ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/359021834

Proceedings of international conference on computational intelligence and Data Engineering iccide 2021

Conference Paper · March 2022

CITATIONS
0

READ
1

3 authors:



AKANDE NOAH OLUWATOBI
Landmark University
56 PUBLICATIONS **188** CITATIONS
SEE PROFILE



Oluwakemi Christiana Abikoye
University of Ilorin
90 PUBLICATIONS **249** CITATIONS
SEE PROFILE



Akande Hakeem Babalola
University of Ilorin
11 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Character Recognition View project

Project Social Media View project

All content following this page was uploaded by Akande Hakeem Babalola on 05 March 2022.

The user has requested enhancement of the downloaded file.

Lecture Notes on Data Engineering and Communications Technologies 99



Nabendu Chaki
Nagaraju Devarakonda
Agostino Cortesi
Hari Seetha *Editors*

Proceedings of International Conference on Computational Intelligence and Data Engineering

ICCIDE 2021

Springer

Lecture Notes on Data Engineering and Communications Technologies

Volume 99

Series Editor

Fatos Xhafa, Technical University of Catalonia, Barcelona, Spain

The aim of the book series is to present cutting edge engineering approaches to data technologies and communications. It will publish latest advances on the engineering task of building and deploying distributed, scalable and reliable data infrastructures and communication systems.

The series will have a prominent applied focus on data technologies and communications with aim to promote the bridging from fundamental research on data science and networking to data engineering and communications that lead to industry products, business knowledge and standardisation.

Indexed by SCOPUS, INSPEC, EI Compendex.

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

More information about this series at https://link.springer.com/bookseries/15362

Nabendu Chaki · Nagaraju Devarakonda ·
Agostino Cortesi · Hari Seetha
Editors

Proceedings of International Conference on Computational Intelligence and Data Engineering

ICCIDE 2021

Springer

Editors
Nabendu Chaki
University of Calcutta
Kolkata, India

Nagaraju Devarakonda
VIT-AP University
Amaravati, India

Agostino Cortesi
Ca' Foscari University
Venice, Italy

Hari Seetha
VIT-AP University
Amaravati, India

ISSN 2367-4512 ISSN 2367-4520 (electronic)
Lecture Notes on Data Engineering and Communications Technologies
ISBN 978-981-16-7181-4 ISBN 978-981-16-7182-1 (eBook)
https://doi.org/10.1007/978-981-16-7182-1

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Preface

This volume is devoted to the quality contributions from the authors and invited speakers of the 4th International Conference on Computational Intelligence and Data Engineering (ICCIDE 2021) held during August 13 and 14, 2021, at VIT-AP University. This conference brought together 400 participants from various academic institutes and industries across the globe. This two-day international conference has been organized by the Department of Computer Science and Engineering of the VIT-AP University.

The academic programs of the conference commenced with the keynote address by Prof. Rolf Drechsler, German Research Center for Artificial Intelligence, University of Bremen, Bremen, Germany, on "Proving Correctness Efficiently: Polynomial Formal Verification of Arithmetic Circuits."

Prior to this, ICCIDE 2021 has been inaugurated in the presence of the Hon'ble Chief Guest Dr. Prem Prakash Jaya Ram, Head Digital Innovation Lab; Guest of Honor Sri Krishna Thiagarajan, Senior Vice-President, Virtusa; and Guest of Honor Dr. G. Viswanathan, Founder and Chancellor, VIT.

The Hon'ble Chief Guest Dr. Prem Prakash Jayaz Ram addressed the audience by providing an insight to the initiatives of the conference. The Guest of Honor Dr. Sri Krishna Thiagarajan and Dr. G. Viswanathan also addressed the audience by improvising the need of a conference in shaping research practices. This conference was also addressed by Dr. Sekar Viswanathan, Vice-President, VIT; Dr. S.V Kota Reddy, Vice-Chancellor, VIT-AP University; and Dr. CLV Sivakumar, Registrar, VIT-AP University. The inaugural session also witnessed the addresses by eminent speakers. Dr. Nagaraju Devarakonda, convener of ICCIDE 2021, VIT-AP University, addressed that constant updation of technical knowledge in new research areas is necessary and also explained about the schedule of conference.

A thorough peer-review process has been carried out by the PC members and associates. While reviewing the papers, the reviewers mainly looked at the novelty of the contributions, besides the technical content, the organization, and the clarity of the presentation. While nearly 500 articles across different themes of the conference were received from 18 countries across the globe, only 35 papers and 1 invited paper from Prof. Rolf Drechsler, IEEE Fellow from Germany, were finally selected

for presentation and publication in this post-conference proceedings. These figures themselves reflect the high quality and standard of the research presented in ICCIDE 2021.

This conference had five presidential keynote speakers from various universities across the globe.

Keynote-1: Prof. Rolf Drechsler, University of Bremen, Germany
Keynote-2: Prof. Gerhard W. Dueck, University of New Brunswick, Canada
Keynote-3: Prof. Kiarash Bazargan, University of Minnesota, USA
Keynote-4: Dr. Anand Nayyar, Duy Tan University, Vietnam
Keynote-5: Prof. Robert Wille Linz, Austria

We thank all the members of the Program Committee for their excellent and time-bound review work. We are thankful to the entire management of VIT-AP University for their warm patronage and continual support to make the event successful. We especially thank Dr. G. Viswanathan, Founder and Chancellor, VIT, for his inspiring presence whenever we approached him for his advises. We sincerely appreciate the parental role of Dr. S. V. Kota Reddy, Vice-Chancellor, VIT-AP University.

We appreciate the initiative and support from Mr. Aninda Bose and his team in Springer Nature for their strong support toward publishing this volume in Springer Nature. Finally, we thank all the authors without whom the conference would not have reached the expected standards.

Kolkata, India — Dr. Nabendu Chaki
Amaravati, India — Dr. Nagaraju Devarakonda
Venice, Italy — Dr. Agostino Cortesi
Amaravati, India — Dr. Hari Seetha

Contents

Genetic Algorithm-Based Vaccine Optimization Technique to Control COVID-19 1
V. Ajantha Devi, Mohd Naved, E. Gangadevi, and Anand Nayyar

Ethereum MongoDB: Integrating Blockchain with Non-relational Databases 17
Sneha Sarkar, Ramesh Dharavath, and Hemika Jadav Badavath

Attention-Based Discrimination of Mycoplasma Pneumonia 29
Tathagat Banerjee, S. Karthikeyan, Amandeep Sharma, K. Charvi, and Sruti Raman

Minimum Weight Dominating Set Using Chemical Reaction Optimization Algorithm 43
Pritam Khan Boni and Md. Rafiqul Islam

A Hybrid Domain Fragile Watermarking Technique for Authentication of Color Images 57
Muzamil Hussan, Shabir A. Parah, Aiman Jan, and G. J. Qureshi

IoT-based Carbon Monoxide Monitoring Model for Transportation Vehicles 65
I. Etier, A. Anci Manon Mary, and Nithiyananthan Kannan

TWEERIFY: A Web-Based Sentiment Analysis System Using Rule and Deep Learning Techniques 75
Oluwatobi Noah Akande, Enemuo Stephen Nnaemeka, Oluwakemi Christiana Abikoye, Hakeem Babalola Akande, Abdullateef Balogun, and Joyce Ayoola

Machine Learning Based Fake News Detection on Covid-19 Tweets Data 89
Vishesh Mehta and Ram Krishn Mishra

Adverse Effect of Electromagnetic Frequency on Human Being Health Under Real-life Scenario: A Comparative Study 97
Gaurav Gandhi and Sharad Sharma

Consistency Driven Cross-Modal Twitter Sentiment Analysis and Opinion Mining 111
P. Agalya, S. Ashvathe, R. Harsini, and S. Chitrakala

A Blockchain-Based Document Verification System for Employers 123
Kshitij Dhyani, Jahnavi Mishra, Subhra Paladhi, and I. Sumaiya Thaseen

Railroad Segmentation Using Modified UNet 139
R. S. Rampriya, Sabarinathan, R. Suganya, S. Babysruthi, Pitchuka Yamini, and Pernamitta Hanuma Reddy

Weight Pruning in Convolutional Neural Network for Efficiency in Plant Disease Detection 151
Priyang Patel, Madhav Tripathi, Mohammed Husain Bohara, and Parth Goel

An Edge Computing-Assisted Internet of Things-Based Secure Smart Home 159
S. Narasimha Swamy, P. Shivanarayanamurthy, Gaurav Purohit, and Kota Solomon Raju

Onion Price Prediction Using Machine Learning Approaches 175
Kunda Suresh Babu and K. Mallikharjuna Rao

Development of an IoT-Based Bird Control System Using a Hybrid Deep Learning CNN-SVM Classifier Model 191
R. PushpaLakshmi

Social Media Analysis for Flood Nuggets Extraction Using Relevant Post Filtration 201
K. P. Aarthy, Kiruthya Ramesh, Vishnu Priya Prasanna Venkatesh, S. Chitrakala, and Shri. C. M. Bhatt

Fog Clustering-based Architecture for Load Balancing in Scientific Workflows 213
Mandeep Kaur and Rajni Aron

Multi-font Curved Text Scene for Enhanced OCR Readability 223
P. M. Jena, B. K. Tripathy, and S. K. Parida

Intelligent Path Planning with Improved BFS (I-BFS) Strategy for Mobile Robot in RFID Equipped Unknown Environment 237
Prabin Kumar Panigrahi, Sukant Kishoro Bisoy, and Hrudaya Kumar Tripathy

Application of Deep Learning in Honeypot Network for Cloud Intrusion Detection 251
S. B. Goyal, Pradeep Bedi, Shailesh Kumar, Jugnesh Kumar, and Nasrin Rabiei Karahroudi

Optimal Real-time Pricing and Sustainable Load Scheduling Model for Smart Homes Using Stackelberg Game Theory 267
Karthikeyan Dheeraja, R. Padma Priya, and Thakur Ritika

Unsupervised Spectral Feature Extraction for Hyperspectral Image Classification 285
Ratna Sravya Sajja and B. L. N. Phaneendra Kumar

Distributed Ledger Framework for an Adaptive University Management System 295
Yaswanth Lingam Sivadanam, R. Ramaguru, and M. Sethumadhavan

Fuzzy Model-Based Robust Clustering with Multivariate *t*-Mixture Distributions 307
Wajid Ali, Ishtiaq Hussain, and Miin-Shen Yang

Tackling COVID-19 Infodemic Using Deep Learning 319
Prathmesh Pathwar and Simran Gill

Automatic Blood Vessel Segmentation in Retinal Fundus Images Using Image Enhancement and Dynamic Gray-Level Thresholding 337
J. Jeslin Shanthamalar and R. Geetha Ramani

Security of Digital Images Based on SHA-3 and Multiplication-Rotation-XOR (MRX) Operations over Galois Field 349
Abdul Gaffar and Anand B. Joshi

Multimodal Depression Severity Detection Using Deep Neural Networks and Depression Assessment Scale 361
H. V. Madhu Sudhan and S. Saravana Kumar

Machine Learning Model for Automated Sleep Scoring Based on Single-Channel EEG Signal Data 377
Santosh Satapathy and Hari Kishan Kondaveeti

Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders, and Attacks 387
Alekya Sai Laxmi Kowta, P. K. Harida, Shruti Varsha Venkatraman, Siddharth Das, and V. Priya

Asymmetric Cryptosystem for Triple Color Images Using 6D Hyperchaotic System and Equal Modulus Decomposition in Fractional Transform Domain 403
Dhanesh Kumar, Anand B. Joshi, and Sonali Singh

On the Evaluation and Implementation of LSTM Model for Speech Emotion Recognition Using MFCC 421
Sheetal U. Bhandari, Harshawardhan S. Kumbhar, Varsha K. Harpale, and Triveni D. Dhamale

Analyzing the Effects of Text Representations on the Performance of Document Clustering in Public Health Tweets 435
Aakansha Gupta and Rahul Katarya

Learning from Imbalanced Healthcare Data Using Overlap Pattern Synthesis 447
Lincy Mathews and Hari Seetha

Polynomial Formal Verification of Arithmetic Circuits 457
Rolf Drechsler, Alireza Mahzoon, and Lennart Weingarten

Author Index 471

About the Editors

Nabendu Chaki is a professor in the Department Computer Science and Engineering, University of Calcutta, Kolkata, India. He is the editor in chief of the Springer Nature book series on Services and Business Process Reengineering. Besides editing about 40 conference proceedings with Springer, Dr. Chaki has authored eight text and research books with CRC Press, Springer Nature, etc. He has published more than 200 Scopus Indexed research articles in Journals and International conferences. Prof. Chaki has served as a visiting professor in different places including US Naval Postgraduate School, California, and in different universities in Italy and Poland. He is the founder chair of ACM Professional Chapter in Kolkata and served in that capacity for three years since January 2014. He has been active during 2009-2015 toward developing several international standards in Software Engineering and Service Science as a Global (GD) member for ISO-IEC.

