27]. Cross-covariance is similar to autocovariance, measuring a pixel at time t and a covariate at time $t - l$ (Eq. 7 and 8).

$$\text{CrossCovariance} = \text{Cov}(\text{pt},\ \text{ct} - \text{l}) = E\big[(\text{pt} - \text{E}[\text{pt}])(\text{ct} - \text{l} - E[\text{ct} - \text{l}])\big] \quad (7)$$

$$\begin{aligned}\text{CrossCorrelation} &= \text{Corr}(\text{pt},\ \text{ct} - \text{l}) \\ &= \text{Cov}(\text{pt},\ \text{ct} - \text{l})/(\ \text{SD}(\text{pt})\text{SD}(\text{ct} - \text{l}))\end{aligned} \quad (8)$$

t = time
pt = NDVIt = pixel value at time t
ct − l = covariate value at time t – l

In this research, Cross-covariance hypothesizes that annual precipitation CHIRPS data delineating the NDVI response at time $t - l$.

2.6 *Autoregressive Models*

Autoregressive models postulate a linear model considering at time t depends on previous pixel values (Eq. 9). This model predicted the next pixel value due to cloud obstacle and fill in the blanks pixel through interpolate on the time series. Linear regression reducer function merging the lagged collection for fitted values into a list. A lagged collection was created where the images list stored in the two previous images by the following specimen in lagged 30 days.

var lagged30 = ee.ImageCollection(lag(filteredLandsat, filteredLandsat, 30));

$$\mathrm{pt} = \beta 0 + \beta 1\mathrm{pt} - 1 + \beta 2\mathrm{pt} - 2 + \mathrm{et} \tag{9}$$

pt = NDVI at time *t*
t = time
et = random error
β's are specified every pixel.

3 Results and Discussion

We adapted Clinton, [28] Earth Engine Developer Relations https://goo.gl/lMwd2Y GEE harmonic regression architecture with the help of Landsat derived NDVI original and fitted time series (2000–2019) for three different forest cover categories in the study area. A detailed workflow of Harmonic regression, Autocovariance, Cross-correlation, and Autoregressive model with NDVI profile estimations are shown in Figs. 6, 7 and 8.

Each harmonic analysis, the frequency, amplitude, and phase were estimated and implemented a linear detrending method to eliminate the gradient from the data (see Figs. 2, 3 and 4). There is reduction in dense forest in the year 2005, 2009, and 2011. Even in 2016, there is reduction in dense forest coverage. This can be easily observed in rise in moderately dense forest and open forest in the corresponding years.

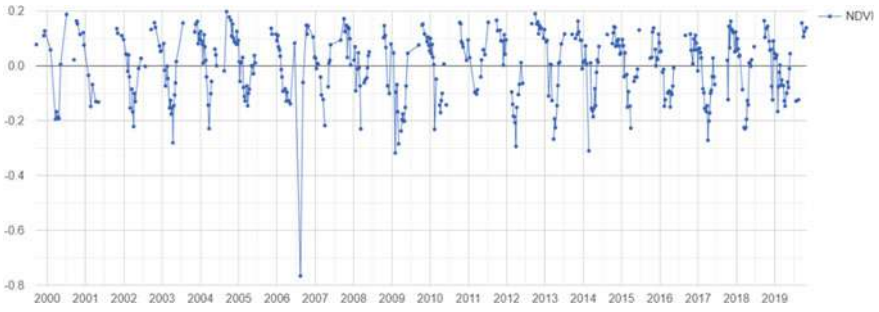


Fig. 2 Detrended Landsat time series for dense forest in the study area

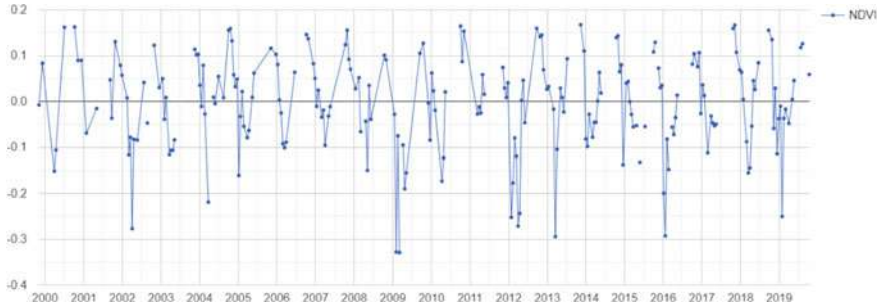


Fig. 3 Detrended Landsat time series for moderately dense forest in the study area

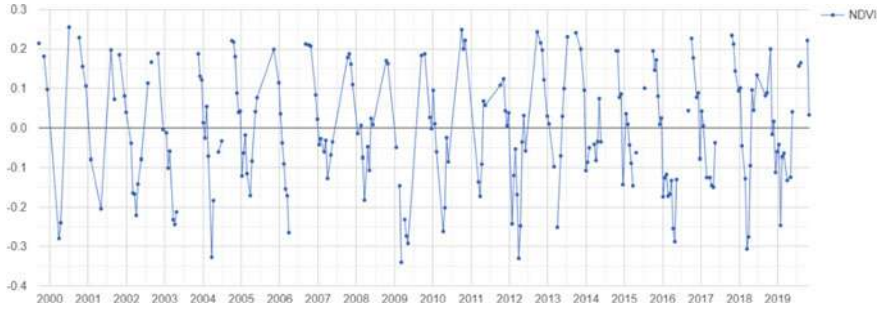


Fig. 4 Detrended Landsat time series for the open forest in the study area

3.1 Harmonic Modeling

Model was fitted to show the linear trend, using the linear Regression() reducer function. The coefficients were plugged to get a time series of fitted values. Although any coefficients can be mapped directly, it is useful and interesting to map the amplitude and phase of the assessed harmonic function as an HSV transform map (see Fig. 5). The amplitude and phase were calculated from the coefficient's values, then maps were produced. The harmonic analysis was carried out per pixel basis NDVI time series for the forest land cover phenology distribution area. The inter-annual variability in amplitude and phase for these distributions were then evaluated. The Harmonic regression model with the original fitted value for each forest category (see Figs. 6, 7 and 8) for illustration of forest health dynamics. It also identified the sensitivity of NDVI time series evaluation with climate change relationship.

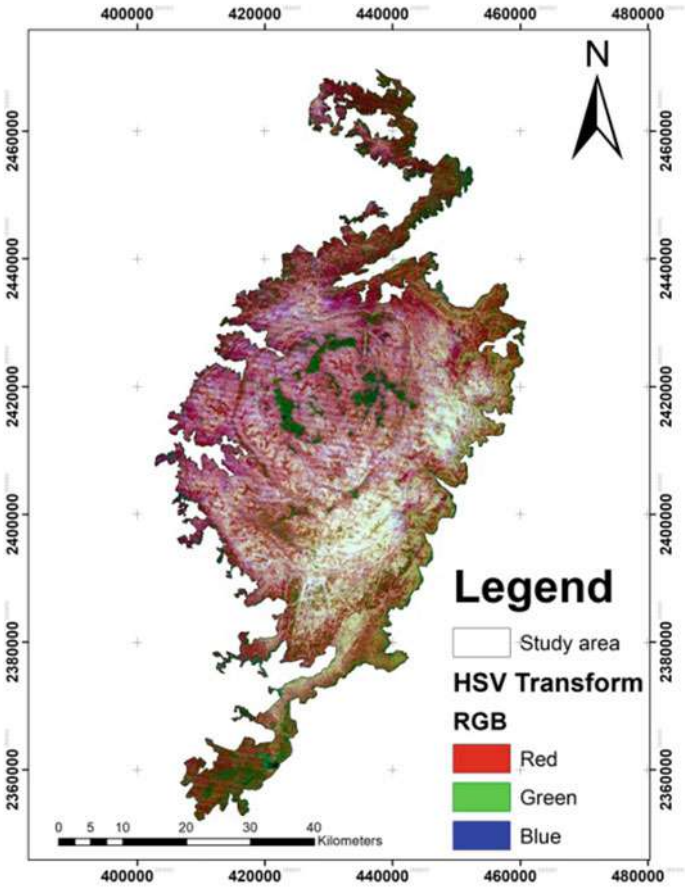


Fig. 5 The phase and amplitude of the estimated harmonic model as HSV transform map

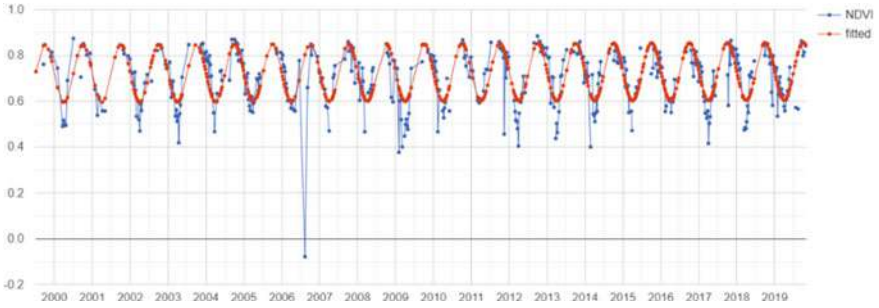


Fig. 6 Harmonic regression model with original fitted value for dense forest

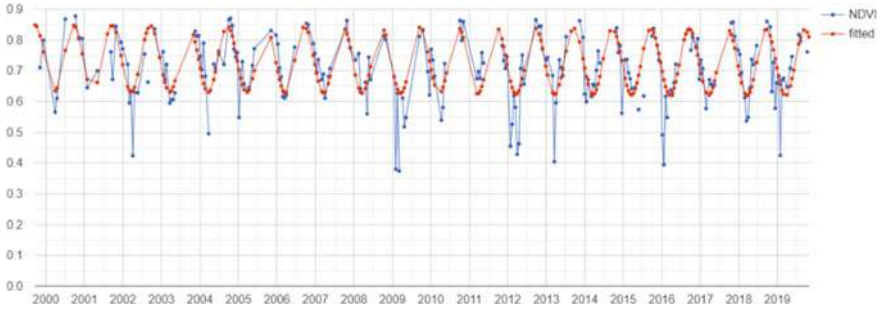


Fig. 7 Harmonic regression model with original fitted value for moderate forest

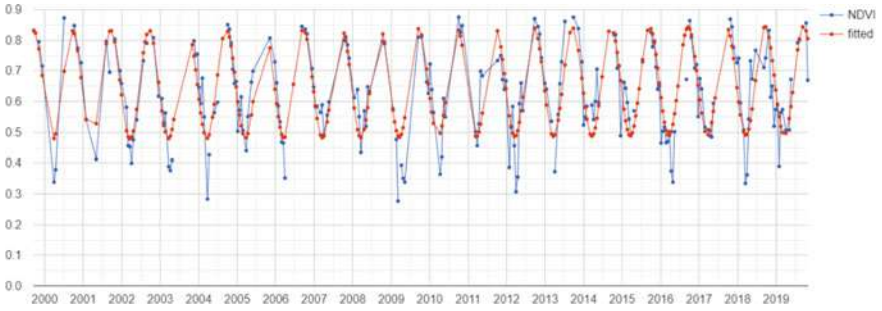


Fig. 8 Harmonic regression model with an original fitted value of open forest

3.2 Autoregressive Models

The autoregressive (AR) model is a function of previous pixel values, which was masked by the linear models of at time t-l. In this situation, missing data due to cloud cover is a big issue. AR model is best suitable for predicting fitting values in forest NDVI time series analysis. With AR model, the original and fitted values were estimated (Figs. 9, 10 and 11), then the missing or masked data used to generate forecasting fitting values of each forest category in the study area (Figs. 12, 13 and 14). Tree phenology is influenced by various climatic traits such as temperature and precipitation. Even the vegetation health parameters are affected by various climatic hazards and anthropogenic disturbance activity. The AR model explained the variation occurs in harmonic parameters estimated with amplitude and phase criteria. In the AR model, there will be no missing data and pattern can be analyzed without any break.