Nagaraju Devarakonda received his B.Tech. from Sri Venkateswara University, M.Tech. from Jawaharlal Nehru University (JNU), New Delhi, and Ph.D. from Jawaharlal Nehru Technological University, Hyderabad. He has published over 75 research papers in international conferences and journals. He is the co-editor of Proceedings of ICCIDE 2017 and 2018, 2020. The proceedings were published in Lecture Notes on Data Engineering and Communication Technologies of SPRINGER. He published papers in ICCI*CC 2017 at OXFORD UNIVERSITY, UK, and ICCI*CC 2018 at UNIVERSITY OF CALIFORNIA, BERKELEY, USA. He is currently working as an associate professor in the School of Computer Science and Engineering at VIT-AP University and has 17 years of experience in teaching. His research areas are data mining, soft computing, machine learning, and pattern recognition. He has supervised 25 M.Tech. students, guided 2 Ph.Ds, and currently guiding 9 Ph.Ds.

Agostino Cortesi PhD., is a full professor of Computer Science at Ca' Foscari University, Venice, Italy. He served as a dean of the Computer Science studies, as department chair, and as vice-rector for quality assessment. Currently, he holds the position of head of the Ph.D. program in Computer Science. His main research

interests concern programming languages theory, software engineering, and static analysis techniques, with particular emphasis on security applications. He has held the position of visiting professor at several universities and research centers, including University of Illinois, Ecole Normale Supérieure Paris, Ecole Polytechnique Palaiseau, IBM T.J.Watson Research Center, and University of Calcutta. He published more than 150 papers in high-level international journals and proceedings of international conferences. His h-index is 20 according to Scopus and 28 according to Google Scholar. He serves as co-editor in Chief of the book series "Services and Business Process Reengineering" edited by Springer Nature. At the Ca' Foscari University, Agostino holds the chairs of "Software Engineering," "Software Correctness, Security and Reliability," "Computer Networks and Information Systems" and "Data Programming."

Hari Seetha obtained her Master's degree from National Institute of Technology (formerly R. E. C.)Warangal and obtained Ph.D. from School of Computer Science and Engineering, V.I.T University. She worked on Large Data Classification during her Ph.D. She has research interests in the fields of pattern recognition, data mining, text mining, soft computing, and machine learning. She received Best paper award for the paper entitled "On improving the generalization of SVM Classifier" in Fifth International Conference on Information Processing held at Bangalore. She has published a few research papers in national and international journals and conferences. She has been one of the editors for the Edited Volume on "Modern Technologies for Bigdata Classification and Clustering" published by IGI Global in 2017. She is a member of editorial board for various International Journals. Three of her students submitted their Ph.D. thesis. She is presently guiding four Ph.D. students. She is listed in the 2014 edition of Who's who in the world published by Marquis Who's Who, as the biographical reference representing the world's most accomplished individuals. She had been a co-investigator to a major research project sponsored by the Department of Science and Technology, Government of India. She served as a division chair for Software Systems division and Program Chair for B.Tech (CSE) program, in the School of Computer Science and Engineering at VIT University, Vellore, India, as well as an assistant director (Ranking and Accreditation) at VIT University, Vellore. She is currently working as a professor and dean for the School of Computer Science and Engineering at VIT-AP, Amravati, India.

Genetic Algorithm-Based Vaccine Optimization Technique to Control COVID-19



V. Ajantha Devi, Mohd Naved, E. Gangadevi, and Anand Nayyar

Abstract Another way to reduce the effects of COVID-19 is to develop vaccines to build viral immunity. Many contemporary assessments concentrate on the state and promise of this vaccine system, with genetic algorithms for feature selection for the predictor being used as a replacement for the first data. While computing the distances to the preparation set examples, the predictors utilized in the count are the ones with no missing qualities for that example and no missing qualities in the preparation set. An intricating factor is that COVID-19 changes genetically and the indications are diverse which is developing it harder to focus on the peptides they contain to fit a sacked tree model for every indicator utilizing the preparation set examples. In this proposed method, we have applied the technique of genetic algorithm (GA) which finds the most vulnerable person who needs vaccine to save their lives. This searching process is taken through GA for better decision and applied for vaccine optimization. This study presents an attempt to distribute a genetic algorithm's population onto the nodes of a complicated network, with crossover and mutation processes limited to the population members. The study given here uses a certain form of genetic algorithm as a tool for balancing the exploration–exploitation trade-off in order to investigate a specific aspect of a network, and it fits into that category.

Keywords Genetic algorithm · Vaccine optimization · Drug discovery · COVID-19

V. Ajantha Devi
AP3 Solutions, Chennai, Tamil Nadu, India

M. Naved
AIBS, Amity University, Noida, India

E. Gangadevi
Loyola College, Chennai, Tamil Nadu, India

A. Nayyar (✉)
Duy Tan University, Da Nang 550000, Vietnam
e-mail: anandnayyar@duytan.edu.vn

Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_1

1 Introduction

As COVID-19 vaccines have started to enter the market, there is high competition to frame its infrastructure, the complete status to distribute to various places, its supply chains, and to know the demand rates. All the countries have taken the pre-planning steps to stock vaccines and distribute them to their people. The pre-planning actions include various steps like determining the roles and responsibilities to plan and coordinate, framing legal and regulatory framework [1], targeting the prioritized people [2] like the most vulnerable person [3] who needs vaccines immediately, taking care of service delivery, educating about the use of vaccines, and monitoring and finally ensuring safety surveillance. Several factors are to be considered for humans to gain the use of vaccines. First of all, we should know about how effective the vaccines will work [4, 5], the duration it takes to get approval, the way it is manufactured, distributed throughout the countries, and lastly how many people utilize it. Even though the vaccines are ready in the market, we must ensure the response of the immune system of the population [3]. The reason is that the reaction of the vaccines will differ from person to person according to their immune level. Finding this kind of differentiation [6] like the human body, their age factor, immunity level, the diseases which they are already affected, etc., becomes a challenging one. The design and infrastructure of the vaccines are also not projected to the public. Hence, it is difficult for the researchers to come up with an accurate solution. There is a big challenge here that with a small sample of vaccines [2], the healthcare centers are in a position to decide which population should be vaccinated first. We also might get situations of vaccine shortages. In this proposed method, we apply a genetic algorithm on optimized vaccination technique, for the population to gain and to reduce the risk.

Genetic algorithm, on the other hand, is applied with vaccine optimization and mathematical models [7] which helps to derive the approximate solutions for near-optimal results to depict the distribution of vaccines. This is achieved by the fitness function to display the results. Initially, this fitness function approximates the solution and gathers the fittest results which are matching to create a new finding that will be good than the existing solution. The main reason for applying a genetic algorithm [8] for vaccine optimization is to search a huge state space to identify the population who needs the vaccines, its multimodal state space to compare the humans with different criteria like age-wise, severity wise, and priority-wise. The result finalized in this approach will be robust. The results found through the genetic algorithm is fast and most effective which focuses on discrete and continuous functions.

We propose that the vaccine distribution system with optimization achieved through a genetic algorithm is much more effective than the normal distribution strategy without having a planning schedule. This method also can generalize the parameters for the total population by optimizing [9] the models. The steps of crossover and mutation of genetic algorithm help to locate each of the individuals from the population and identify the persons to take up the vaccines.

2 Vaccine Optimization: A Literature Review

Mathematical models come as an important factor in bioinformatics, where they help improve vaccine production [10]. This approach focused on the influenza virus and its varying nature as it spreads. This experiment uses viruses that replicate in a single cell nature. The interesting area is the population balance model, which focuses on the distribution of the influenza cells [11] while observing their internal components. These components include proteins in the virus that define how the cells behave. The model also added to the external actions surrounding the cells and allowed the experts to understand their dynamic behavior under various conditions. The results of the study indicate that varying the proteins present in the cells [12] results in significant changes in the behavior of the virus.

The application of nanotechnology in biomedicine is progressing steadily, as illustrated by Doll et al. [13], where experiments target vaccines' delivery through nanoparticles. The nanoparticles developed in this study are icosahedral self-assembling protein ones, which allow the vaccine particles to attach to them for delivery to the required area. The self-assembling part is critical in determining the linker regions each particle possesses. Comparison of the effectiveness of different nanoparticles was possible through further modeling of predictions in computer simulations. These models have allowed the study of the linking regions of each nanoparticle type to be categorized based on their performance. The study shows successful designs of the particle T2i88, which shows a high level of self-assembly properties. The results have paved the way for better optimizations in how the nanoparticles handle vaccine applications.

DNA-based vaccines are a new approach to handling serious illnesses. An example by Sefidi-Heris et al. [14] on a tuberculosis vaccine based on DNA illustrates promising results. This unique look into vaccines allows promising handling of immune responses at the cellular level while still incurring low costs in production. The effectiveness of the vaccines is also highly stable and can apply to multiple diseases at a time. The process leverages nanoparticles in the delivery of the vaccines to the cells. The results indicate an added advantage to the effectiveness of the vaccines but are highly affected by the mode of transmission, selection of antigen, and formulation of the vaccine. However, these results point to a successful vaccine against TB, which will be much quicker in its effectiveness and low possibility of infecting people with the disease.

Minimizing illnesses and possible deaths in an area with limited vaccine amounts requires a unique approach. As shown by Patel et al. [15], genetic algorithms come in handy when studying distributions of vaccines. The study looks into the mathematical nature of populations, with an additional simulation of epidemic situations and application of the random mutation hill climbing algorithm. Modeling a possible pandemic of influenza has allowed the researchers to understand the best distribution tactics in cases where the population access to the vaccine is limited to around 10–90%. The results of these simulations and comparisons are the optimized versions

of vaccine distribution produce many more effective results than choosing the recipients of the vaccines randomly. A further conclusion is that genetic algorithms can be ported to other types of diseases and infections, making them applicable in the proper handling of vaccines.

There is a lot to learn from multi-objective optimization, and evolutionary algorithms as more complicated problems in optimization keep rising. Wong et al. [16] illustrate the claim by coming up with a hybrid algorithm that focuses on these types of optimization problems. The researchers approach the issue with benchmarking functions and compare the results with four other optimization algorithms. The unique algorithm focuses on a selection of antibodies that are selected, edited, and sorted depending on their affinity. Several cycles operate to a stop condition, with a recording of the repeating times, performance, and time. The results show stability in how the algorithm performs on the multi-objective functions, with potential improvement in the handling of sets of issues. The overall costs of handling the optimization problem were thus reduced while still achieving similar results to the rest.

A different study by Florentino et al. [17] provides a different perspective on genetic algorithms to understand infection control. This research case handles dengue fever; a viral caused disease currently under scrutiny by researchers. The approach to modeling the disease using multi-objective genetic algorithms divides up the mosquitoes, which transmit the virus, into different sections such as sterile, male, and female. Further subdivisions in the different stages of the mosquitoes' life cycle allow the calculation of probabilities of their infection rates. This approach seeks to reduce the costs of controlling these mosquito types and, by extension, the virus. The multi-objective approach has allowed variability of the conditions as the state of the system changes, which is the spread of dengue. The results show interesting results where the availability of choices in different conditions is possible, thereby better handling the virus.

Artificial neural networks (ANNs) also come in handy when handling genetic algorithms (GAs), which describe how viruses spread and are produced [18]. Looking at the elements in the observations and how they changed when genetic algorithm processes like selection, mutation, and crossover were applied has provided the study with interesting conclusions. The research, however, combines the experimental method of uniform design with ANN and GA, which seeks to keep the number of tries at the lowest possible levels, especially in complex cases. This approach shows a positive result in more efficiency by combining the uniform design to ANN models, saving time and costs when producing vaccines for viruses. The researchers conclude that the model is applicable in other viruses and vaccine production, provided that the optimal conditions are adhered to.

Applying the knowledge and benefits of genetic algorithms to optimizing multi-minima functions is an area with interesting results. Such a study by Xing et al. [19] shows possible improvement in efficiency benefiting from the selection and mutation processes of genetic algorithms. This paper illustrates the development process of intelligent genetic algorithms for global optimization cases. This approach is new

compared to other genetic algorithm approaches which it compares to. By intelligently adding control points on the different optimization parameters, the researchers were able to get detailed performance data. In-depth testing of the functions with known minima provided good results that could be compared to other approaches. The intelligent GA produces better performance than the rest, with the possibility to further improve the handling of the size of the population and the process of selecting the optimization operators.

Immune genetic algorithm (IGA) and binary trees have yet another viewpoint in the optimization of dynamic systems. As [20] indicates in their study, the binary tree contains nodes with points of knowledge. The processes approach the optimization by studying the actions of a biological immune system and looking at how it handles different situations. The immune genetic algorithm approach has shown an advantage of relying on immune memory and the diverse effects of antibodies when dealing with illnesses. By basing the optimization on the property of time, the improved IGA has better results compared to the typical one. These results leave enough room to apply the method to other situations, such as handling viruses. The researchers also conclude that expanding on the adaptability of the system can also benefit from such an approach for optimization.

Focusing on the facility for manufacturing a vaccine has shown to be important [21]. The study looks for possible ways to optimize the area the vaccine production happens by looking at the stages of the process, which include processing, purifying the virus, split virus purification, and organizing the resulting vaccine into packages. Modeling the process included looking into the complexity provided by the manufacturing process and mapping them into a computer. The bottlenecks present in any instant of activity affect the overall efficiency of the vaccine production. The complex nature of external parameters and vaccine production lead to difficulty in satisfying all issues. However, the result shows a detailed plan, and anticipation of extreme cases can provide enough room to plan for the upcoming cases.