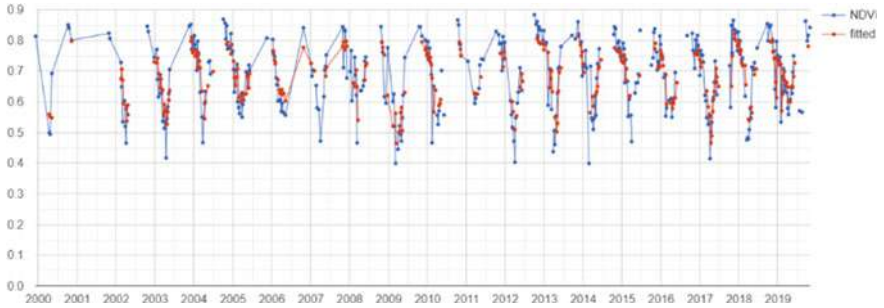


Fig. 9 Autoregressive model with original and fitted values for dense forest

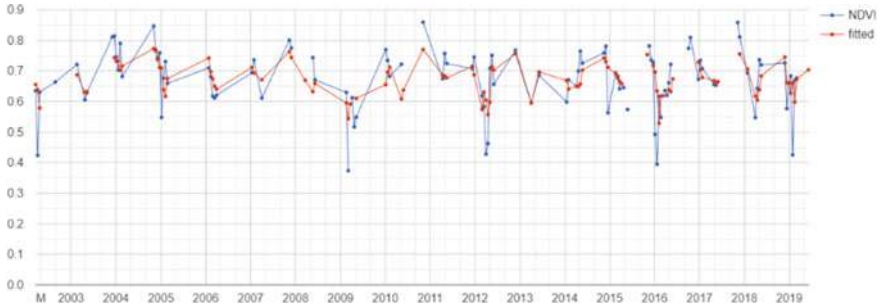


Fig. 10 Autoregressive model with original and fitted values for moderate forest

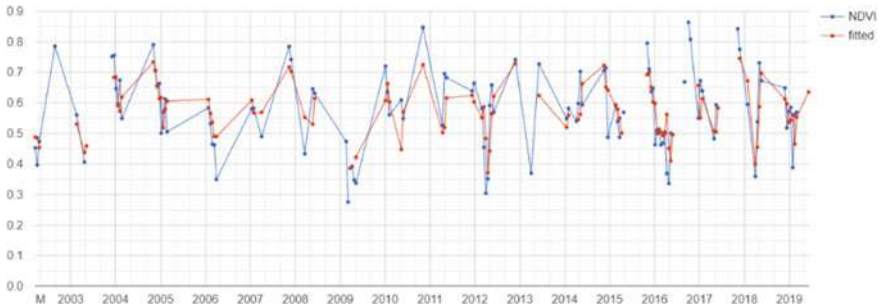


Fig. 11 Autoregressive model with original and fitted values for open forest

3.3 NDVI–Climate Relationships

As we understand, the variation in forest density existed between different temporal zones or years. These variations could be caused due to the change in climate parameters. Twenty years CHIRPS precipitation data was acquired by the GEE cloud analysis in the STR region (see Fig. 15a, b). We can clearly observed that there is reduction in

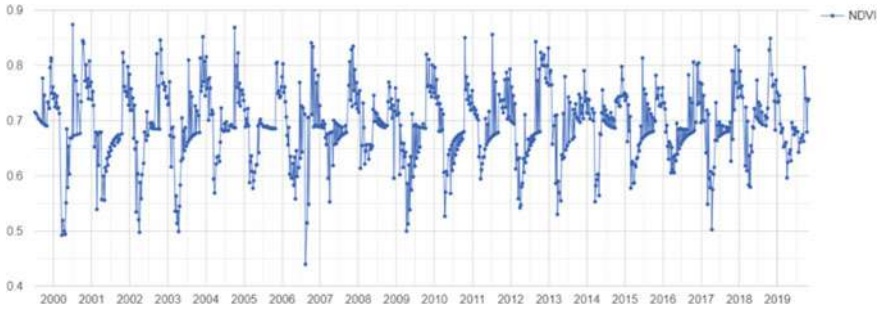


Fig. 12 Filling in missing data and forecasting with an AR model for dense forest

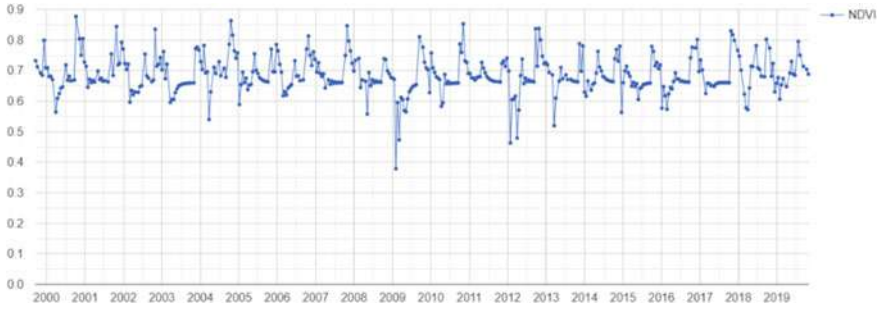


Fig. 13 Filling in missing data and forecasting with an AR model for moderate forest

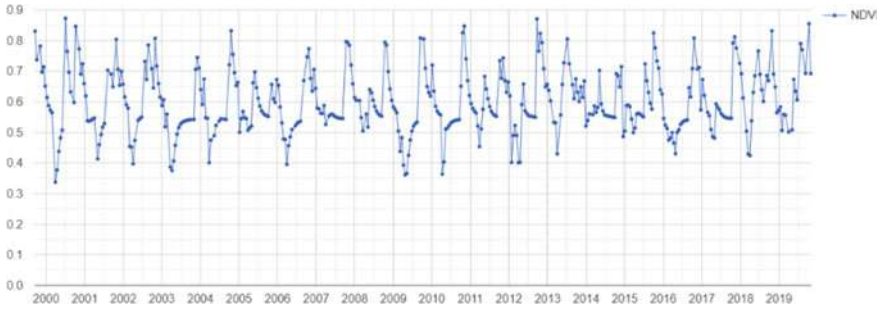


Fig. 14 Filling in missing data and forecasting with an AR model for open forest

level of average precipitation in the year 2004, 2009, 2010, 2012, and 2015. In this study NDVI time series of Landsat data evaluating the forest phenology distribution with the precipitation level. Autocorrelation time lag function allowed inter-annual variability between the NDVI and precipitation (see Fig. 16). The CHIRPS data arranged by pen Tad's and lag of five days store at each Landsat image are listed and

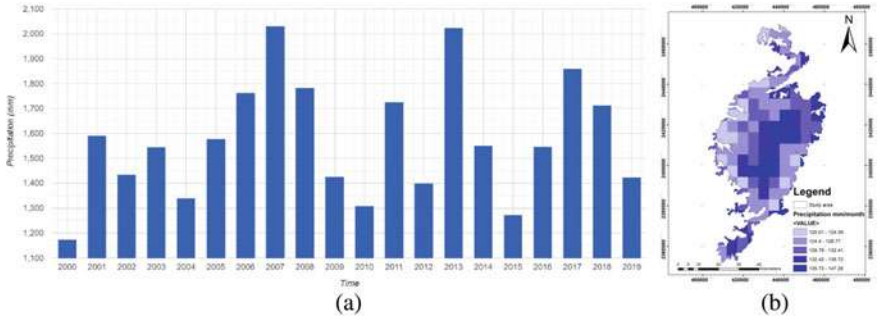


Fig. 15 Climate Hazards Group InfraRed Precipitation with Station Data, **a** annual precipitation bar plot **b** mean monthly precipitation map in the study area

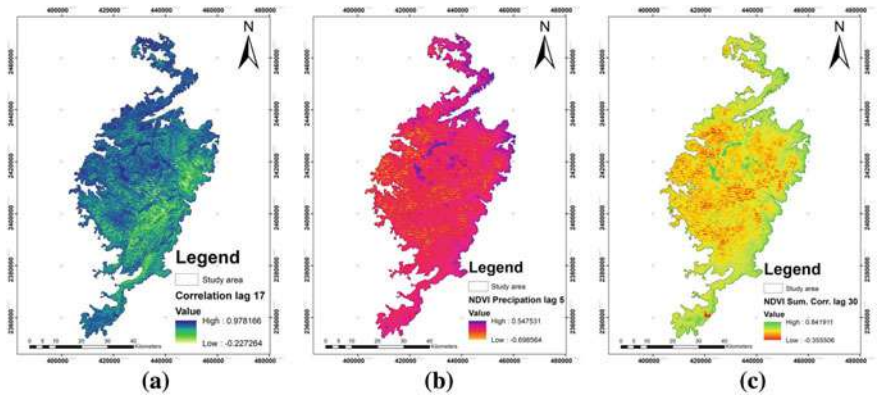


Fig. 16 ACF NDVI–climate relationships maps with time lags **a** correlation at lag 17 of NDVI **b** covariance between NDVI and precipitation 5 days **c** NDVI sum correlation 30 days

then merged the bands. We estimate each Landsat image to have a band that represents precipitation at time $t-1$ or $t-5$ days then operate covariance and correlation function.

NDVI time series of 2000–2019 years compared to the climactic series of precipitation levels using regression analysis. The association of NDVI and CHIRPS precipitation data (Table 1) was analyzed with covariance (Cov.), correlation (Corr.) to NDVI, lag (days), and summation period (days) (see Fig. 16a–c). The highest lag 5 days correlation was found at the open forest (0.28) and highest correlation of lag 30 was found at the open forest (0.62) (see Fig. 16b, c). The northern part of the STR region found a very good relationship with the NDVI and precipitation level. So long-term lag periods are a very good correlation between precipitation and NDVI in the STR region. The highest NDVI summation correlation found in 5 lag and 30 lag periods for three different forest categories such as dense forest, moderately dense forest, and open forest, respectively (Table 1).

Table 1 Precipitation of covariance (Cov.), correlation (Corr.) to NDVI, lag (days), and summation period (days) for three forest types in the STR region

				lag 17		lag 5		lag 30	
Plot ID	Long	Lat	Forest type	Cov. lag	Corr. lag	NDVI Precip Cov	NDVI Precip Corr	NDVI sum Cov	NDVI sum Corr
A1	86.3503	21.9061	Dense forest	0.016	0.825	0.173	0.110	0.173	0.322
A2	86.2419	21.7161	Dense forest	0.019	0.894	0.096	0.057	0.096	0.360
A3	86.3277	21.9169	Dense forest	0.019	0.904	0.180	0.095	0.180	0.321
A4	86.2803	21.8411	Dense forest	0.010	0.834	0.069	0.053	0.069	0.347
A5	86.3064	21.5176	Dense forest	0.025	0.933	0.320	0.153	0.320	0.538
A6	86.4369	21.9213	Dense forest	0.015	0.859	0.172	0.090	0.172	0.399
A7	86.2535	21.971	Dense forest	0.009	0.826	0.011	0.008	0.011	0.277
A8	86.235	21.8404	Dense forest	0.018	0.905	0.077	0.059	0.077	0.348
A9	86.2776	21.8411	Dense forest	0.011	0.836	0.081	0.056	0.0 81	0.319
A10	86.338	21.7722	Dense forest	0.016	0.886	0.079	0.045	0.079	0.363
A11	86.4135	21.8723	Dense forest	0.011	0.885	0.028	0.021	0.028	0.378
A12	86.2563	21.8774	Dense forest	0.011	0.826	0.091	0.077	0.091	0.316
A13	86.3902	22.0188	Dense forest	0.032	0.892	0.454	0.153	0.454	0.418
A14	86.3703	21.6408	Dense forest	0.008	0.917	0.070	0.064	0.070	0.356
A15	86.3888	21.5559	Dense forest	0.021	0.900	0.160	0.088	0.160	0.426
A16	86.1693	21.8973	Dense forest	0.015	0.813	0.176	0.111	0.176	0.350
A17	86.3598	21.9282	Dense forest	0.009	0.848	0.035	0.026	0.035	0.266
B1	86.384	21.5348	Moderately dense	0.010	0.698	0.249	0.130	0.249	0.400
B2	86.4403	21.8309	Moderately dense	0.012	0.734	0.077	0.048	0.077	0.295
B3	86.3236	21.4741	Moderately dense	0.007	0.729	0.158	0.142	0.158	0.474

(continued)

Table 1 (continued)

				lag 17		lag 5		lag 30	
Plot ID	Long	Lat	Forest type	Cov. lag	Corr. lag	NDVI Precip Cov	NDVI Precip Corr	NDVI sum Cov	NDVI sum Corr
B4	86.2837	21.4562	Moderately dense	0.005	0.670	0.150	0.109	0.150	0.472
B5	86.5584	21.9647	Moderately dense	0.017	0 .725	0.216	0.101	0.216	0.455
B6	86.27	21.755	Moderately dense	0.004	0.714	0.019	0.020	0.019	0.368
B7	86.1334	21.7773	Moderately dense	0.014	0.783	0.032	0.020	0.032	0.284
B8	86.1972	21.637	Moderately dense	0.007	0.683	0.013	0.010	0.013	0.278
B9	86.4642	21.7765	Moderately dense	0.012	0.694	0.239	0.122	0.239	0.391
B10	86.4666	21.6588	Moderately dense	0.013	0.742	0.327	0.159	0.327	0.487
B11	86.3673	21.4906	Moderately dense	0.010	0.703	0.461	0.174	0.461	0.558
B12	86.2417	21.5238	Moderately dense	0.007	0.701	0.102	0.069	0.102	0.431
B13	86.5579	21.9626	Moderately dense	0.024	0.729	0.494	0.174	0.494	0.410
B14	86.5629	21.9186	Moderately dense	0.023	0.714	0.264	0.095	0.264	0.380
B15	86.5564	21.9062	Moderately dense	0.010	0.639	0.158	0.113	0.158	0.417
B16	86.4644	21.7657	Moderately dense	0.014	0.754	0.316	0.161	0.316	0.468
B17	86.3971	21.5605	Moderately dense	0.012	0.739	0.231	0.112	0.231	0.398
C1	86.5762	21.8258	Open forest	0.005	0.592	0.638	0.289	0.638	0.597
C2	86.5048	21.7218	Open forest	0.007	0.674	0.162	0.075	0.162	0.489
C2	86.5495	21.8366	Open forest	0.005	0.609	0.234	0.174	0.234	0.460
C3	86.5714	21.8283	Open forest	0.009	0.519	0.393	0.218	0.393	0.569
C4	86.5948	21.9436	Open forest	0.013	0.592	0.530	0.219	0.530	0.472

(continued)

Table 1 (continued)

				lag 17		lag 5		lag 30	
Plot ID	Long	Lat	Forest type	Cov. lag	Corr. lag	NDVI Precip Cov	NDVI Precip Corr	NDVI sum Cov	NDVI sum Corr
C5	86.4485	21.614	Open forest	0.003	0.611	0.236	0.141	0.236	0.549
C6	86.3984	22.2261	Open forest	0.005	0.519	0.176	0.118	0.176	0.387
C7	86.443	22.2954	Open forest	0.010	0.631	0.258	0.116	0.258	0.423
C8	86.5135	21.753	Open forest	0.002	0.403	0.285	0.173	0.285	0.523
C9	86.5129	21.7124	Open forest	0.007	0.664	0.434	0.194	0.434	0.586
C10	86.4401	21.4995	Open forest	0.009	0.552	0.565	0.190	0.565	0.582
C11	86.5991	21.8798	Open forest	0.011	0.657	0.498	0.223	0.498	0.554
C12	86.5349	21.7858	Open forest	0.006	0.503	0.481	0.216	0.481	0 .492
C13	86.6096	21.9234	Open forest	0.005	0.456	0.408	0.215	0.408	0.629
C14	86.5928	21.9457	Open forest	0.016	0.678	0.773	0.280	0.773	0.575
C15	86.5798	21.9533	Open forest	0.008	0.503	0.347	0.178	0.347	0.495
C16	86.4953	22.0097	Open forest	0.007	0.463	0.041	0.022	0.041	0.389
C17	86.6004	21.9008	Open forest	0.008	0.592	0.447	0.229	0.447	0.627

4 Discussion

This paper demonstrates precious insights into how satellite-based time series, can be effectively optimized in guiding forest health ecosystems analysis. A freely available GEE fast cloud computing was used to processing big multitemporal Landsat images within a limited period. The powerful GEE cloud computing approach provides quick processing of big data in real-time and near real-time analysis. This research investigated long-term forest phenology pattern correspondence to precipitation level in Similipal Tiger Reserve, Odisha, India for calendar years 2000–2019. Spatial analysis of forest health dynamics estimated by the harmonic regression were corrected through autocovariance, autocorrelation, and autoregressive model in GEE platform. Three different forest covers mainly dense forest, moderately dense, and open forest area were used for the NDVI profiling. Sensitivity of phenology is a vital indicator for understanding the climate change in biosphere processes [29]. Harmonic optimization was also incorporated with data series length and degree for the better ability of seasonal vegetation signal [30, 31].

Our study defined the long-term relationship between NDVI and precipitation in specific periods. This relationship showed through the autocovariance and autocorrelation methods. We demonstrate these data to be rich in spatiotemporal statistics, proving distinct phenology distribution as a consequence of multifaceted overlying gradients forest ecosystem and climate change pattern [32, 33]. This research successfully encounters the missing value due to cloud cover problem with the help of linear trend autoregressive model.

The bioregional level outcome is a great concern of phenological response with the precipitation between forest land cover ecosystem and climatic gradients [34, 35]. Generally, tropical regions are strongly associated with precipitation. So, this research (Table 1) found a strong correlation with precipitation level, though at different forest type lags and summation periods such as dense forest, moderately dense forest, and open forest correlation with lag 17 days of 0.93, 0.783, and 0.678, respectively. However, wide areas showed the highest NDVI summation correlations with the precipitation level in tropical biomes which specify a synoptic phenological pattern of correlation, lag, and summation period at the global to local scale [36–38]. This recommends that tropical biomass is affected by precipitation occurring approximately two months preceding the plant's response, with the summation accounting for about six weeks and a lag of about two weeks [39–41]. These spatial patterns derive entirely from temporal evidence, signifying that the remote sensing technique can notify the understanding and prediction of forest health dynamics [42–45]. Our study showed that the amplitude of the strongest harmonic term was more sensitive to precipitation than the dry season NDVI (Fig. 16c).

Our research considerate and applying the RS-based GEE for NDVI phenology distribution for forest cover monitoring and mapping. GEE code editor is flexible for developing the suitable code with assistances (JavaScript, Python), and a steady web service for the spatial analysis of forest health status. We recommend that spatial

phenology information through earth observation for land cover classification [46] climate change vulnerability analysis, and anthropogenic disturbance studies.

5 Conclusion

This research investigated long-term forest phenology pattern correspondence to precipitation level in Similipal Tiger Reserve, Odisha, India for calendar years 2000–2019. NDVI enabled harmonic metrics linking between quantifications of forest gradient and strategic forest ecosystem management for a long-term basis. Landsat optical data are support in time series analyses for seasonal forest phenology variability for three forest covers, such as dense, moderately dense, and open forest areas. Inter-annual NDVI data compared to the climactic series of CHIRPS precipitation data presented the synoptic phenological pattern in the tropical biome. The models successfully identify the causes of variation of forest covers through auto regression and precipitation lag maps. GEE platform can be recommended for forest ecosystem management and sustainable environment planning with the help of remote sensing data.

Conflicts of Interest The authors declare no conflicts of interest.

Acknowledgements The authors do hereby acknowledged the contribution Visva-Bharati (A CentralUniversity),West Bengal, India for facilitating this research work and sincerely thank the anonymous reviewers and the editors for their valuable comments and constructive suggestions.