Obtaining essential hypothetical proteins from a Database of Essential Genes (DEG) and eliminating the ones not essential in a virus or vaccine is a unique approach to bioinformatics. This process was taken by Sah et al. [22] to identify therapeutic targets by mining for Salmonella essential proteins. The process targeted five different versions of the Salmonella virus and a collection of protein versions specific to the viral infections. Computer-based simulation of the effects of the different proteins allowed the elimination of the ineffective ones, eventually narrowing them down to ones that have a high potential of leading to vaccines. The approach, however, has limitations in the accuracy values of the prediction models. The results of the approach were useful in that the team identified useful targets to help handle Salmonella.

Minimizing the impact of COVID-19 has become a worldwide priority, with measures to understand its dynamics already in place. Understanding the dynamics of the virus through genetic algorithm steps is a new approach [23]. The researchers modeled the data from several countries to allow probabilistic simulations on possible outcomes. The different models handle the spreading of infections, dynamics of change, and the optimization of the pandemic variables. The process also respects

the real-world data on how the virus is spreading. The results of the study indicate that the approach is successful in predicting the possible spread patterns of the pandemic, and the parameters involved can help control the virus. The optimization of virus spreading nature requires as many parameters as possible, to match the real-world situations.

3 Genetic Algorithm Flow

3.1 The Optimization Issue

The issue of study and upgrade immunization conventions utilizing stochastic enhancement procedures or potentially AI strategies is very new. In [24] shows how to utilise GA to find the best immunisation for a large public organisation, building on the work of [25]. The system introduced in [26] incorporates immunizer highlight and capacity information with AI to give a chance to deal with finding and evaluating multivariate invulnerable connections. We previously depicted in numerical terms the issue to streamline the organization timetable of a malignant growth immunization in [27], and we reassume here the introduced outcomes. The principal thing one needs to look at during the time spent deciphering the organic streamlining issue into numerical terms is spoken to by the idea of immunization adequacy.

The optimal implementation of recombination administrators with regards to a phylogenetic GA is likewise not self-evident. GAs share with reproduced tempering calculations the capacity to escape from neighborhood optima by traveling through valleys of lower wellness in the hunt space. The utilization of recombination administrators to unite bits of numerous arrangements likewise takes into account "hopping" through the hunt space and getting away from nearby optima. Regular GAs look to keep up a lot of variety in the arrangements present in the populace throughout a hunt and hope to lose this variety as the populace, in the long run, approaches the worldwide ideal. This cycle is named intermingling, and loss of variety in the populace before arriving at the worldwide ideal is known as an untimely assembly. The pace of assembly is needy principally on the quality of determination, what is more, the populace size.

In particular, in immune informatics, a typical improvement issue is the hunt for ideal inoculation plans. Looking for an ideal dose and timing plan is prominently significant in disease immune-preventive methodologies since one has the prerequisite to keep up an elevated level of defensive insusceptibility for the whole lifetime of the host. From this thought, it gives the idea that the utilization of a GA to look for ideal immunization systems could be of incredible value. The fundamental issue is to characterize, in numerical terms, the capacity that must be upgraded by the GA.

3.2 Process of Vaccine Selection

A variety of lab tests and methodologies provide correlative data on changes in the antigenic and hereditary characteristics of COVID infections, which can be used to choose the best pneumonia antibody infections. However, inherent limitations in the organic metrics used, as well as significant differences in the outcomes obtained by different research institutions, complicate data collection and interpretation (Fig. 1).

The outline shows that the individual strides in the choice of competitor immunization infections and improvement of normalizing reagents for occasional pneumonia and a potential COVID pandemic are basically proportional.

For occasional antibodies, the timetables are as follows:

- **Data Collection**: the selection, disengagement, and thorough antigenic analysis Furthermore, the hereditary depiction of ongoing infection detaches continues in a consistent manner
- **Genetic and Antigenic Analysis**: Examinations of the acknowledgment of delegate ongoing infections by immunization prompted antibodies in human

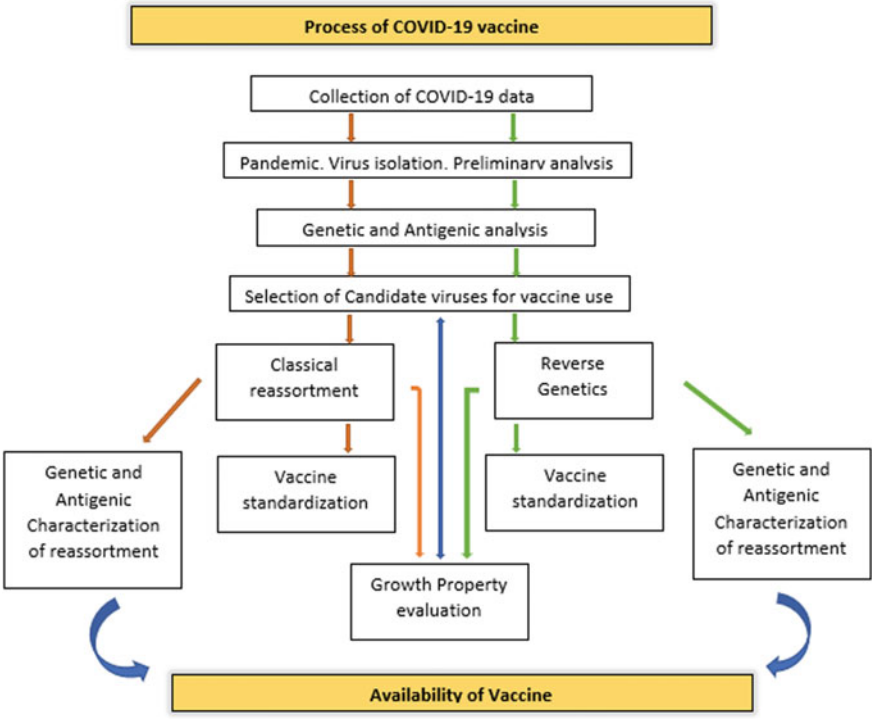


Fig. 1 GA flow to process of COVID-19 vaccine

- **Selection through Classical Reassortments:** Competitor infections for immunization use are looked into and chose, and high-development reassortments are arranged and described after recognizable proof of (potential) antigenic variations.
- **Growth Property Evaluation**: Evaluation of their development properties is led in an opportune way and antibody synthesis.
- **Vaccine Standard Reagent Development:** Preparation of the normalizing reagents for new immunization segments is started once the specific antibody infection has been chosen.

3.3 Serological Study

To support the presentation of seroepidemiological studies to survey the effect of COVID in a populace, nations ought to be upheld in setting up all around described serum banks old enough separated agent sera as a public, local, and worldwide asset.

The accessibility of counter-acting agent guidelines would not just upgrade the similarity of serological information created in various research centers, what is more, nations yet additionally encourage the correlation of immune response reactions to various antibodies.

Expanding consideration regarding pneumonia immunization adequacy studies will prompt the accessibility of all the more continuous information for contrasting clinical advantage and the level of antigenic relatedness of antibody and coursing infections. Such examinations, particularly those dependent on research facility affirmed results, ought to give proof of the particular advantages of inoculation. Predictable examinations giving appraisals of antibody viability over progressive pneumonia seasons ought to improve comprehension of the impacts of little as opposed to major antigenic contrasts among antibody and flowing infections on clinical results and should assist with easing concerns emerging from an apparent immunization confuse brought about by the development of infection clades displaying almost no antigenic float.

3.4 Technology Development

Ongoing advances in high-throughput hereditary sequencing might prompt an incredibly improved comprehension of the hereditary changes happening in pneumonia infections also, the developmental collaborations that happen between co-flowing infections. Top to bottom examinations of the exact components engaged with the advancement and the study of disease transmission of COVID would require progressed bioinformatics apparatuses to exhaustively mine the information.

3.5 Mathematical Model

Various numerical displaying procedures have now been utilized to pick up experiences into the instruments that underlie both the development and the study of disease transmission of COVID infections. For instance, exploratory models have been created to produce and test different theories to clarify the generally confined decent variety of pneumonia infections in the wording of compelled antigenic collection and to investigate the fundamental nature of insusceptibility. They have likewise been utilized to improve comprehension of the degree of between-subtype also, between-type rivalry, and of the likely results of such associations for patterns in the rate of occasional pneumonia infections.

Phylogenetic models have likewise been utilized to distinguish changes in particular imperatives according to antigenic float also, between species transmission. At the point when dependent on the amino corrosive replacements related with mammalian host transformation, such models may help evaluation of the pandemic capability of avian and other creature infections.

Phylodynamic showing, which is based on publicly available arrangement information and reinforced with antigenic information, has recently been successfully utilised to trace the development of new antigenic and hereditary variations, as well as their geographic spread.

Notwithstanding, without incredibly improved comprehension of the hidden transformative and organic components and different cycles included, the limit of current numerical displaying methods to anticipate the timing and nature of future antigenic and hereditary changes is restricted. The naturally stochastic nature of pneumonia development may make such a prescient demonstrating very testing. Where changes happen over brief timeframe scales, the utilization of easier non-unthinking factual calculations, for example, those utilized as the premise of antigenic mapmaking, is probably going to be more valuable in encouraging antibody infection determination than endeavors to create prescient models from the current complex dynamical models of pneumonia development and transmission.

3.6 Impact of New Vaccine Technologies

All new pneumonia antibody innovations have suggestions for antibody infection choice and administrative and producing measures. Be that as it may, any possible necessity to tailor the infection determination cycle to explicit kinds of immunization is probably not going to be an urgent issue, particularly if propels in immunization innovation and speed of creation lead to more noteworthy adaptability in the circumstance of proposals. Even though live-weakened immunizations are not yet all around authorized, the current immunization piece proposal measure is utilized. Nonetheless, neutralizer reaction is certifiably not a decent connect of insurance for such antibodies, and the distinguishing proof of a genuine associate may influence the necessity for yearly refreshing.

4 Vaccine Optimization

To get better results, we need to compare all the body types, and according to that, we have to do tests on them. To optimize the dose and antidotes, we need to know strong humoral immune which is **S377-**588-Fc at 1 μg results in better immune. Most of the countries in the world have controlled the epidemic using social distance, lockdown, and have reduced the effective reproduction numbers.

Hence, due to this, we analyzed the optimal allocation strategies. For optimization routine, we have to distribute the population into five vaccinating groups (1) age-0–19 years, (2) 20–49 years, (3) 50–64 years, (4) 65 above, and (5) those who are already having diseases like asthma, diabetic, allergy, and respiratory-related diseases so that they could get recovered. As we know that the percentage of death rate related to this diseases is more as immunity power is less.

In some humans, we cannot find the symptoms; for such patients, there is an incubation period in which the test is held in three stages: (1) day one test (2) 7th-day test and (3) 14th-day test. During these tests, they are home quarantine. Due to which others are not affected.

5 Genetic Algorithm-Based Vaccine Optimization

A higher-level function for meta-heuristic problems to find, generate, and choose the best optimal solution can be achieved by a genetic algorithm (GA) being nature inspired one. As an evolutionary algorithm, GA is inspired by the steps of natural selection. Even if the infected people do not have any symptoms, they can be more contagious, and so the virus can spread easily to their immediate vicinity. According to health-care data, infected people are manifested to be more infectious before the onset of symptoms and the development of their condition. Patients who develop persistent kind of disease can be infectious for a prolonged period. The infected people enter the dormant and development period on average of 5–6 days during which the symptoms are not shown. Followed by this status, the infectious period on average of 14 days may show the symptoms of COVID (Fig. 2).

In this proposed method, the optimization of vaccination is achieved through the genetic algorithm. GA is the best search and optimization approach based on Darwin's principle to completely focus on the survival of the fittest. The main purpose of this paper is to manifest the genetic algorithm-based vaccination done through optimization. The robustness and the quality of distributing vaccines to the people depend on the choice of the accurate selection process that is done through GA. The more they are vulnerable and endangered, the more priority and attention must be given. Accordingly, the decisions are derived from the categories of people we have and prioritize them for taking vaccines. An optimal vaccination strategy helps us to maintain its efficiency with a minimum number of vaccines. By keeping this in mind, our objective is to follow a quantitative method with the help of optimization and a

Fig. 2 Split the population



genetic algorithm-based approach to help biologist to design and produce vaccine protocols.

The vaccine optimization problem begins with considering the quantity of the vaccine and the rate of the population. In addition to that, it focuses on the illness attack ratio of the pandemic disease. To reduce the influence of the epidemic, the proportion of each age group among the population to be vaccinated must be calculated. For the routine of optimization, we split the population into five age groups.

The five different categories of the population are indexed as p1, p2, p3, p4, and p5, where $i = 1,\ldots,5$, and V be the available number of vaccine doses. V_i be the total number of people who gained vaccines. Under each category, the number of persons is denoted as i, and the total population size is $n = \sum_1^5 n_i$. Let μ_i be the average illness rate in each age group. Then, the total number of vaccine doses supplied will be $T = \sum_1^5 n_i V_i$.

In this situation, $T \leq V$, because the requirement of vaccines will be less than the number of available vaccines. The reason is that, all over the country, biologists are under the process of preparations of vaccines in different structures. In some instances, only one dose of vaccine will not be sufficient. Rather, doctors need to inject different levels of doses as per the physical body condition of each individual. Keeping this in mind, our approach of applying the genetic algorithm focuses mainly on the vulnerable cases which attains the best fit function. Under each age category, the vaccine will be distributed as $\frac{V_i}{n_i}$ for the whole population. So, the optimization strategy is derived as

$$\sum_{1}^{5} \mu_i n_i$$

And

Hence,

$$\sum_{1}^{5} n_i V_i \leq V$$

An optimal sharing of vaccines is carried out for a different number of doses present based on the objective function. Hence, for each possible combination of vaccines V, a desired optimal solution is obtained. The optimization technique also obtains the objective function whether to minimize or maximize the population value, set of variables that target the objective function, and the set of constraints that permits the unknown variables to hold certain values. Based on the objective function value, the complete population is carried to the next generation through reproduction. The crossover operator then randomly selects the pairs of parents from the given population that are mated to obtain new offspring. Then comes mutation which replaces the existing individuals from the population, and this iteration gets repeated until the fitness function is derived.

Algorithm

We have applied the following algorithm to choose the vaccination optimization using the genetic algorithm for the population.

Begin
Get the parent information by obtaining the total population.
Group the population into five different age-wise category.
Identify two legitimate comparison.
Compare with the number of vaccines present and the number of vaccines required.
If $T \leq V$, start with V as a new parent
Else
Start with T as a new parent
End.

The genetic algorithm process is applied for vaccination optimization as follows:

Initialization

The total number of populations at a given area is considered as a parent function. Initialization depends upon the size of the total population. All the possible solutions can be obtained through search space that satisfies $\sum_{1}^{5} n_i V_i \leq \mathrm{V}$.

Encoding

To regulate the performance of the genetic algorithm, encoding helps to translate the given input data to a specific extent. Any kind of dimensional parameters can be translated into binary coding.

Fitness function

Fitness function plays a vital role in evaluating how close a given input to the optimum solution for the current problem. In our problem, it identifies the group of people who immediately requires a vaccine who exactly fits the purpose. $\frac{V_i}{n_i}$ status is verified to obtain the intuitive solution.

Crossover

Crossover considers more than one parent solutions to deliver an offspring from that population. Here, it takes the available number of vaccines and the required number of people to find a new destination. The binary crossover technique can be applied to make a new solution. Two random crossover positions in the parent are created, and the same is applied to the first offspring. The number of iterations repeated in the crossover, and the chromosomes exchange their pairs. In this process, the search for the persons who are going to get vaccinated depends mainly on the creation of the category of persons concerning their age and their physical condition. This process confirms the swapping of genetic properties between parents to create chromosomes which seems to be good than the parents.

Mutation

The process of mutation changes one or more solutions compared with the existing ones to attain the best result. Since the randomly generated individuals are modified, the fitness value also gets updated. Until we yield a better individual than the current one, the process of mutation is iterated.

$$\begin{cases} v_n^i + \dfrac{\min(v - \sum_1^5 v_n^i)}{n_i} & \text{if } \ T \leq V, \end{cases} \tag{1}$$

The above mutation function makes us minimize the function and allows us to know the distance of the parent.

Termination

After completion of all the above steps, the process will be iterated until we conclude with the number of persons under each group who are vaccinated. The optimization process is also restricted by the cost and time of the fitness function.

6 Conclusion

Our study has finalized a bioinspired approach for saving human life from COVID-19 by selecting the human population for vaccination through a genetic algorithm. As this kind of technique solves the local maximum search, we acquire a fine result to cover the most vulnerable persons. The main advantage of using this approach is it locates the complete population by visiting each person through mutation and

crossover steps on an iterative basis. This process completes after a required number of iterations depending upon the change of the individuals at each step. The quality of the result produced through optimization will be dependent on the goodness of the parameters. Here, the mathematical model used represents the rate of transmission of the disease, and it shows the rate of the optimized solution after finding the result.

The proposed algorithm helps to derive the sample individuals in a step-by-step manner from the population. We also concluded that the age factor also arises a lot of effect for the COVID-19 as the immunity level decreases. We have also recognized the optimal allocation functions and the changing characteristic behavior of vaccine by validating the approach through the input models. As a future enhancement, the proposed approach receives a result for the local maximum technique. We can achieve a global optimum problem by applying a meta-heuristic algorithm along with a genetic algorithm. In this method, the genetic algorithm helps us to come up with the best solution to solve such kind of combinatorial optimization problems.

When the next pandemic strain is discovered somewhere on the planet, vaccine development and manufacture should begin immediately. The epidemic simulation model using the existing population structure can be calibrated once the age-specific sickness attack rate patterns have been determined. The optimization model can then be used to determine the best vaccination distribution based on the quantity of vaccines available.

References

1. Verbeken G, Pirnay JP, De Vos D, Jennes S, Zizi M, Lavigne R, Huys I et al (2012) Optimizing the European regulatory framework for sustainable bacteriophage therapy in human medicine. Arch Immunol Ther Exp 60(3):161–172
2. Medlock J, Galvani AP (2009) Optimizing influenza vaccine distribution. Science 325(5948):1705–1708
3. Buckner JH, Chowell G, Springborn MR (2020) Optimal dynamic prioritization of scarce COVID-19 vaccines. medRxiv
4. Standaert B, Van Vlaenderen I, Van Bellinghen LA, Talbird S, Hicks K, Carrico J, Buck PO (2019) Constrained optimization for the selection of influenza vaccines to maximize the population benefit: a demonstration project. Appl Health Econ Health Policy 1–13
5. Pappalardo F, Pennisi M, Castiglione F, Motta S (2010) Vaccine protocols optimization: in silico experiences. Biotechnol Adv 28(1):82–93
6. Pinti M, Appay V, Campisi J, Frasca D, Fülöp T, Sauce D, Cossarizza A et al (2016) Aging of the immune system: focus on inflammation and vaccination. Eur J Immunol 46(10):2286–2301
7. Rodrigues RF, da Silva AR, da Fonseca Vieira V, Xavier CR (2018) Optimization of the choice of individuals to be immunized through the genetic algorithm in the sir model. In: International conference on computational science and its applications. Springer, Cham, pp 62–75
8. Jiao L, Wang L (2000) A novel genetic algorithm based on immunity. IEEE Trans Syst Man Cybern-A 30(5):552–561
9. Enayati S, Özaltın OY (2020) Optimal influenza vaccine distribution with equity. Eur J Oper Res 283(2):714–725
10. Duvigneau S et al (2018) Mathematical modeling as a tool to improve influenza vaccine production processes. IFAC-PapersOnLine 51(19):1–4

11. Carter DM, Darby CA, Lefoley BC, Crevar CJ, Alefantis T, Oomen R, Ross TM et al (2016) Design and characterization of a computationally optimized broadly reactive hemagglutinin vaccine for H1N1 influenza viruses. J Virol 90(9):4720–4734
12. Li ZC, Zhou XB, Lin YR, Zou XY (2008) Prediction of protein structure class by coupling improved genetic algorithm and support vector machine. Amino Acids 35(3):581–590
13. Doll TA et al (2015) Optimizing the design of protein nanoparticles as carriers for vaccine applications. Nanomed Nanotechnol Biol Med 11(7):1705–1713
14. Sefidi-Heris Y, Jahangiri A, Mokhtarzadeh A, Shahbazi MA, Khalili S, Baradaran B, Santos HA et al (2020) Recent progress in the design of DNA vaccines against tuberculosis. Drug Discovery Today
15. Patel R, Longini IM Jr, Halloran ME (2005) Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. J Theor Biol 234(2):201–212
16. Wong EY, Yeung HS, Lau HY (2009) Immunity-based hybrid evolutionary algorithm for multi-objective optimization in global container repositioning. Eng Appl Artif Intell 22(6):842–854
17. Florentino HO, Cantane DR, Santos FL, Bannwart BF (2014) Multiobjective genetic algorithm applied to dengue control. Math Biosci 258:77–84
18. Takahashi MB, Rocha JC, Núñez EGF (2016) Optimization of artificial neural network by genetic algorithm for describing viral production from uniform design data. Process Biochem 51(3):422–430
19. Xing LN, Chen YW, Cai HP (2006) An intelligent genetic algorithm designed for global optimization of multi-minima functions. Appl Math Comput 178(2):355–371
20. Xue C, Dong L, Liu J (2012) Enterprise information system structure optimization based on time property with improved immune genetic algorithm and binary tree. Comput Math Appl 63(7):1155–1168
21. Tsang KH, Samsatli NJ, Shah N (2006) Modelling and planning optimization of a complex flu vaccine facility. Food Bioprod Process 84(2):123–134
22. Sah PP, Bhattacharya S, Banerjee A, Ray S (2020) Identification of novel therapeutic target and epitopes through proteome mining from essential hypothetical proteins in Salmonella strains: an In silicon approach towards antivirulence therapy and vaccine development. Infect Genet Evol 104315
23. Ghosh S, Bhattacharya S (2020) A data-driven understanding of COVID-19 dynamics using sequential genetic algorithm based probabilistic cellular automata. Appl Soft Comput 96:106692
24. Sanders L Optimal vaccination of a general population network via genetic algorithms. BioRxiv preprint. https://doi.org/10.1101/227116
25. Patel R, Longini IM, Halloran ME (2005) Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. J Theor Biol 234(2):201–212
26. Choi I, Chung AW, Suscovich TJ, Rerks-Ngarm S, Pitisuttithum P, Nitayaphan S, Kaewkungwal J, O'Connell RJ, Francis D, Robb ML, Michael NL, Kim JH, Alter G, Ackerman ME, Bailey-Kellogg C (2015) Machine learning methods enable predictive modeling of antibody feature: function relationships in RV144 vaccines. PLoS Comput Biol 11(4):e1004185
27. Pennisi M, Pappalardo F, Zhang P, Motta S (2009) Searching of optimal vaccination schedules. IEEE Eng Med Biol Mag 28(4):67–72. https://doi.org/10.1109/MEMB.2009.932919

Ethereum MongoDB: Integrating Blockchain with Non-relational Databases



Sneha Sarkar, Ramesh Dharavath, and Hemika Jadav Badavath

Abstract Blockchain is a decentralized distributed digital ledger that provides tamper-resistance but is very slow in search query processing with a considerable amount of transaction cost associated with it. With the advent of newer technologies and large-scale data collections, it is extremely important to increase the security level of databases. Thus, enabling the features of non-relational databases that support immutable transactions. In this paper, we propose an open-source system, Ethereum MongoDB, which is developed by integrating Ethereum blockchain features with the MongoDB database. The non-relational database uses a storage model optimized for specific requirements of the type of data being stored. It provides well-designed data formatting and cost-effective, quick query processing but suffers from data reliability. The proposed Ethereum MongoDB, on the other hand, is a blockchain-enabled database that stores data in a non-relational database with the transaction log being stored in blockchain thus guaranteeing the consistency, immutability, and auditability of transaction log along with the faster query processing of databases in a cost-effective manner. In a blockchain, every transaction has some cost associated with it so, mapping the transactions of the database onto Ethereum's smart contract is not a straightforward process. By modularizing the code in certain ways, we can bring down the transaction cost considerably. A comprehensive real-time application has been developed to demonstrate the effectiveness of the system.

Keywords Blockchain · Non-relational databases · Smart contracts · MongoDB · Ethereum

S. Sarkar · R. Dharavath (✉)
Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad 826004, Jharkhand, India
e-mail: drramesh@iitism.ac.in

H. Jadav Badavath
Department of Production Engineering, National Institute of Technology, Agartala 799046, Tripura, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_2

1 Introduction

1.1 Overview of Blockchain

Blockchain is a tamper-resistant distributed shared digital ledger without any central authority. It allows users to record transactions immutably, i.e., no transaction of the blockchain can be changed once published. It contains transactions that are signed cryptographically and grouped into blocks [1]. Whenever a new block is added to the blockchain, it is linked to the previous block through a consensus model, thus creating tamper resistance [4, 5]. Blockchain has several features that make it suitable for decentralized applications.

- **Ledger**: Blockchain uses an append-only ledger to store transactions. This makes the transactions in a blockchain immutable, unlike that of traditional databases.
- **Security**: Blockchains are cryptographically secure, ensuring that the data contained within the digital ledger is tamper resistant.
- **Shared**: Blockchain is shared amongst multiple participants providing transparency across the users in the blockchain network.
- **Distributed**: Blockchain is distributed without any central authority. Scaling the number of nodes of a blockchain network reduces the impact of a bad actor on the consensus model used by the blockchain.
- **Slower query processing**: Blockchain processes queries much slower than relational and non-relational databases primarily due to the linked data storage and the absence of a well-defined data indexing structure for various queries.
- **Higher cost**: Every transaction in blockchain has a cost associated with it thus, making the blockchain operations more costly as compared to traditional database operations.
- **Smart contract**: A smart contract is a collection of code and data that is deployed using cryptographically signed transactions on the blockchain network, e.g., Ethereum's smart contracts.

1.1.1 Non-Relational Databases

Like traditional databases, non-relational databases do not use the tabular schema of rows and columns. Instead, non-relational databases use a schema-less document-oriented model optimized for the specific user requirement. For instance, the stored data could be in JSON format or key/value pairs. Non-relational databases provide several advantages over traditional databases for large-scale storage.

- Non-Relational databases can store large amounts of unstructured data.
- They provide scalability and flexibility to meet changing user requirements.
- Associated transactional cost is almost negligible as compared to blockchain transactional cost.
- They are document-oriented.

– They provide a schema-less data model with the ability to easily store and combine data of any structure without the need to modify a schema.