References

1. Coppin P, Jonckheere I, Nackaerts K, Muys B, Lambin E (2004) Review ArticleDigital change detection methods in ecosystem monitoring: a review. Int J Remote Sens 25:1565–1596
2. Wang Y, Ziv G, Adami M, Mitchard E, Batterman SA, Buermann W, Schwantes Marimon B, Hur B, Junior M, Reis SM et al (2019) Mapping tropical disturbed forests using multi-decadal 30 m optical satellite imagery. Remote Sens Environ 21:474–488
3. Nyland KE, Gunn GE, Shiklomanov NI, Engstrom RN, Streletskiy DA (2018) Land cover change in the lower Yenisei River using dense stacking of landsat imagery in Google Earth Engine. Remote Sens 10:1226
4. Gorelick N, Hancher M, Dixon M, Ilyushchenko S, Thau D, Moore R (2017) Google Earth Engine: planetary-scale geospatial analysis for everyone. Remote Sens Environ 202:18–27
5. Amani M, Ghorbanian A, Ahmadi SA, Kakooei M, Moghimi A, Mirmazloumi SM, Alizadeh M, Sayyed H, Mahdavi S, Ghahremanloo M, Parsian S, Wu Q, Brisco B (2020) Google Earth Engine cloud computing platform for remote sensing big data applications: a comprehensive review. IEEE J Selected Topics Appl Earth Observations Remote Sens 1–1. https://doi.org/10.1109/JSTARS.2020.3021052
6. Singha C, Swain KC, Sahoo BB, Ghosh P, Swain SK (2019) Assessment of bio diversity conservation using geospatial models. J Pharmacognosy Phytochem 8(1):1177–1186

7. Swain KC, Swain SK, Singha C (2017) Status mapping of landuse/land cover change of forest resources. J Energy Res Environ Technol 234–239
8. Smith-Tripp S (2020) 2020–22 Remote sensing as a tool for efficient forest health and landscape monitoring in Metro Vancouver's water supply areas. UBC Sustainability (2020). Scholar. https://sustain.ubc.ca/sites/default/files/2020-22_Remote%20sensing%20as%20a%20tool_Smith-Tripp.pdf. Last accessed 2020/12/21
9. Zhu Z, Woodcock CE, Holden C, Yang Z (2015) Generating synthetic Landsat images based on all available Landsat data: predicting landsat surface reflectance at any given time. Remote Sens Environ 162:67–83
10. Shimizu K, Ota T, Mizoue N (2019) Detecting forest changes using dense Landsat 8 and Sentinel-1 time series data in tropical seasonal forests. Remote Sens 11:1899
11. Roy DP, Kovalskyy V, Zhang HK, Vermote EF, Yan L, Kumar SS, Egorov A (2016) Characterization of Landsat-7 to Landsat-8 reflective wavelength and normalized difference vegetation index continuity. Remote Sens Environ 185:57–70
12. Wilson BT, Knight JF, McRoberts RE (2018) Harmonic regression of Landsat time series for modeling attributes from national forest inventory data. ISPRS J Photogramm Remote Sens 137:29–46. https://doi.org/10.1016/j.isprsjprs.2018.01.006
13. Pasquarella VJ, Holden CE, Woodcock CE (2018) Improved mapping of forest type using spectral-temporal Landsat features. Remote Sens Environ 210:193–207
14. Adams B, Iverson L, Matthews S, Peters M, Prasad A, Hix DM (2020) Mapping forest composition with landsat time series: an evaluation of seasonal composites and harmonic regression. Remote Sens 12(4):610. https://doi.org/10.3390/rs12040610
15. Landmann T, Eidmann D, Cornish N, Franke J, Siebert S (2019) Optimizing harmonics from Landsat time series data: the case of mapping rainfed and irrigated agriculture in Zimbabwe. Remote Sens Lett 10(11):1038–1046. https://doi.org/10.1080/2150704x.2019.1648901
16. Vogeler JC, Braaten JD, Slesak RA, Falkowski MJ (2018) Extracting the full value of the Landsat archive: Inter-sensor harmonization for the mapping of Minnesota forest canopy cover (1973–2015). Remote Sens Environ 209:363–374
17. Ibrahim S, Balzter H, Tansey K, Tsutsumida N, Mathieu R (2018) Estimating fractional cover of plant functional types in African savannah from harmonic analysis of MODIS time-series data. Int J Remote Sens 39(9):2718–2745. https://doi.org/10.1080/01431161.2018.1430914
18. Fernández-Manso A, Quintano C, Fernández-Manso O (2011) Forecast of NDVI in coniferous areas using temporal ARIMA analysis and climatic data at a regional scale. Int J Remote Sens 32(6):1595–1617. https://doi.org/10.1080/01431160903586765
19. Clinton N, Yu L, Fu H, He C, Gong P (2014) Global-scale associations of vegetation phenology with rainfall and temperature at a high spatio-temporal resolution. Remote Sens 6(8):7320–7338. https://doi.org/10.3390/rs6087320
20. Wang J, Meng JJ, Cai YL (2008) Assessing vegetation dynamics impacted by climate change in the Southwestern Karst region of China with AVHRR NDVI and AVHRR time series. Environ Geol 54(6):1185–1195
21. Barka I, Bucha1 T, Molnár T, Móricz N, Somogyi Z, Koreň M (2019) Suitability of MODIS-based NDVI index for forest monitoring and its seasonal applications in Central Europe. Central Eur Forestry J 65:206–217. https://doi.org/10.2478/forj-2019-0020
22. Meng M, Ni J, Zong M (2011) Impacts of changes in climate variability on regional vegetation in China: NDVI based analysis from 1982 to 2000. Ecol Res 26(2):421–428
23. Brunsell N (2006) Characterization of land-surface precipitation feedback regimes with remote sensing. Remote Sens Environ 100:200–211
24. Hong Y, Hsu KL, Sorooshian S, Gao X (2004) Precipitation estimation from remotely sensed imagery using an artificial neural network cloud classification system. J Appl Meteorol Climatol 43:1834–1852
25. Shumway RH, Stoffer DS (2017) Time series analysis and its applications with R examples, Fourth Edition, Pittsburgh, PA. Springer International Publishing, Cham, Switzerland
26. Venkatappa M, Sasaki N, Shrestha RP, Tripathi NK, Ma HO (2019) Determination of vegetation thresholds for assessing land use and land use changes in Cambodia using the Google Earth

Engine cloud-computing platform. Remote Sens 11(13):1514. https://doi.org/10.3390/rs11131514
27. Abbas S, Nichol JE, Wong MS (2021) Trends in vegetation productivity related to climate change in China's Pearl River Delta. PLoS ONE 16(2):e0245467. https://doi.org/10.1371/journal.pone.0245467
28. Clinton N (2019) Time series analysis in Earth Engine. Earth Engine Developer Relations. https://goo.gl/lMwd2Y. Last accessed 2020/11/21
29. Sharma M, Bangotra P, Gautam AS, Gautam S (2021) Sensitivity of normalized difference vegetation index (NDVI) to land surface temperature, soil moisture and precipitation over district Gautam Buddh Nagar, UP, India. Stoch Env Res Risk Assess. https://doi.org/10.1007/s00477-021-02066-1
30. Naif SS, Mahmood DA, Al-Jiboori MH (2020) Seasonal normalized difference vegetation index responses to air temperature and precipitation in Baghdad. Open Agric 5(1):631–637. https://doi.org/10.1515/opag-2020-0065
31. Guha S (2021) Dynamic seasonal analysis on LST-NDVI relationship and ecological health of Raipur City, India. Ecosyst Health Sustain 1927852. https://doi.org/10.1080/20964129.2021.1927852
32. Su Y, Wang D, Zhao S, Shi J, Shi Y, Wei D (2021) Examining long-term natural vegetation dynamics in the Aral Sea Basin applying the linear spectral mixture model. PeerJ 9:e10747. https://doi.org/10.7717/peerj.10747
33. Guechi I, Gherraz H, Alkama D (2021) Correlation analysis between biophysical indices and Land Surface Temperature using remote sensing and GIS in Guelma city (Algeria), Bulletin de la Société Royale des Sciences de Liège [En ligne] 90:158–180. https://popups.uliege.be/0037-9565/index.php?id=10457. Last accessed 2021/11/22
34. Li Y, Qin Y, Ma L, Pan Z (2020) Climate change: vegetation and phenological phase dynamics. Int J Climate Change Strategies Manage 12(4):495–509. https://doi.org/10.1108/IJCCSM-06-2019-0037
35. Pei Z, Fang S, Yang W, Wang L, Wu M, Zhang Q, Han W, Khoi DN (2019) The relationship between NDVI and climate factors at different monthly time scales: a case study of Grasslands in Inner Mongolia, China (1982–2015). Sustainability 11(24):7243. https://doi.org/10.3390/su11247243
36. Adepoju K, Adelabu S, Fashae O (2019) Vegetation Response to recent trends in climate and landuse dynamics in a typical humid and dry tropical region under global change. Adv Meteorol 1–15. https://doi.org/10.1155/2019/4946127
37. LouzeiroGoogle R (2021) Earth Engine—how it can be used in Environmental Science. Integrate Sustainability. https://www.integratesustainability.com.au/wp-content/uploads/2021/06/20210601-Google-Earth-Engine-2.pdf.last. Accessed 2021/09/22
38. Lin X, Niu J, Berndtsson R, Yu X, Zhang L, Chen X (2020) NDVI dynamics and its response to climate change and reforestation in Northern China. Remote Sens 12:4138. https://doi.org/10.3390/rs12244138
39. Osei JD, Andam-Akorful SA, Osei Jr EM (2019) Long term monitoring of Ghana's forest reserves using Google Earth Engine. Preprints 2019090016. https://doi.org/10.20944/preprints201909.0016.v1
40. Grantham HS, Duncan A, Evans TD, Jones KR, Beyer HL, Schuster R, Walston J, Ray JC, Robinson JG, Callow M, Clements T, Costa HM, DeGemmis A, Elsen PR, Ervin J, Franco P, Goldman E, Goetz S, Hansen A, Hofsvang E, Jantz P, Jupiter S, Kang A, Langhammer P, Laurance WF, Lieberman S, Linkie M, Malhi Y, Maxwell S, Mendez M, Mittermeier R, Murray NJ, Possingham H, Radachowsky J, Saatchi S, Samper C, Silverman J, Shapiro A, Strassburg B, Stevens T, Stokes E, Taylor R, Tear T, Tizard R, Venter O, Visconti P, Wang S, Watson JEM (2020) Anthropogenic modification of forests means only 40% of remaining forests have high ecosystem integrity. Nat Commun 11:5978. https://doi.org/10.1038/s41467-020-19493-3
41. Kalisa W, Igbawua T, Henchiri M, Ali S, Zhang S, Bai Y, Zhang J (2019) Assessment of climate impact on vegetation dynamics over East Africa from 1982 to 2015. Sci Rep 9:16865. https://doi.org/10.1038/s41598-019-53150-0

42. Galvão LS, dos Santos JR, Roberts DA, Breunig FM, Toomey M, de Moura YM (2011) On intra-annual EVI variability in the dry season of tropical forest: A case study with MODIS and hyperspectral data. Remote Sens Environ 115:2350–2359
43. Small C (2012) Spatiotemporal dimensionality and time-space characterization of multitemporal imagery. Remote Sens Environ 124:793–809
44. Morton DC, Nagol J, Carabajal CC, Rosette J, Palace M, Cook BD, Vermote EF, Harding DJ, North PRJ (2014) Amazon forests maintain consistent canopy structure and greenness during the dry season. Nature 506:221–224
45. Swain KC, Zaman QU, Jayasuriya HPW, Zhang F (2009) Estimation of rice yield and protein content using remote sensing images acquired by radio controlled unmanned helicopter. In: ASABE conference at providence, Rhode Island, June 29–July 2, 2008, 1 p
46. Singha C, Swain, KC, Swain SK (2020) Best crop rotation selection with GIS-AHP technique using soil nutrient variability. Agriculture 10(6):213. https://doi.org/10.3390/agriculture10060213

Chapter 53
AI-Based Tracking System from Real-Time CCTV Captures

N. Malarvizhi, Arun Kumar Dash, V. Manikanta, and Athreayasa Kalyan

1 Introduction

In a populous country like India, catching criminals is definitely not an easy task for the police. Thus, the usage of man power and fuel plays a huge role in searching for criminals. According to factly.in, an average of 200 children go missing in India every day. Girls made up 52,049 missing cases in 2019 while boys accounted for 21,074 cases. It has also been reported that 15 transgender children went missing in 2019. The count of missing people increases day by day which eventually decreases the rate of successful identification of missing people, thereby increase in number of unclosed cases.

Every day, there are a minimum of 50 suspects and numerous identified criminals who are roaming freely in the public. Huge crowd gathering places like shopping malls, cricket stadiums, music concerts are places for terrorists to gather to execute their plans before police identifies them in that area from cameras. The objective of this project is to help police and higher authorities to track down missing people and wanted criminals quickly. The usual process to track a person is using investigation which requires time and experience to ask right questions. Most of the time, investigation method works pretty well, but it is time consuming and can be unsuccessful if the person missing has been shifted/moved to different location (city/country). In

N. Malarvizhi · A. Kumar Dash (✉) · V. Manikanta · A. Kalyan
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India
e-mail: vtu11182@veltech.edu.in

N. Malarvizhi
e-mail: drnmalarvizhi@veltech.edu.in

V. Manikanta
e-mail: vtu15955@veltech.edu.in

A. Kalyan
e-mail: vtu15933@veltech.edu.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_54

such cases, the ideal approach is to go through CCTV footages and evidences. Again, this can be very time-consuming, and given the number of people that go missing every day, it can be a challenge to keep up with it. The existing automated system uses an outdated algorithm on a slower hardware environment without considering any ageing factors or resolution-based challenges faced by the CCTV cameras.

The previous system used Haar cascade classifier to recognize faces in rotated and blurred images. It uses steps like feature selection, creating integral images, AdaBoost training and cascading classifier. This is done by Facepplib library from Python. Our system uses face_recognition library in Python which was developed by Adam Gaitgey. This library uses some faster methods to detect and identify images compared to the previous library. It uses several steps such as generating HOG version of an image, comparing it to detect faces, pose identification, making the head to face forward, generating a matrix of facial features and finally recognition.

2 Literature Survey

Ayyappan and Matilda propose a system to detect and recognize faces from online blogs by extracting the multimedia from the websites and recognition of faces using multiple classifiers. It uses Haar classifier to detect faces, and recognition is done by extracting the features and comparison of templates with the help of Facepplib library of Python. Although the system is proved faster in terms of detection, this system will be eventually slower as it does not use any cloud technologies for performance [1].

Navneet Dalal proposes an algorithm to easily detect a face from an image with increased speed overcoming the system of pixel comparison. Histograms of oriented gradient (HOG), an algorithm to detect face by creating a histogram with arrow heads all over the image by comparing nearby pixels and identifying the direction of light. This method is faster as it only stores few values and compares them using a neural network with previously identified faces to detect a face in the image [2].

Adam Gaitgey proposed a step-by-step methodology to recognize a face using the previous method of HOG to detect face and change pose and compare it with other faces to output the most similar face [3].

Han provides in his paper the techniques to detect multiple faces in a single image. As the detection of a human was a challenge by that time. It proposed an algorithm to detect multiple objects (humans) from videos. Till then, only one object was able to track from a single image. Even though it over came the difficulties of the previous systems, it uses single classifier and outdated architecture. It was able to deal with irregular object motions, trajectories of an object which helps to track the motion of human [4].

Anil Jain discusses the challenges faced in forensics to recognize a face from the available image of a suspect. His paper discusses the challenges such as the ageing of criminals by time, development of facial marks, recognition of criminals using sketches produced by the department, detection of faces from near-infrared (NIR)

images and soft biometrics. This analysis of challenges paved way for development of better recognition technologies [5].

Jose proposed a face recognition-based surveillance system using FaceNet dataset. FaceNet is deep convolutional neural network from Google. It easily converts a facial image to vectors to process easily for comparison and identify near probabilistic images. It proposes the usage of MTCNN over other classifiers and deployed over Jetson TX2. This was fast compared to previous hardware modules due to advances in portable microcontroller devices with high end memory and good cooling technologies from Nvidia, and usage of pre-trained models increases speed and accuracy [6].

Klontz has discussed various challenges faced tracking down the suspects of the Boston Marathon Bombings in his paper. The challenges faced over face recognition in CCTV footages such as ageing of suspects, camera footages, where they were seen, but were not able to recognize because of caps, masks covering their face even with the advances in technology are discussed [7].

Papaioannou proposed a system which uses the radio and other inertial sensors to easily detect the faces of humans in motion in highly dynamic environments such as construction sites, where it is more complicated to detect faces from the environment as many objects are in motion and the scene gets more complex to the technology to detect faces. The coordinates of the user are obtained from the sensory radio frequency devices used by the targets and that locations are focused to detect the faces in that environment [8].

Shirsat proposes the usage of Azure Cognitive Services using the existing Haar cascade classifier to recognize faces of criminals over the CCTV footages as it is a better alternative to process the classifier faster than the previous systems. It is the first system to use Azure Cognitive Services for facial detection. They have proposed a low cost less accuracy which uses only Haar algorithm and uses only face recognition service of Azure and a high cost more accuracy system involving the usage of most of the services of Azure using Azure API for detection of criminals [9].

Son has used attendance taking support system (ATSS) to take attendance over the real-time CCTV of indoor security camera using face recognition and Haar cascade classifier. This system adds new data of previously registered faces whenever it feels that the frames are from the same face, but those angles are not recognized, thereby increasing the size of the dataset which leads to increase in accuracy of the system over time. The results are compared using confusion matrix, heat maps and processing time with previous systems [10].

Wong in 2011 proposed a methodology to detect and continuously recognize faces from a video input improving the detection of faces over various pose variations and image qualities. It continuously captures frames over the video and chooses frames which are sharp enough to run the algorithm to detect human faces. The previous system was able to detect faces which are sharp and aligned perfectly over the image. This system proposed was able to align and rotate the faces and also could recognize blurry images thereby avoiding it for recognition. This system revolutionized face detection algorithms [11].

Yang proposed a learning-based approach to identify faces from very low-resolution images using a dataset which consists of both very low resolution (VLR) and high resolution (HR) of facial data. Whenever the system manages to detect a VLR facial data from its dataset, it adds the facial data of the nearby frames to the dataset, thereby increasing accuracy of the system. The dataset was designed by collecting VLR and HR facial data of same image and converting them to a matrix of facial features. The matrices are mapped together using a transformation function [12].

3 Proposed System

3.1 Target Image Acquisition

The image of the target, which is the missing person or the wanted criminal, is received as input over a front-end web application which securely connects over a network used by the police department. The target's image is sent to generate multiple images of the target's face by applying all combinations of face features such as face masks, glasses, facial hair, face marks, beard and moustache of all possibilities. These images are replicated over multiple combinations by altering the contrast and resolution over multiple environments, thereby producing a good amount of target's image data.

3.2 Face Recognition

The face recognition module uses real-time CCTV data from the server and captures frames from the video data to recognize faces from the data. As it is not feasible to capture all the frames from the video, it consistently uses a single frame from one second of the video data to detect faces. The faces are compared with the dataset consisting of target image replications. The target dataset is updated real time as images of targets are updated to the database. The following detailed face recognition technique by Adam Geitgey is explained below.

HOG. Histogram of oriented gradients (HOG) algorithm is used to encode images to create a simplified version of the image. It was first created by Navneet Dalal and Bill Triggs. The gradient difference between nearby pixels is used to create an arrow-only image using this algorithm, which will increase the pace to detect a face in the image by comparing the gradient-oriented data of many other faces using a RNN. The HOG version of an image and the detected face is displayed below in Fig. 1.

Pose identification. After the face is detected, the posture of the face is identified by laying a landmark on the features of the face such as chin, eyes, nose and mouth enclosing. Once the landmarks are identified, the pose is adjusted such that the image

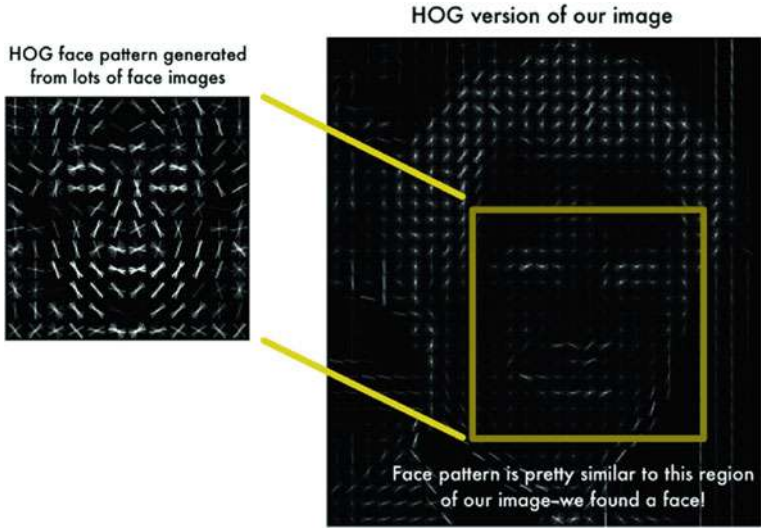


Fig. 1 HOG version of an image to detect face

faces front and nose and mouth come to the centre. This results in a perfectly centred face image of the target. This technique is depicted as a sequence of images showing how the landmarks are placed and used in Fig. 2.