Thus, the creation and maintenance of a non-relational database are much faster and cheaper. However, the major limitation of non-relational databases is that their data is prone to tampering by a fraudulent database administrator or by a malicious user. Therefore, a blockchain incorporated database system is essential in which the immutability of blockchain to data modification and the faster-querying speed of non-relational databases are simultaneously achieved. There is no straightforward approach to combine blockchain and non-relational databases as they are two different types of data structures designed to meet contrasting business requirements. Therefore, we propose integrating non-relational databases with blockchain in Ethereum MongoDB. The advantage of decentralized and trustless data storage in the blockchain has been incorporated into that fast query processing of non-relational databases.

1.2 Ethereum MongoDB

Ethereum MongoDB features the integration of a schema-less document-oriented database system with an API such as MongoDB with Next.JS into an Ethereum blockchain. Due to the authorization requirement of the Ethereum wallet, we have a safe, reliable mechanism to access the blockchain. Ethereum MongoDB allows users to insert, modify, delete and view data in a way much similar to that of other databases. Figure 1 describes how data is being accessed in Ethereum MongoDB. For a write operation, the transaction log is stored in the blockchain, whereas the actual data is being stored in the database. For a read operation, the data is being retrieved from MongoDB. Ethereum MongoDB combines the key benefits of blockchain and non-relational databases, as Table 1 summarizes. As the history log of Ethereum MongoDB is immutable, any modifications on the actual data are preserved, thereby incorporating security and audibility features of the blockchain into the existing database systems.

The programming language used by Ethereum to write smart contract is Solidity. In Solidity, the functions are compiled into byte code and executed on the Ethereum Virtual Machine (EVM). A smart contract can perform calculations, store information, and automatically transfer funds to other accounts. However, it should be noted that for every blockchain transaction there is some gas amount associated with it. Hence, in order to develop a cost-efficient system, the smart contract should be written smartly.

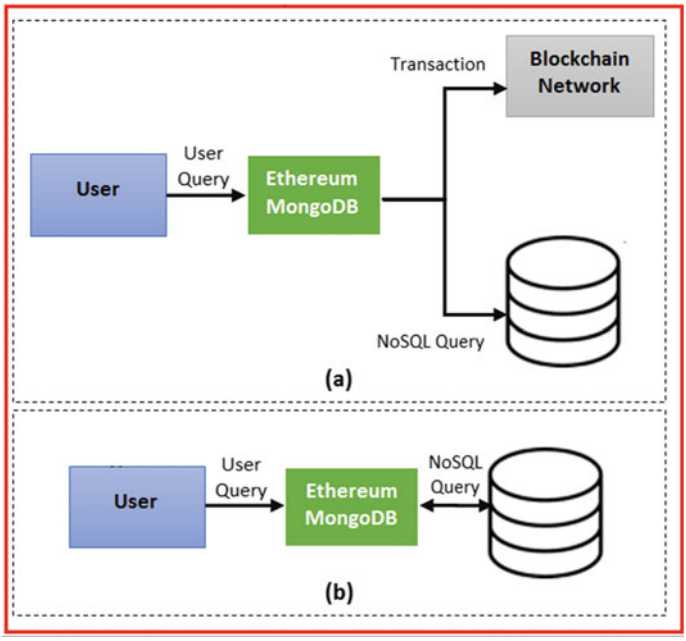


Fig. 1 Ethereum MongoDB access mechanism for (**a**) write operations and (**b**) read operations

Table 1 Comparison between Ethereum MongoDB, blockchain, and non-relational databases

Features	Blockchain database	Non-relational database	Ethereum MongoDB
Immutability	Yes	No	Yes
No central authority	Yes	No	Yes
Faster query processing	No	Yes	Yes
High throughput	No	Yes	Yes
Low latency	No	Yes	Yes

2 Related Work

Traditionally, blockchain such as Bitcoin served as cryptocurrency and was specifically used in the financial domain to store digital assets ownership [17]. However, with the advent of newer blockchains [7], such as Ethereum [16], decentralized applications can be built on blockchain, allowing more generalized applications to be built as per user requirements [2]. For example, Safecast is a global organization that provides access to open sensor data for monitoring radiation levels [23]. It uses blockchain schema for its untampered data. Some blockchain systems, such as Bitcoin [17] and Ethereum [16], follow order-execute architecture. In this architecture,

the transactions are first added to the digital ledger block through a consensus model, and then, they are sequentially executed on all their peer nodes. Although there is a wide variety of blockchain applications in financial transactions and Apps, we focused our attention on blockchain characteristics that can be used as an infrastructure for database systems. Some work has previously been done in this field, such as Ethereum Solidity [8], BigChainDB [9] and EtherQL [23]. BigchainDB [9] uses the Tendermint consensus model [13, 14] for immutability, and the decentralization of data [16], it uses several independent MongoDB [6, 15] instances. EtherQL [23] provides access to the query layer added above the blockchain. EthernityDB [2] provides insights on the blockchain to be used as a database system.

In recent times, data trustworthiness is of huge importance. Ethereum MongoDB focuses on performing reliable transactions in an unreliable environment, where even the user with access to the underlying database system can be malicious. Some work has been done in this regard, keeping in view the relational database system [3]. Scenarios on the detection of tampering of data by malicious users through network-based and data inspection models have been discussed previously [10–12]. Several researchers have also worked on database-level forensic inspection [18–22, 24, 25]. However, we modelled the Ethereum MongoDB database on the non-relational database MongoDB, with the history log being maintained in the Ethereum blockchain. Next, JS providing a UI layer on top of the underlying MongoDB database.

3 Proposed Approach

The proposed approach for Ethereum MongoDB facilitates a database system with the transactional history being logged in the blockchain. It enables to store of data in a non-relational database with the transaction log being stored in blockchain. It guarantees the consistency, immutability, and auditability of transaction log along with the faster query processing of non-relational databases in a cost-effective manner. Ethereum MongoDB provides a user-friendly interface to create, edit, view, and delete records on the proposed database system in a way much similar to that of other database systems. Considering that there is some gas amount associated with every CPU operation of a smart contract in blockchain, we have to design our system so that we store a minimal yet sufficient amount of data in the transaction log that can be used to trace back user activity.

3.1 Design

One of the important factors to consider for choosing the type of blockchain network to be used depends on the proposed system's intended application. Our application requires the deployment of smart contracts and decentralized applications to be built and run without any downtime, fraud, control, or interference from a third party.

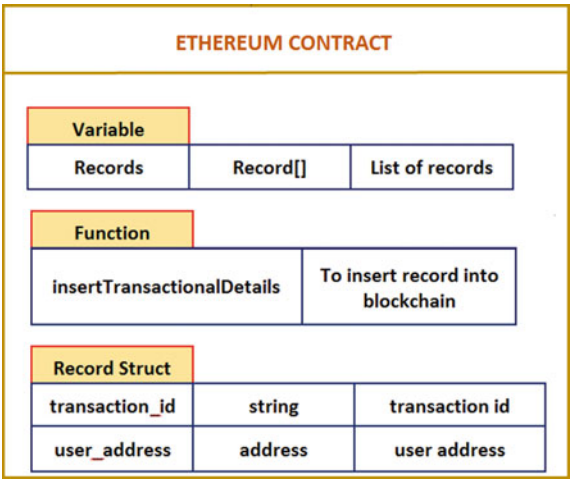


Fig. 2 Ethereum smart contract architecture

Hence, Ethereum is a suitable choice due to its ability to deploy smart contracts and decentralized applications with ease. The smart contract is one of the most expensive modules in a blockchain system. For this reason, as shown by the architecture of the Ethereum smart contract in Fig. 2, we arrange the smart contract as minimal as possible and designed our system so that the Ethereum smart contract is deployed only once. Incase the Ethereum smart contract is updated, the administrator simply needs to reference the web driver to the new smart contract instance. The web driver is responsible for inserting records into the transaction log.

Another important design consideration is the choice of the database system to be used. We want to store large amounts of data with little structure to make it suitable for many changing business requirements. Therefore, MongoDB is an appropriate choice as per our requirements. It provides scalability and flexibility to meet the changing requirements. The schema-less document-oriented data storage of MongoDB provides a flexible data model to easily store and combine data of any structure without the need to modify the schema. To perform the CRUD operations of the proposed database system, we need a user-friendly interface. For this, we use a React-based framework Next.js. Next.js being a pre-rendered React app on the client-side allows users to view and interact with data in the front-end. At the same time, it also does server-side rendering and API routes which can perform server-side code and access data in the database and can be considered as back-end.

3.2 Data Representation and Storage

As per the overall architecture of Ethereum MongoDB, as shown in Fig. 3, for the database storage, we have chosen a schema-less document-oriented database, i.e.,

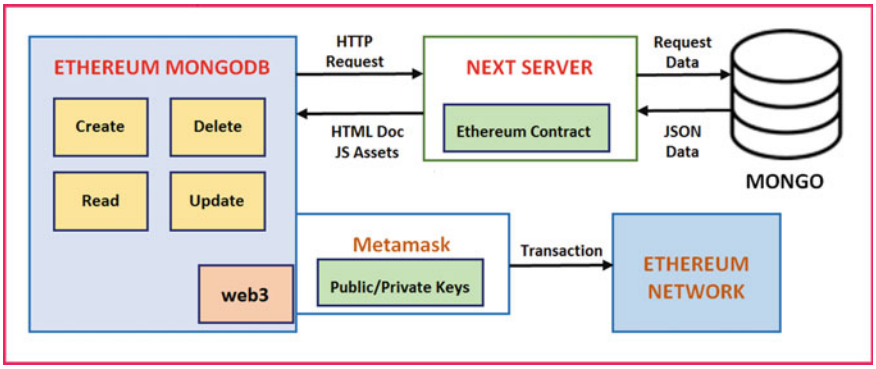


Fig. 3 Architectural overview of Ethereum MongoDB

Table 2 System dependencies

Name	Description
Metamask	Chrome extension to access Ethereum enabled distributed applications
next	v4.1.4 or above
React	v16.8 or above required for react hooks
react-dom	v16.8 or above
Web3	Stable version 1.0.0-beta.26 or v1.0.0-beta.35
Truffle/hdwallet-provider	To connect to the Web3 provider as the current provider
Ganache-cli	To connect to a local blockchain for testing decentralized applications
Solc	v0.4.17, to use solidity programming language.
Mocha	Optional for testing smart contracts
Semantic-ui-react	React components to use the themes from Semantic UI
Semantic-ui-css	To use the CSS stylesheets from Semantic UI
Mongoose	To connect to MongoDB.
Fs-extra	To add files systems methods that are not present in the native module
Isomorphic-unfetch	witches between unfetch and node-fetch for client and server

MongoDB. In MongoDB, data is stored as documents in JSON (JavaScript Object Notation) format. This allows related data to be stored within the document instead of an external table. This makes our application suitable for a wide variety of business requirements. For storing transaction logs we have used the Ethereum blockchain. For every creation, updation, or deletion of a record, a transaction is being recorded in the Ethereum blockchain. After successful completion of the blockchain transaction, a receipt is obtained containing a 256-bit hash of the transaction record. Since records are permanently stored in blockchain this makes the transaction log of our proposed Ethereum MongoDB auditable and immutable. The proposed system uses the dependencies as presented in Table 2.

3.3 Pseudo-Algorithm

Ethereum MongoDB works in a multi-phase process. The pseudo-code for the working of Ethereum MongoDB is given in Algorithm 1. The first phase involves the creation and deployment of smart contracts to the blockchain network. During the second phase, for any insertion, updation, or deletion of data, a transaction log is entered into the blockchain as shown in Algorithm 2, thus making the database log immutable. The third phase involves the storing or retrieval of actual data from the non-relational database. The insert, update, view, and delete operation for Ethereum MongoDB is described by Algorithm 3, Algorithm 4, Algorithm 5, and Algorithm 6, respectively.

Algorithm 1 An outline of the working of Ethereum MongoDB

```
Function ETHEREUMMONGODB (T: user transactional input)
Ti.operation : Operation type (create, update, view, delete)
for all transaction t in T do
  if t.operation = "create" then
    repeat
      title = user input for title of record
      desc = user input for description of record
      tid = CREATERECORD (title, desc)
      if tid != -1 then
        msg = CREATEBLOCKCHAINTRANSACTION (tid, title, desc)
      end if
    until tid != -1 and msg != error
  else if t.operation = "update" then
    display t.transactionID
    repeat
      title = user input for title of record
      desc = user input for description of record
      tid = UPDATERECORD (t.transactionID, title, desc)
      if tid != -1 then
        msg = CREATEBLOCKCHAINTRANSACTION (tid, title, desc)
      end if
    until tid != -1 and msg != error
  else
    display t.transactionID
    r = VIEWRECORD (t.transactionID)
    display r.title, r.description
    if t.operation = "delete" then
      DELETERECORD (t.transactionID)
    end if
  end if
end for
```

Algorithm 2 Logging of transactional data to the blockchain

```
Function CREATEBLOCKCHAINTRANSACTION (tid: transactionID, desc: description, addr:
address)
insert record (tid, desc, addr) into blockchain
repeat
   blockchain mining
until transaction is complete
if transaction is successful then
   msg = transaction receipt
else
   msg = error
end if
```

Algorithm 3 Insertion of data to non-relational database

```
Function CREATERECORD (title: title, desc: description)
generate tid: transactionID
if VALIDATE(title) and VALIDATE(desc) then
   insert record (tid, title, desc) into mongoDb
   t = tid
else
   t = -1
end if
return t
```

Algorithm 4 Updation of data in a non-relational database

```
Function UPDATERECORD (tid: transactionID, title: title, desc: description)
if VALIDATE(title) and VALIDATE(desc) then
   r = fetch record from mongoDb where transactionID = tid
   update record set r.title = title, r.description = desc where transactionID = tid
   t = tid
else
   t = -1
end if
return t
```

Algorithm 5 Viewing data from non-relational database

```
Function VIEWRECORD (tid: transactionID)
r = fetch record from mongoDb where transactionID = tid
return r
```

Algorithm 6 Deletion of data from non-relational database

```
DELETERECORD (tid: transactionID)
delete record from mongoDb where transactionID = tid
msg = successful
return msg
```

3.4 Result Analysis

To elaborate on Ethereum MongoDB's effectiveness, we run the Ethereum MongoDB application on a local server with the blockchain being deployed on the rink by test network and actual data being stored on MongoDB atlas. The snapshot of insert and update operation is shown in Figs. 4 and 5, respectively. It is to be noted that upon the record creation request given by the user, Ethereum MongoDB assigns a unique transaction Id to each record that helps in auditing the operations performed on that record.