Feature measurements. The features are measured, and a 128-size matrix consisting of measurements is generated by the neural network by using the score for the facial features in each hidden layer. This matrix is used to compare faces of the targets and see whose measurements are closer to the target and return the result.

This is depicted in Fig. 3.

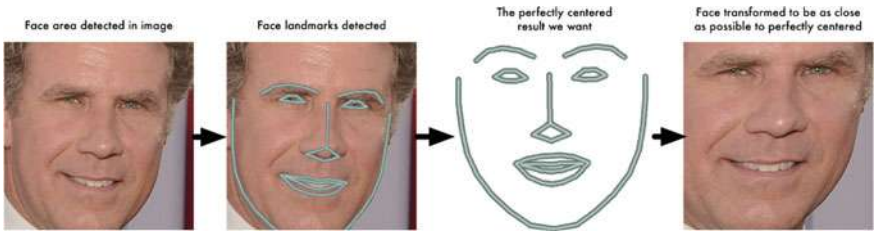


Fig. 2 Image landmarks used to centre a face

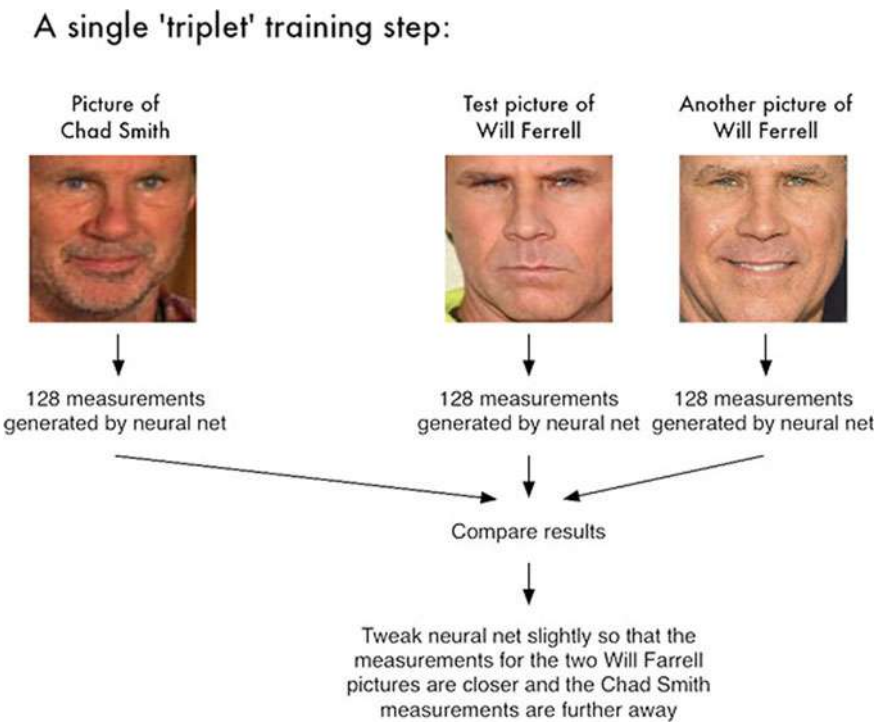


Fig. 3 128 measurements obtained from the face

3.3 Output

Whenever any of the targets' face is recognized, the target's details along the camera feed details are passed to the police department of the sector as per the camera's location, which further helps them to track the criminal or missing person immediately. The detection of target's face leads to the networking module which uses the API over the cloud server to send alerts to the police department via short message service and via e-mail. Moreover, a special message passing service is maintained which will alert the senior police in-charge through audio devices like walkie-talkie to alert them if they are on duty. This intimates them to check the location, where the target was most recently detected and can be tracked easily.

3.4 Deploying in Cloud Services

Cloud services such as AWS, Azure can be used to deploy this project and connect it with the front-end user interface to accept and feed data to the dataset. The neural

network runs as fast as possible, and the target's multiple replications are generated faster using the cloud services. This service helps to boost up the speed of the face recognition module to the maximum and help them tackle the target as soon as possible.

The detailed architecture of the proposed system is presented in Fig. 4.

4 Results and Discussions

This system was tested using various images taken under various circumstances. The dataset consisted of various blurry images, low-resolution images, old images which could result in ageing problem in the target's image, photographs of various dimensions and size and various colour grading, saturation, contrast, etc. which were taken from photocopies, paper prints, news articles, posters, group photographs with previous CCTV footages of criminals and other missing people available over the Internet. This system managed to produce good results with various accuracy measures. The accuracies are listed in Table 1.

Thus, the accuracies for various types of images recognized under CCTV footages prove the feasibility of the system. This system can be further modified and can be used to recognize prohibited items and alert for the same using real-time CCTV footages. The results show that this system successfully identifies targets from CCTV snapshots providing the highest accuracy of all the other types. This system provides less accuracy than others for low-resolution images. The failed cases show that this system cannot identify faces more accurately in low-resolution and high-contrast images and thus the accuracy.

5 Conclusion

Thus, our system helps in tracking of missing people and wanted criminals real time by generating multiple replications of the target image and using them to recognize faces from real-time CCTV captures with the help of neural network deployed over the cloud services. As this system is faster compared to previous systems because of the advanced face recognition algorithm and cloud services, we are sure this will help the police department if implemented successfully.

5.1 Future Enhancements

This system can be further enhanced to identify weapons or other illegal objects through CCTV cameras.

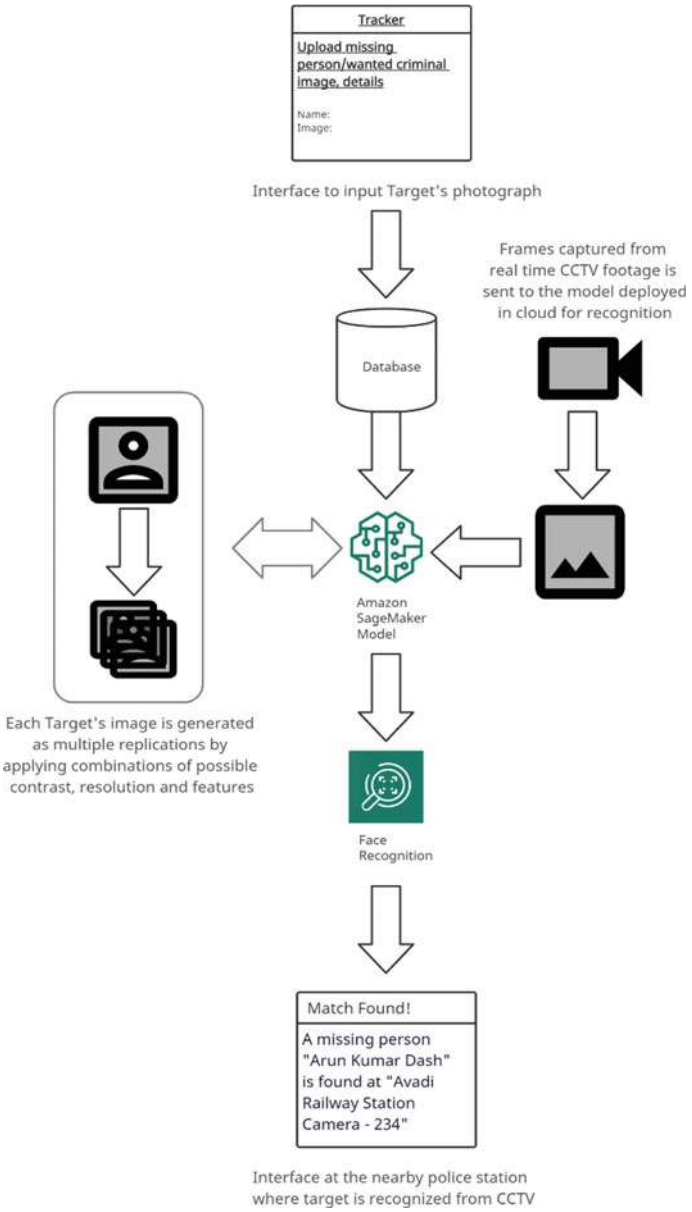


Fig. 4 Proposed architecture

Table 1 Accuracies produced by various types of images

Type of image	No. of different targets	Number of targets recognized	Accuracy (In %)
High contrast	30	20	66.67
Photographs from various sources	120	97	80.83
Low resolution	35	19	54.28
CCTV snapshot	650	580	89.23

References

1. Ayyappan S, Matilda S (2020) Criminals and missing children identification using face recognition and web scrapping. In: 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, pp 1–5
2. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 886–893
3. Geitgey A (2016) Machine learning is fun! part 4: modern face recognition with deep learning. In: Medium. Medium Corporation 24
4. Han M, Sethi A, Hua W, Gong Y (2004) A detection-based multiple object tracking method. In: 2004 international conference on image processing, ICIP'04, vol 5. IEEE, pp 3065–3068
5. Jain AK, Klare B, Park U (2011) Face recognition: some challenges in forensics. In: 2011 IEEE international conference on automatic face & gesture recognition (FG). IEEE, pp 726–733
6. Jose EM, Greeshma M, Haridas TP, Supriya MH (2019) Face recognition based surveillance system using facenet and mtcnn on jetson tx2. In: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS). IEEE, pp 608–613
7. Klontz JC, Jain AK (2013) A case study of automated face recognition: the boston marathon bombings suspects. Computer 46(11):91–94
8. Papaioannou S, Markham A, Trigoni N (2016) Tracking people in highly dynamic industrial environments. IEEE Trans Mob Comput 16(8):2351–2365
9. Shirsat S, Naik A, Tamse D, Yadav J, Shetgaonkar P, Aswale S (2019) Proposed system for criminal detection and recognition on CCTV data using cloud and machine learning. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). IEEE, pp 1–6
10. Son NT, Anh BN, Ban TQ, Chi LP, Chien BD, Hoa DX, Thanh LV, Huy TQ, Duy LD, Khan MHR (2020) Implementing cctv-based attendance taking support system using deep face recognition: a case study at fpt polytechnic college. Symmetry 12(2):307
11. Wong Y, Chen S, Mau S, Sanderson C, Lovell BC (2011) Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In: CVPR 2011 workshops. IEEE, pp 74–81
12. Yang M-C, Wei C-P, Yeh Y-R, Frank Wang Y-C (2015) Recognition at a long distance: very low resolution face recognition and hallucination. In: 2015 International Conference on Biometrics (ICB). IEEE, pp 237–242

Chapter 54
Determination of Avalanche Effect to Compute the Efficiency of Association Rule Hiding Algorithms

Dinesh Audichya, Prashant Sharma, and Pankaj Kumar Vaishnav

1 Introduction

Preserving the privacy of data is an important aspect of data mining. Today an enormous amount of data is generated from several data sources. Generated data is further taken for analysis; it contains sensitive as well as non-sensitive information. If a transactional dataset is collected from the retail store for analysis purposes, it consists of several types of correlations among the purchased items. Association rule mining is the way to find important inferences and relations from the dataset. It finds the items in the dataset which exist together or correlated with other items. In the hiding process of sensitive items and their relationship, any correlation or inference clues are completely hidden, so that these types of evidence could not be exposed by any third party or miner. Sensitive information needs to be hidden. It is necessary to keep the confidentiality up to a certain level and also to keep the usefulness of the data which is to be mined by a miner. Various techniques have been developed to hide sensitive information from data mining. Association rule hiding is a popular technique to stop exposure of sensitive data and relations during data sharing. It preserves the confidentiality of rules of the desired specification. These techniques mainly use the concept of support and confidence to alter the data for sensitive rule hiding.