For every insert, update, or delete operation, a transaction log is being maintained in the blockchain. Since every transaction has some associated cost, the model displays the required information asking for the confirmation of blockchain transaction and the associated gas amount required to perform that operation. Upon the completion of storing the transactional log on the blockchain, a receipt is received by the user, as given in Fig. 6. In Ethereum MongoDB, any tampering of actual data results in an immutable log being made in the blockchain thus, enabling auditibility as an inherent feature.

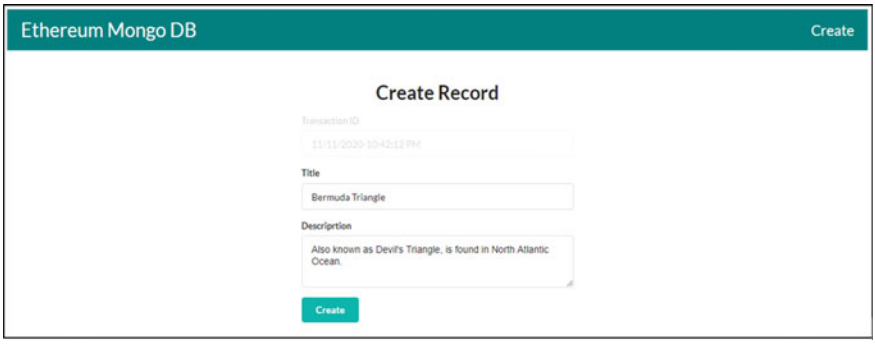


Fig. 4 Creation of record for Ethereum MongoDB

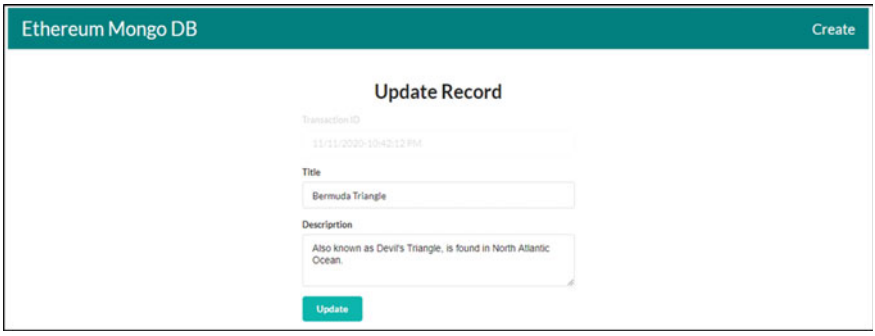


Fig. 5 Updation of record for Ethereum MongoDB

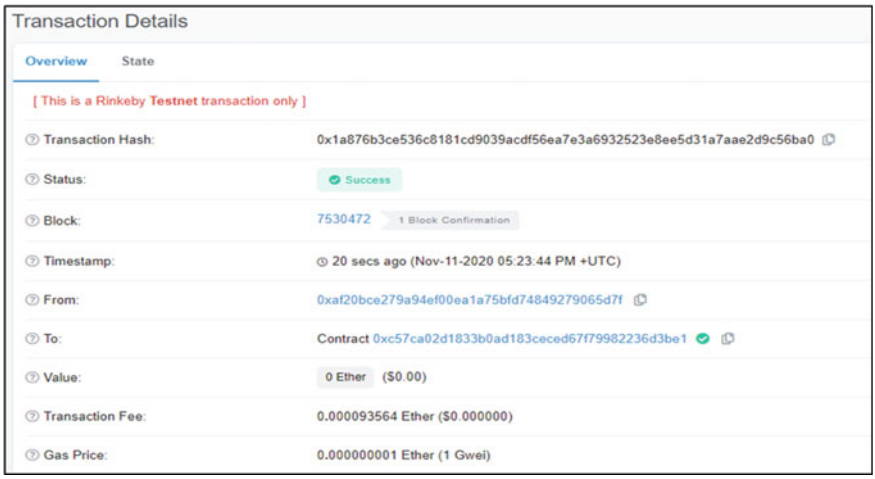


Fig. 6 Transaction receipt on successful completion of blockchain transaction

4 Conclusion and Future Work

With the ever-increasing amount of data, the security of the database is becoming a serious concern. As we have seen, data in relational and non-relational databases can be tampered with by malicious users or database administrators. Blockchain is in a very nascent stage, so as per the current scenario, although blockchain is immutable yet we cannot use it as a blockchain database mainly because of the large amount of time taken for query processing and the cost associated with every blockchain transaction. So, a system needs to be developed that has the advantages of both the blockchain and non-relational database and is cost-effective at the same time.

In this paper, we presented the first system of its kind, Ethereum MongoDB, which illustrated blockchain integration into the non-relational database. It supports a tamper-resistant transaction log along with the fast query processing of a non-relational database. Ethereum MongoDB, being built on a schema-less non-relational database, is highly scalable and provides the flexibility to be used in various applications. Therefore, it offers promising application scenarios for the future. In future work, we envision that the immutable transaction log stored on the blockchain network can be used to support automatic repair on tampered database instances. Also, this data model can be used as the basis for formulating other blockchains integrated relational or non-relational databases as per our requirements.

Acknowledgements This research work is supported by the Indian Institute of Technology (ISM), Dhanbad, Govt. of India. The authors wish to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India, for providing their research support.

References

1. Muzammal M, Qu Q, Nasrulin B (2019) Renovating blockchain with distributed databases: an open source system. Future Gener Comput Syst 90:105–117
2. Helmer S et al (2018) Ethernitydb-integrating database functionality into a blockchain. In: European conference on advances in databases and information systems. Springer, Cham
3. Beirami Amin, Zhu Ying, Ken Pu (2019) Trusted relational databases with blockchain: design and optimization. Procedia Comput Sci 155:137–144
4. Yaga D et al (2019) Blockchain technology overview. arXiv preprint arXiv:1906.11078
5. Tech R, Neumann K, Michel W (2016) Blockchain technology and open source sensor networks. In: Interdisziplinäre Konferenz zur Zukunft der Wertschöpfung. Hamburg, pp 125–134
6. Chodorow K (2013) MongoDB: the definitive guide: powerful and scalable data storage. O'Reilly Media, Inc.
7. CoinDesk: Ethereum's big switch: the new roadmap to proof-of-stake (2017). https://www.coindesk.com/ethereums-big-switch-the-new-roadmap-to-proofof-stake/
8. Dannen Chris (2017) Introducing Ethereum and solidity, vol 318. Apress, Berkeley
9. McConaghy T et al (2016) Bigchaindb: a scalable blockchain database. white paper, BigChainDB
10. Crosby SA, Wallach DS (2009) Efficient data structures for tamper-evident logging. USENIX security symposium
11. Wagner J et al (2017) Carving database storage to detect and trace security breaches. Digital Invest 22:S127–S136
12. Wagner J et al (2018) Detecting database file tampering through page carving. In: 21st international conference on extending database technology
13. Kwon J (2014) Tendermint: consensus without mining. Draft v. 0.6, fall 1.11
14. Tendermint. https://tendermint.com/
15. Mongodb. https://www.mongodb.com/
16. Wood Gavin (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151(2014):1–32
17. Nakamoto S (2019) Bitcoin: a peer-to-peer electronic cash system. Manubot
18. Fabbri D, Ramamurthy R, Kaushik R (2013) Select triggers for data auditing. In: 2013 IEEE 29th international conference on data engineering (ICDE). IEEE
19. Hauger WK, Olivier MS (2014) The role of triggers in database forensics. IN: 2014 information security for South Africa. IEEE
20. Dharavath R et al (2014) Accuracy of atomic transaction scenario for heterogeneous distributed column-oriented databases. In: Intelligent computing, networking, and informatics. Springer, New Delhi, pp 491–501
21. Ramesh D, Sinha A, Singh S (2016) Data modelling for discrete time series data using Cassandra and MongoDB. In: 2016 3rd international conference on recent advances in information technology (RAIT). IEEE
22. Ramesh, D, Khosla E, Bhukya SN (2016) Inclusion of e-commerce workflow with NoSQL DBMS: MongoDB document store. In: 2016 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE
23. Li Y et al (2017) EtherQL: a query layer for blockchain system. In: International conference on database systems for advanced applications. Springer, Cham
24. Ramesh D, Kumar A (2018) Query driven implementation of Twitter base using Cassandra. In: 2018 international conference on current trends towards converging technologies (ICCTCT). IEEE
25. Ramesh D, Kumar A, Kumar A (2019) Similarity-aware EQUALS and IN operator in Cassandra and its application in agriculture. In: 2019 IEEE international symposium on smart electronic systems (iSES)(Formerly iNiS). IEEE

Attention-Based Discrimination of Mycoplasma Pneumonia



Tathagat Banerjee, S. Karthikeyan, Amandeep Sharma, K. Charvi, and Sruti Raman

Abstract Mycoplasma is a rather affinitive type of pneumonia whose only known common symptom is dry cough. In recent dimension of events, it seems to be a coinfection closely related to coronavirus. In the radiography domain, its detection has now not only become more significantly important than before, but due to its passive presence and difficult symptoms, it is classified as hard to detect further because of similarity with coronavirus and typical pneumonia, and its symptoms are often misunderstood. Understanding these trade-offs, the paper aims to improvise deep learning methodologies to detect the presence of pneumonia and then classify further into its subcategories. The analysis involved takes into consideration the use of different transfer learning methodologies.

1 Introduction

Mycoplasma pneumonia is also known as atypical pneumonia. Its nature is known to be highly contagious respiratory infection. General causes take into account bacterium mycoplasma pneumonia. It spreads through the shallow respiratory droplets that serve as a linings home for bacteria to breed. Once the entry operation for these droplets is successful, it can cause tracheobronchitis, pneumonia. As highlighted above, its symptoms are nothing more than that of common cold like dry cough, fever, shortness of breath and few others.

A similar genesis of viral pneumonia is caused by viruses like influenza rhinovirus and coronavirus. Thus, at a basic distinction for biologist and lab experimenter, it is the formational change of bacteria and virus. However, it is worth noting that to detect a single sample of these three subclasses and one normal class it takes more than

T. Banerjee · A. Sharma · K. Charvi (✉) · S. Raman
Computer Science and Engineering, VIT-AP, Amaravati, India
e-mail: charvi.19bce7002@vitap.ac.in

S. Karthikeyan
School of Computer Science and Engineering, VIT-AP, Amaravati, India
e-mail: karthikeyan.s@vitap.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_3

4 min. Hence, the idea of automation using a deep learning model which can deduce the same results in less than one hundredth of a second is not only revolutionary but also shall test the understanding of our deep learning techniques for the same.

The paper aims to first gather relevant and trustworthy data sources. It also tries to create an unsupervised learning methodology to foment new data samples under the same tag label, finally enhancing use of basic convolutional neural network architectures to use Resnet and VGG16 to find correct and dependable prediction. The final section of the paper introduces the novel use of transformer architecture to enhance the results forward and hypothesize on the utility of attention mechanism in understanding the salient features of mycoplasma or atypical pneumonia.

Figure 1 illustrates the different samples of normal and pneumonia lung radiography images. Figs. 2, 3 and 4 illustrate the stipulating difficulty in different samples at the simple CT scan scale.

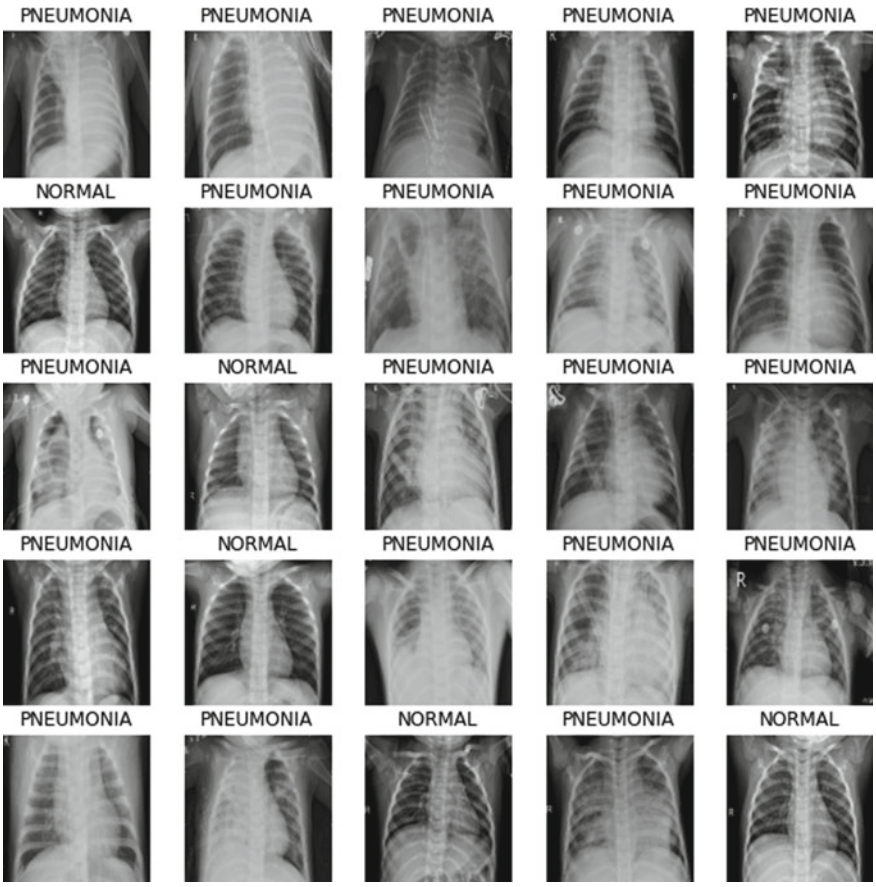


Fig. 1 Normal and pneumonia class

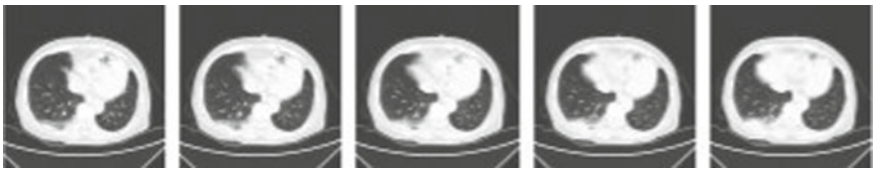


Fig. 2 Mycoplasma pneumonia

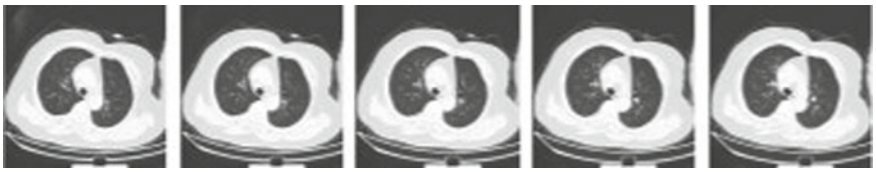


Fig. 3 Typical viral pneumonia

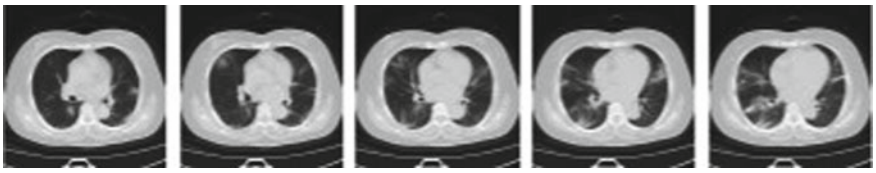


Fig. 4 Coronavirus pneumonia

Thus, the research aims to draw distinctive higher-dimensional discrimination between seemingly similar images of mycoplasma, typical and coronavirus pneumonia. The idea also revolves around the fine feature extraction genesis of neural networks and deep learning algorithms.

CT scan images of different subclasses:

In the upcoming sections, we go through related works in Sect. 2 and methodology in Sect. 3. Finally, we compare the results of the model in Sect. 4 and conclude the approach in Sect. 5.

2 Related Works

Wu et al. [1] developed a multi-class generic pneumonia classification model which can distinguish between the CT scan images of coronavirus, pneumonia and normal images. In the further advancement of Zhang et al. [1] they created an artificial intelligence solution to diagnose the same. However, the class labels were for coronavirus and pneumonia distinction. Researchers have utilized deep learning methods

for detection of bacterial presence information at different parts of the human body like chest et al. [3, 4], eye et al. [5–9], brain [10, 11] and skin [12, 13]. In most of the utilities of biomedical imaging, the usage of such high-end devolution models is not only heavily used but trusted throughout the verses.

Yan et al. [14] discussed the multi-scale convolutional network. It identified a normal sample to various other categories of pneumonia. Asana Oui et al. [14] used a variety of deep convolutional analysis to describe classification on X-ray and CT scan images. Transfer learning-based automated detection of pneumonia by Rahman et al. [15] on four pre-trained deep learning methods. Butt et al. [16] detected coronavirus pneumonia samples utilizing deep learning networks and then drawing correlation between detection rates of RT-PCR and the estimated model ideas. A similar diagnosis by Refs. [17, 18] was also estimated along the same lines to detect COVID-19, bacterial pneumonia and healthy cases.

Besides the hectic ideas and similar demerits to the models, we have come out heavy wishes. The usage of deep learning methodologies signifying the hegemony of network and neural architectures thus also motivates to further classify and analyze over the leftover semantics of mycoplasma or atypical pneumonia.

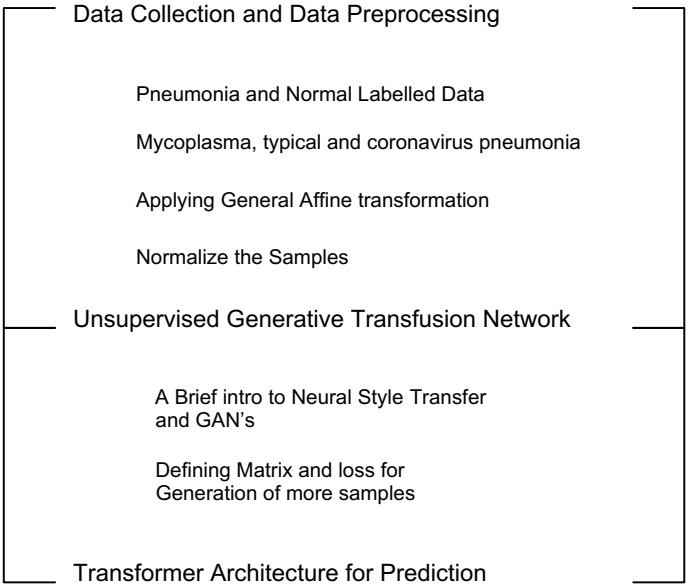
3 Proposed Model

In this section, the paper tries to showcase the prior mentioned ideas and produce appropriate illustration to suffice it. Before we dive into dynamic segregation of the task and its working, it is only important to illustrate a workflow module by module. Flowchart 1 illustrates the working diagram we shall be referencing to in future.

The below-mentioned submodules shall be discussed in the upcoming headings, followed by exhaustive result analytics in Sect. 4.

3.1 Data Collection and Preprocessing

The paper utilizes real-time images with the help of available public repositories of available pneumonia samples to normal image and further collaborates with local lab samples to get real-time images for mycoplasma, typical and coronavirus pneumonia images. A general quality of the image is $500 \times 500 \times 3$. However, for the purpose of training, we have strictly used $256 \times 256 \times 3$. Thus, improving the image quality could serve a potential improvement over our suggested algorithm. Beside the collection of quality data, preceding for a deep learning classification task, it is essentially important to feed in a good quantity as to generalize the concept of high-dimensional feature extraction. Furthermore, it is important to understand the possibility of slight rotational as well as shear affine changes due to human interventions while gathering samples. Thus, in order to facilitate both the quantity and affine adulterations to the samples, we apply data augmentation techniques for rotation and



Flowchart 1 Workflow/algorithmic structure

other required transforms. Tables 1, 2 and 3 describe the data samples and related information on affine changes.

Above tables along with Figure 1 provide quite a detailed idea about the dataset over which the predictions and model hypothesis shall stand. Also, it is only worth nothing that all the samples are normalized as elaborated in Table 3.

Table 1 Pneumonia versus normal class distribution

Class	Samples
Pneumonia	1895
Normal	1055

Table 2 Pneumonia subclass distribution

Class	Samples
Mycoplasma pneumonia	564
Viral pneumonia	1598
Coronavirus pneumonia	1687

Table 3 Affine transformation information ($\mathfrak{E}$ = mean and $\mathfrak{F}$ = standard deviation)

Affine transformation category	Ratio on different classes of images (%)
Rotation (−1.5 to +1.5 degrees)	12.3
Translation (vertical)	5
Normalize ($\mathfrak{E} = 0$, $\mathfrak{F} = 0.32$)	100

3.2 *UGTN: Unsupervised Generative Transfusion Network*

According to Johnson et al. [19], in the domain of neural style transfer, the use of image transformation model was revolutionary. After Gatys et al. [20], where the VGG architecture was frozen or rather, in technical terms set on an evaluation mode such that weights shall not be updated and the feature extraction from image net shall be used to optimize the input image. This marked the start of a new domain of explainable AI called neural style transfer. However, within a year, Johnson [19] came up with a neural network architecture inspired from the generative adversarial networks et al. [21] that could create an image such that would have both the style and content features and the loss could be minimized. Figure 5 explains the architecture of Gatys [20]. Figure 6 demonstrates the network of Johnson [19], and Figure 7 illustrates the use of GAN.

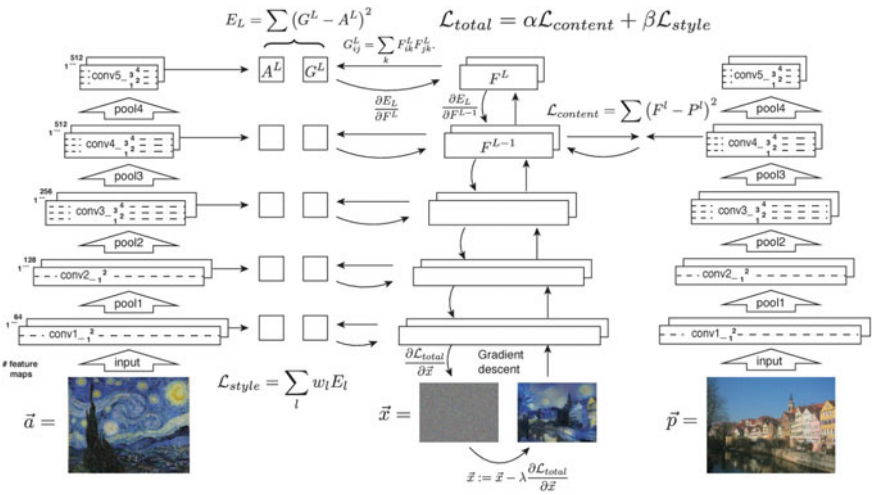


Fig. 5 Gatys et al. [20] working flowchart Johnsons et al. [19]

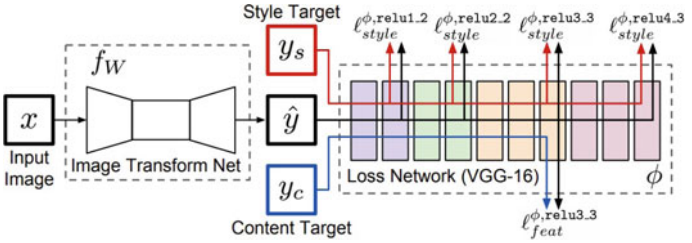


Fig. 6 Johnsons et al. [19]

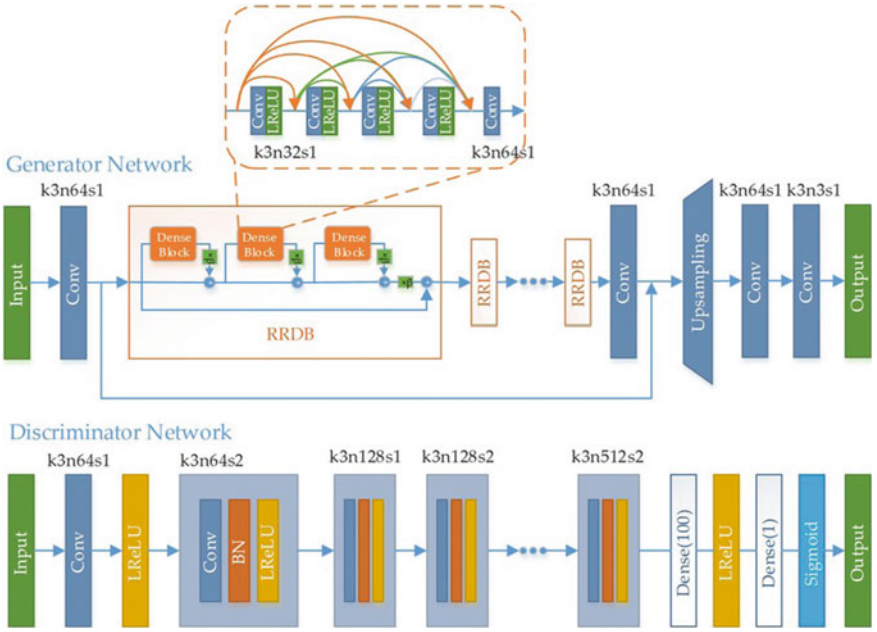


Fig. 7 Generative adversarial network [21]

NST and GANs:

After looking into these beautiful and vivid ideas in the context of medical image processing, we understood that with an appropriate style and enough computational power we can map a completely artificial image generation along with desired stylized form.