D. Audichya (✉) · P. Sharma · P. K. Vaishnav
Pacific (PAHER) University, Udaipur 313003, India
e-mail: dineshaudichya@gmail.com

P. Sharma
e-mail: prashant.sharma@pacific-it.ac.in

P. K. Vaishnav
e-mail: panvas23@pacific-it.ac.in

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3_55

1.1 Support

It measures the occurrence of item or item sets in the transactions with respect to all the transactions. The support of A with respect to the transaction database T can be calculated as below:

- $|A|$ denotes a total number of transactions containing item A.
- $|N|$ denotes a total number of transactions, then

$$\text{Support}(a) = \frac{\text{count}(A)}{|N|}$$

1.2 Confidence

It is a probability of item sets present together in the transactions. It can be calculated as a proportion of number of transactions having both item sets A and B for the rule $A \rightarrow B$ with respect to the support of B.

Confidence can be defined for the rule $A \rightarrow B$ as:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

Similarly, confidence can be defined for the rule $B \rightarrow A$ as:

$$\text{Confidence}(B \rightarrow A) = \frac{\text{support}(A \cup B)}{\text{Support}(B)}$$

1.3 Avalanche Effect

It is the proportion of the no. of bits changed in the sanitized database and the total no. of bits present in the database. If any rule hiding algorithm changes 5 bits out of total 100 bits in order to hide the rule, then avalanche effect of it would be 5%.

Market basket analysis is a widely used approach for data analysis in marketing; main aim is to find items that items are bought together so that a marketing strategy could be built to promote the selling of targeted items by considering the fact of positive correlation with another item or with a set of items. For such type of analysis, apriori algorithm is used. It was introduced by Agarwal [1] in 1994. It uses a bottom-up approach to find the associations. In hiding algorithms after frequent item sets mining, sensitive associations are identified by using the criteria of support and confidence. After finding sensitive association rules, the next step is to apply the hiding technique before exposing the transactional database to the third party

for further mining. There are many techniques available for preserving the privacy of sensitive association rules. Some popular techniques were taken as a base for the development of new algorithms in order to provide better output. Various association rule hiding approaches have been developed to maintain the privacy of data by decreasing the support and confidence of the data for given values. Charu et al. [2] focused the study on preservation of privacy in data mining domain. Multiple surveys were done for quantification of mining algorithms and transformation of the data for mining. Various rule hiding algorithms were compared and compared the results with chosen mining algorithms. Modi et al. [3] proposed the rule hiding algorithm which decreases support of R.H.S. item of rule clusters (DSRRC), it promised hiding rules without affecting data in large amount and sanitize the databases for larger number of association rules. Zala [4] performed a comparative study of DSR, ISL, and new variable hiding counter algorithm. An analysis on advantages and limitation of these three algorithms was done and found new variable hiding counter algorithm performed better.

Shah et al. [5] proposed two algorithms, advanced decrease support of R.H.S. items of rule cluster (ADSRRC) and remove and reinsert L.H.S. of rule (RRLR) for hiding sensitive association rules and to overcome the limitation of decrease support of R.H.S. items of rule cluster (DSRRC). Both algorithms are analyzed concerning hiding failure, misses cost, artifactual patterns, dissimilarity, and completeness. Natarajan et al. [6] implemented an association rule hiding algorithm which would be efficient in providing confidentiality and improved the performance at the time when the database stores and retrieves the huge amount of data. The proposed algorithm was compared with existing algorithms ISL and DSR. The proposed algorithm hides the association rule on both sides (LHS and RHS) in comparatively reduced time. Maja et al. [7] measured lift and confidence at separate timing as interestingness measures and discussed the advantage of the association rules at various temporal values that helped Web admin for better management of the websites. Jadav et al. [8] performed study on various heuristic approach for the hiding of rules, and described the challenges in hiding rules and novel approaches for the challenges. Vijayarani et al. [9] proposed an algorithm based on a genetic algorithm for association rule hiding. Hiding techniques were described like ISL, DSR, genetic algorithm, Tabu search, ant colony, and dummy items creation algorithms for rule hiding. Genetic algorithm was modified to enhance its performance in which modification was done into existing transactions only. In this paper, a mutation was used instead of a crossover operation in order to avoid the generation of new transactions. Yewale et al. [10] proposed a new algorithm for hiding association in order to maintain privacy, algorithm modified item values in fewer places with an aim to hide most number of rules. It performed better than classical algorithm for transactional database without compromising the quality of original data. Kaur et al. [11] discussed various approaches for association rule hiding approaches like heuristic-based approaches, reconstruction -based approach, and cryptography-based approach. Existing algorithms increase support of LHS (ISL) and decrease support of RHS (DSR), association rule were hidden using a hidden counter, fast hiding sensitive association rules (FHSAR),decrease support of R.H.S items in rule clusters (DSSR), and modified decrease support of R.H.S item

of rule clusters (MDSRRC) were discussed and found that neither of these techniques provides data quality to protect sensitive association rule. Upadhyay et al. [12] provided a survey of association rule hiding methods for privacy preservation. Various algorithms have been designed for it in recent years. This paper summarized them and surveyed current existing techniques for association rule hiding. A comparative study, including the advantages and limitations of each technique, was done. Bhatt et al. [13] proposed a new algorithm based on two classical algorithms ISL and DSR, they found the new algorithm performance better in terms of CPU time taken for hiding the sensitive rules. Singh [14] developed a tool to apply apriori Algorithm and data available in MS excel format. It allows setting various measures such as support, confidence, and number of rules to be mined. It is useful for mining association rules from a given dataset. Kulkarni et al. [15] discussed association rule mining and apriori algorithm for the mining of healthcare databases. They used the apriori and FP growth algorithm for finding a relation between symptoms and viral disease. Pathak et al [16] proposed two algorithms DSRFASTPREDICTIVE and ISLFASTPREDICTIVE, respectively, it was developed to perform better than DSR and ISL, respectively. The results were showing increased performance gain in the proposed algorithms. Surendra [17] proposed the sanitizing algorithm based on value distortion in which closed item sets were sanitized in place of transactions of database. They found it scalable for the database without changing relational attributes of the items. The results were better in terms of modifications in the database with compared to transaction modifications. Sulova [18] proposed the use of association rule mining in team selection of an IT company. RapidMiner was used for that purpose to implement different algorithms on the database in order to make the best decision. Kholod [19] explained the methods of applying data mining techniques, mainly market basket analysis for finding the associations between the brands and their respective stores in Japan. The researcher confirmed the usefulness of the result in order to decide marketing strategy for stores. Ameta [20] proposed selective flip bit method for hiding sensitive association rules. This method was used to rank the sensitive association rules after those sensitive associations were hidden.

1.4 Association Rule Hiding Approaches

Association rule hiding approaches can be categorized into five types. In this research work, Heuristic approach for association rule hiding is used; it is further divided into categories. Heuristic approach provides basically two types of technique, i.e., data distortion technique and data blocking technique.

Heuristic: These approaches generate efficient, scalable, and faster results. It generates side effects like lost rules, Artifactual patterns in output database which is modified by the algorithms.

Cryptographic: It is used in a distributed environment or in case of multiple parties having different parts of the dataset. Mining is done in a highly secured manner.

Border Based: Modification is done on border of frequent and infrequent rules. It adds considerable computation cost.

Reconstruction: It is better than heuristic approach but has higher complexity.

Exact: It provides the best solution without modifying the database much and has very few side effects. On the other hand, it has a very high computation cost.

1.5 Data Distortion Technique

In this, input data is available in the form of bits. Each bit can be either a 0 or 1 only. This technique modifies the bits into 0 to 1 or from 1 to 0 for achieving the values of support and/or confidence less than the given limits. In this method, support and/or confidence are decreased or increased in process of the blocking the reachability to sensitive rules.

1.6 Data Blocking Technique

This approach also takes input database similar to data distortion, i.e., in bits which could be either a 0 or 1. The main difference is that replacement of either 0 or 1 is done with an unknown value represented as '?' or '*'. It makes it difficult for the consumer of the output database to retrieve the original value by reverse technique. This approach provides higher data privacy.

1.7 ISL

In this algorithm, support of left-hand side of the rule is increased in order to hide the rule from the output database. This approach has a major inbuilt drawback that it makes changes only on right-hand side of the association rule in attribute values. It is not applicable to both sides of the rule. It has a very less computational cost but it might add new rules in the output database as it increases the support, which creates new issues in the hiding process. Limitations of ISL are seen when the algorithm cannot hide the rule even after changing every bit of the LHS of the rule. In such cases, DSR is useful. As it is easy to implement, it is a popular choice for researchers to modify this algorithm and design a new one.

1.8 DSR

This algorithm deals only with right hand side (RHS) of the rules; it decreases the presence of right-side attribute value in the transactions. Like ISL, this algorithm works on one side of the rule only. The basic principle is that decreasing the dividend in the formula of confidence will decrease the value of confidence. It is achieved by flipping the bits from '1' to '0' where RHS item and LHS item are present together in the transaction. It works faster than ISL but has side effects like hiding of other useful association rules. Its side effects increase with the number of items are present in the rule set which in turn yields a high degree of difference in the actual and sanitized database. Also, it hides many useful rules which are not sensitive.

1.9 Selective Flip Bit (SFB)

This method was proposed to overcome the limitations of ISL and DSR. In this method, bits are changed from '1' to '0' for hiding the sensitive rules from the database. The advantage of this method is that it can work on both sides of item attributes after calculation of sensitivity. Prior to this, bits to be flipped are taken into account after measuring the sensitivity among sensitive rules of the transactional database. For keeping the sensitive rules out of the discovery to a third party, it flips the bits intelligently by changing them from '1' to '0' only, it does not touch bits having value as '0'. The advantage of this method, it does not add new artifactual rules like ISL does to hide rules. This method generates a very few numbers of lost rules, false rules, and ghost rules.

2 Research Methodology

The output of ISL, DSR, and selective flip bit method algorithms is collected, and results are compared with each other. For this, same set of sensitive rules and the same transactional database were taken as an input for each rule hiding algorithm. In this paper, avalanche effect is calculated for each algorithm and research supports the algorithm which outputs the minimum value of avalanche effect. A real-time data from a supermarket was collected for analysis. After preprocessing, a relevant dataset was extracted. This dataset selected was consisting of details of 221 transactions and 16 items. After that, WEKA tool was used to determine the best rules at different minimum confidence levels.

The following Table 1 gives aggregation information about the database.

In this paper, avalanche effect is calculated for each algorithm, and the paper supports the algorithm which outputs the minimum value of avalanche effect.

Table 1 Summary of the retail database

Entity	Value
Total transactions	221
Total items	16
Total bits	2336

Flowchart for calculating the avalanche effect is presented to describe the operations at each step (see Fig. 1). As shown, no. of bits are compared before and after the sanitization of rules, and ratio of modified and total bits is obtained as an avalanche effect. Minimum confidence values were supplied one after another as 90, 80, 70, and 60% to get the best rules in the provided criteria, and results were received as per Table 2. Following rules were marked as sensitive.

Thereafter, a tool was developed for hiding sensitive association rules as per Table 2. In this tool, item values of the database are represented as '0' or '1' to represent

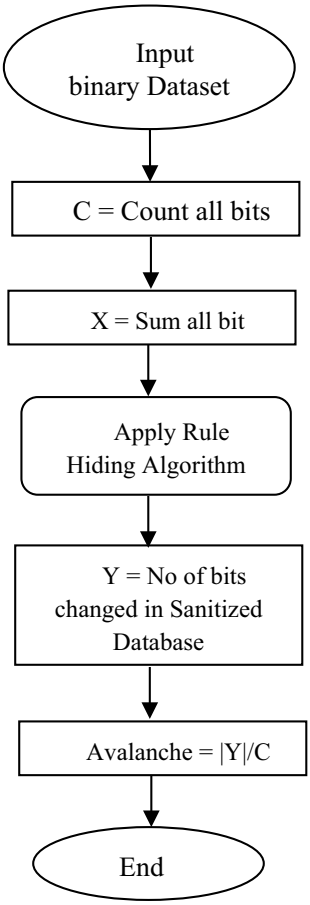


Fig. 1 Calculation of avalanche effect

Table 2 Sensitive rules chosen for hiding

Sr. No	Rule set	Confidence (%)	Notation
1	Chana Dal, Saras Pure Ghee, Sugar => Poha Sahakar Sing Dana => Sugar	90	RS_{90}
2	Chana Dal, Saras Pure Ghee, Sugar => Poha Sahakar Sing Dana => Sugar	80	RS_{80}
3	Poha => Sugar Sugar => Poha	70	RS_{70}
4	Poha => Sugar Sugar => Poha	60	RS_{60}

the absence and presence of an item in transactions. These values are referred to as bits for experimental implementation.

Another tool was developed to determine the avalanche effect from the results received for association rule hiding algorithms like ISL, DSR, and SFB method. Results from each algorithm were collected and analyzed for the comparison to find the most efficient one. The least avalanche effect leads to fewer changes in the databases during the process of association rule hiding. Lesser changes lead to good database quality along with hiding all marked sensitive association rules.

3 Result Analysis

The output of each algorithm is successfully received and noted for given rule sets and confidence values, in the end, result of each algorithm is compared. The same set of inputs is supplied to each algorithm evaluation. For the first set of rules in Table 3, ISL modified three bits, DSR modified two bits, and selective flip bit method modified two bits; thus, their avalanche effect values were calculated as 0.08%, 0.06%, and 0.06%, respectively. According to Table 4 where confidence was 80%, ISL algorithm modified 16 bits; whereas, DSR and selective flip bit modified 11 and 9 bits, respectively. Hence, these two algorithms yield very little avalanche effect, i.e., 0.31% and 0.25%, respectively, which denotes their superior efficiency concerning ISL which has avalanche effect value of 0.45%. In Table 5, for a confidence value as 70%, ISL, DSR, and SFB modified 48 bits, 44 bits, and 42 bits, respectively. Avalanche effect of SFB is smallest with the comparison of other two algorithms. SFB

Table 3 Bits modified with avalanche effect to hide rule at minimum confidence = 90%

Rule set	Confidence (%)	Algorithm	Bits modified	Avalanche effect (%)
RS_{90}	90	ISL	3	0.08
RS_{90}	90	DSR	2	0.06
RS_{90}	90	RS_{60}	2	0.06

Table 4 Bits modified with avalanche effect to hide rule at minimum confidence = 80%

Rule set	Confidence (%)	Algorithm	Bits modified	Avalanche effect (%)
RS_{80}	80	ISL	16	0.45
RS_{80}	80	DSR	11	0.31
RS_{80}	80	RS_{60}	9	0.25

Table 5 Bits modified with avalanche effect to hide rule at minimum confidence = 70%

Rule set	Confidence (%)	Algorithm	Bits modified	Avalanche effect (%)
RS_{70}	70	ISL	48	1.35
RS_{70}	70	DSR	44	1.24
RS_{70}	70	RS_{60}	42	1.19

Table 6 Bits modified with avalanche effect to hide rule at minimum confidence = 60%

Rule set	Confidence (%)	Algorithm	Bits modified	Avalanche effect (%)
RS_{60}	60	ISL	58	1.64
RS_{60}	60	DSR	47	1.33
RS_{60}	60	RS_{60}	35	0.99

yields avalanche effect value of 1.19%; whereas, ISL and DSR give avalanche effect as 1.35% and 1.24%. Thus, SFB performed much better than other two algorithms. In the last variation of input parameters for the confidence value of 60%, from Table 6 it can be observed that ISL made the highest number of modifications of 58 bits, DSR made changes of 47 bits and selective flip bit made changes of 35 bits only. As a result, selective flip bit yields avalanche effect as 0.99% only; whereas, ISL and DSR yield 1.64% and 1.33%, respectively.

4 Conclusion

From the analysis of the output, it shows that SFB always performed better than ISL and DSR for each set of rules and confidence value as avalanche effect of ISL was the highest in each comparison table. From the result of Tables 3, 4, 5, and 6, it is concluded that DSR performed moderately but SFB outperformed DSR and ISL in each set of rules and varied confidence values. An avalanche effect of SFB was proven as the lowest among all three algorithms. This makes its performance better and efficient as compared to the other two algorithms.

5 Future Work

In order to enlarge the comparison, a more classical and modern rule hiding algorithms can be taken for evaluating the performance in a wide spectrum, and one can find results to decide the better algorithm for the best fit that keeps the avalanche effect minimum. Future work can be done by adding DSRRC, ADSRRC, and RRLR rule hiding algorithms and their results into the comparison.

Acknowledgements We thank all the researchers and the author of the research papers referred for this study which helped to understand and experiment the techniques in a very effective way. Also, special thanks to Pacific University, CSE, Udaipur, for providing the required laboratory facilities and well-qualified guide Prashant Sharma who gave me valuable knowledge and direction in this research.

References

1. Agarwal S (2000) Privacy-preserving data mining. In: ACM SIGMOD, pp 439–450
2. Charu CA, Philip SY (2008) Privacy-preserving data mining models and algorithms, vol 34. Springer Science Business Media, New York
3. Modi CN, Rao UP, Patel DR (2010) Maintaining privacy and data quality in privacy preserving association rule mining. In: Seventh international conference on machine learning and applications, vol 2. Springer International, Switzerland, pp 1–6
4. Zala K (2012) Comparison of ISL, DSR, and new variable hiding counter algorithm of association rule hiding. Int J Sci Eng Res: IEEE 3(5)
5. Shah K, Thakkar A, Ganatra A (2012) Association rule hiding by heuristic approach to reduce side effects & hide multiple R.H.S. items. Int J Comput Appl 45
6. Natarajan R, Sugumar R, Mahendran M, Anbazhagan K (2012) Design and implement an association rule hiding algorithm for privacy preserving data mining. Int J Adv Res Comput Commun Eng: IEEE 1
7. Maja D, Tanja K (2013) Association rules for improving website effectiveness: case analysis. Online J Appl Knowl Manage 1(2)
8. Jadav K, Vania J (2013) A survey on association rule hiding methods. Int J Comput Appl. ISSN: 0975-8887, vol 82
9. Vijayarani S., Tamilarasi A (2014) An efficient technique for hiding association rules in privacy preserving data mining, vol 12. Shodhganga, India, pp 141–163
10. Yewale A, Rajput Y, Shirapure S, Patel H (2014) Privacy preserving association rule mining in retail industries. Int J Adv Res Comput Commun Eng
11. Kaur K, Bansal M. A review on various techniques of hiding Association rules in privacy preservation data mining. IJECS 4(6):12947–12951
12. Upadhyay N, Tripathi K, Mishra A (2015) A survey of association rule hiding approaches. Int J Comput Sci Inf Technol Security 5(1). ISSN: 2249-9555
13. Bhatt N, Patel M (2015) Updated hybrid ISL and DSR based technique for protecting sensitive information. Int J Innov Res Technol 2(6)
14. Singh G, Jassi S (2017) A review paper: a comparative analysis on association rule mining algorithms. Int J Recent Technol Eng 6(2):2277–3878
15. Kulkarni AR, Shivaji D, Mundhe SD (2017) Data mining technique: an implementation of association rule mining in healthcare. Int Adv Res J Sci Eng Technol 4(7)
16. Pathak K, Silkari S, Chaudhari NS (2017) Privacy preserving informative association rule mining. Int J Appl Inf Syst 12. ISSN 2249-0868

17. Surendra H, Mohan HS (2018) Distortion-based privacy-preserved association rules mining without side effects using closed itemsets. In: Emerging technologies in data mining and information security. vol 813. Springer, Berlin, pp 591–601
18. Sulova S (2018) Association rule mining for improvement of IT Project management. TEM J 7(4):717–722
19. Kholod M (2018) Market basket analysis of convenience store POS data. In: Science reports. Tohoku University. vol 71, pp 61–82
20. Ameta G, Bhatnagar D (2019) Development of an association rule hiding algorithm for privacy preserving in market basket databases. Int J Innov Technol Exploring Eng 8(6S)

Author Index

M. Pandit et al. (eds.), *Artificial Intelligence and Sustainable Computing*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-19-1653-3