The stylization in our case is rather different than in Gatys [20] or Johnson [8], and similarly, generation is also vividly unique than GANs [21]. This is because here neither we are just trying to mimic the content of the extrapolated higher-dimensional feature map in image space like GAN [21–24] nor are we trying to reduce the content requirements and stylize it like neural style transfer [19–21]. The main idea is to

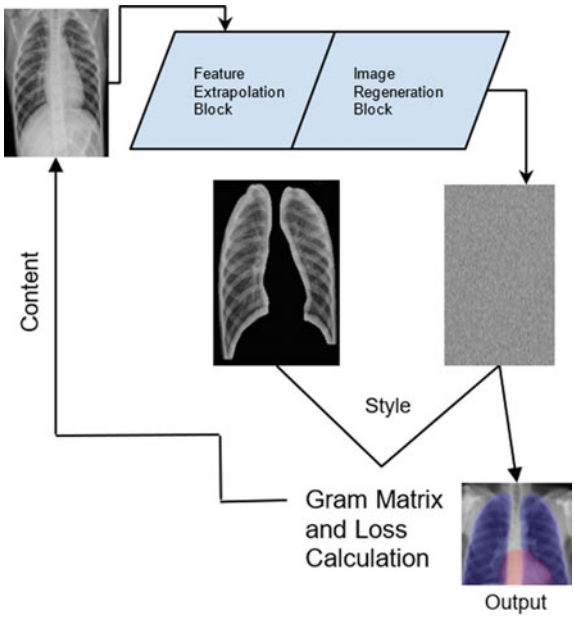


Fig. 8 Brief model overview

transfusion both style and content which are very similar and are probably like masks of image segmentation such that the output image is not stylized but rather supervised masked. These masks are not expected to fit the exact localization of lungs as simple CT samples could not correctly localize any pneumonia but rather formulate a region of high speculation. Hence, for every image output from UGTN, we shall use the gram matrix between image and its available bio-designed lungs counter plot. Figure 8 elaborates on the idea in a more explanatory way.

Before proceeding to any further discussion on the outputs, let us have a fine discussion on the UGTN architecture, which lies at the core of this model. We can also suspect UTGN to be an unsupervised or semi-supervised version of neural style transfer and generative adversarial network that is able to learn the crop on a CT scan image such that to provide a high probabilistic feature map for any further deep learning model. Thus, the architecture of UGTN should not only deal with the vanishing and exploding gradients but should be fast enough to carry out the task. That is why a residual utility connection is improvised taking inspiration from Resnet et al. [32]. Figures 9 and 10 showcase the layer-wise architecture for the same.

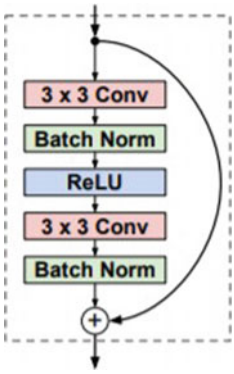
3.3 Transformer for Predictive Modeling

According to Vaswani et al. [23], attention is a very decisive mechanism where a sequence of vector representation is able to not only relate different vectors but

Fig. 9 UGTN network architecture

```
Input 3 × 256 × 256
Reflection Padding (40 × 40) 3 × 336 × 336
32 × 9 × 9 conv, stride 1 32 × 336 × 336
64 × 3 × 3 conv, stride 2 64 × 168 × 168
128 × 3 × 3 conv, stride 2 128 × 84 × 84
Residual block, 128 filters 128 × 80 × 80
Residual block, 128 filters 128 × 76 × 76
Residual block, 128 filters 128 × 72 × 72
Residual block, 128 filters 128 × 68 × 68
Residual block, 128 filters 128 × 64 × 64
64 × 3 × 3 conv, stride 1/2 64 × 128 × 128
32 × 3 × 3 conv, stride 1/2 32 × 256 × 256
3 × 9 × 9 conv, stride 1 3 × 256 × 256
```

Fig. 10 Residual block



enhance its relational feature embeddings to the final output. Although the transformer architecture we reuse different feature maps for machine translation use-cases. However, we have divided the image into 16 × 16 lateral image vectors and then fed into the network. Figure 11 elaborates the transformer architecture as described in the paper attention is all you need.

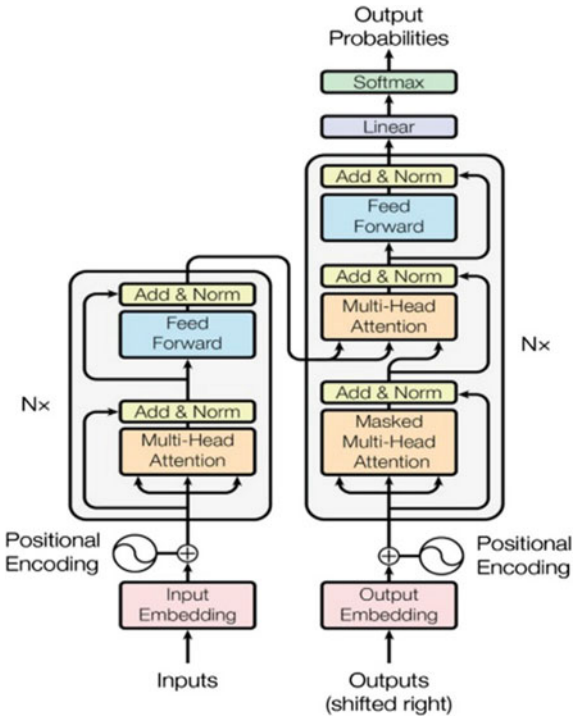


Fig. 11 Transformer architecture

In relevance to this particular use case, in a brief note, the transformer architecture abstracts the image feature based on its peers or neighbors. In simpler words, one can say the idea is to extract high-dimensional feature representation along with spatial space representation, so to extract deep networked features. We have used eight layers of encoder and eight layers of decoder. The reconstructed feature space embedding is probed in decoder along with encoder-wise feature map. This enhances mainly the following:

1. After the output of UGTN, the cropped image features which shall correlate to the final feature extractor of the transformer shall lie in similar vicinity due to its patchy nature.
2. This high-dimensional feature entanglement in the image feature space not only is a key finding of this research but is pretty useful in reducing computation in the transformer.
3. A similar example is when an radiopathy lab technician uses this short-term understanding of pneumonia feature and then further generalizes the concept by looking at close proximity of those features. Thus, this is the same way a sequence-to-sequence model is built.

Till this point, we have presented a fine understanding as to why we are biased in using transformers than normal CNN or transfer learning architectures or object detection methods like our predecessors. In the upcoming section, we shall showcase

Table 4 Pneumonia versus normal band classification

Network	Accuracy	Sensitivity	Specificity
ResNet-50	0.76	1.00	0.62
ResNet-18	0.74	1.00	0.61
Mobile Net-v2	0.72	1.00	0.59
VGG	0.73	1.00	0.60
AlexNet	0.76	1.00	0.63
DenseNet-121	0.69	1.00	0.57
Our	0.81	1.00	0.71

Table 5 Subclasses of pneumonia classification

Network	Accuracy	Sensitivity	Specificity
ResNet-50	0.89	1.00	0.86
ResNet-18	0.88	1.00	0.86
Mobile Net-v2	0.89	0.98	0.87
VGG	0.88	0.96	0.83
AlexNet	0.85	1.00	0.88
DenseNet-121	0.90	0.99	0.87
Our	0.92	0.98	0.88

the results and analyze the output of this novel transformer-based cephalometric algorithm, with other existing techniques.

4 Results

Based on our crop close hypothesis, the utility of UGTN algorithm has showcased well versed in the accuracy of detecting against coronavirus pneumonia as well as viral pneumonia for mycoplasma. Tables 4 and 5 advocate the same based on qualitative performance analysis.

5 Conclusion

It is evident that all different types of pneumonia have similar and confusing symptoms. During medical diagnosis, a radiological lab technician uses 4 min per sample to determine which pneumonia subclass is present or is the sample normal. However, even this tedious process is not without fault, and accuracy of 97.64% is observed by humans which is variable to a large extent on experience and data availability. However, in the paper, we identify this problem and try to produce an architecture

using unsupervised generative transfusion network (UGTN) along with state-of-the-art transformers using eight layers encoder and decoders to achieve an accuracy of 92 and 81% according to Tables 4 and 5 which is comparable and in some cases superior to our predecessor networks. Finally, a rigid UTGN architecture is proposed which is a mutant of style transfer family with traits of generative adversarial networks. Overall, this research shall help in further engineered advancements using unsupervised learning ideas to port in biomedical classification. Also, it shall demonstrate the ability of unsupervised probability distribution localized cropping in future innovations.

References

1. Wang C, Liu X, Zhang K, Shen J, Zhang K, Li Z, Sang Y, Liang W, Wu X, Wang K, Zha Y et al (2020) Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of COVID-19 pneumonia using computed tomography. Cell
2. Chowdhury ME, Rahman T, Khandakar A, Islam KR, Kadir MA, Islam KF, Mahbub ZB, Kashem S (2020) Transfer learning with deep convolutional neural networks (CNN) for pneumonia detection using chest X-ray. Appl Sci 10(9):3233
3. Elazab A, Wang C, Hu Q, Wu J (2016) A novel approach for tuberculosis screening based on deep convolutional neural networks. In: Medical imaging 2016: computer-aided diagnosis, vol 9785. International Society for Optics and Photonics, p 97852W
4. Serener K, Serener A (2018) Automated age-related macular degeneration and diabetic macular edema detection on OCT images using deep learning. In: IEEE 14th international conference on intelligent computer communication and processing (ICCP). IEEE, pp 265–269
5. Serte S, Serener A (2019) Dry and wet age-related macular degeneration classification using OCT images and deep learning. In: Scientific meeting on electrical-electronics and biomedical engineering and computer science (EBBT). IEEE, pp 1–4
6. Transfer learning for early and advanced glaucoma detection with convolutional neural networks. In: Medical technologies congress (TIPTEKNO). IEEE, pp 1–4
7. Serener A, Serte S (2019) A generalized deep learning model for glaucoma detection. In: 3rd international symposium on multidisciplinary studies and innovative technologies (ISMSIT). IEEE, pp 1–5
8. Serte S, Serener A (2020) Geographic variation and ethnicity in diabetic retinopathy detection via deep learning. Turk J Electr Eng Comput Sci 28(2):664–678
9. de Leeuw F-E, van Ginneken B, Bergkamp M, Wissink J, Obels J, Keizer K, Ghafoorian M, Heskes T, Marchiori E, Karssemeijer N et al (2017) Deep multi-scale location-aware 3D convolutional neural networks for automated detection of lacunes of presumed vascular origin. NeuroImage: Clin 14:391–399
10. Ying S, Shi J, Zheng X, Li Y, Zhang Q (2017) Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of Alzheimer's disease. IEEE J Biomed Health Inform 22(1):173–183
11. Esmail P, Serener A, Kaymak S (2018) Deep learning for two step classification of malignant pigmented skin lesions. In: 14th symposium on neural networks and applications (NEUREL). IEEE, pp 1–6
12. Serener A, Serte S (2019) Keratinocyte carcinoma detection via convolutional neural networks. In: 3rd International symposium on multidisciplinary studies and innovative technologies (ISMSIT). IEEE, pp 1–5
13. Pacheco KD, Pekala M, Burlina PM, Bressler NM, Joshi N, Freund DE (2017) Automated grading of age-related macular degeneration from color fundus images using deep convolutional neural networks. JAMA Ophthalmol 135(11):1170–1176

14. Huang M, Diao K-y, Lin B, Liu C, Lee H-C, Xie Z, Ma Y, Robson PM, Chung M, Mei X et al (2020) Artificial intelligence–enabled rapid diagnosis of patients with COVID-19. Nat Med 1–5
15. Chun D, Butt C, Gill J, Babu BA (2020) Deep learning system to screen coronavirus disease 2019 pneumonia. Appl Intell 1
16. Brunese L, Reginelli A, Mercald F, Santone A (2020) Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. Comput Methods Programs Biomed 105608
17. Hu S, Gao Y, Niu Z, Jiang Y, Li L, Xiao X, Wang M, Fang EF, Menpes-Smith W, Xia J et al (2020) Weakly supervised deep learning for COVID-19 infection detection and classification from CT images. IEEE Access
18. Yan T (2020) CCAP. [Online]. Available: https://doi.org/10.21227/ccgv-5329
19. Goodfello IJ, Ozair S, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Courville A, Bengio Y Generative adversarial networks
20. Suwajanakorn S, Kemelmacher-Shlizerman I, Seitz SM (2014) Illumination-aware age progression. In: Proceedings of computer vision and pattern recognition
21. Tong Y, Park U, Jain AK (2010) Age-invariant face recognition. IEEE Trans Pattern Anal Mach Intell 32(5):947–954
22. Antipov G, Baccouche M, Dugelay J (2017)Face aging with conditional generative adversarial networks. In: IEEE international conference on image processing (ICIP), Beijing, pp 2089–2093. https://doi.org/10.1109/ICIP.2017.8296650
23. Uszkoreit J, Gomez AN, Vaswani A, Parmar N, Shazeer N, Jones L, Polosukhin I, Kaiser L Attention is all you need. https://arxiv.org/abs/1706.03762
24. Yang Y, Jing Y, Yu Y, Ye J, Feng Z, Song M (2020) Neural style transfer: a review. IEEE Trans Visual Comput Graphics 26(11):3365–3385. https://doi.org/10.1109/TVCG.2019.2921336
25. Li L, Wu X, Niu M, Wang L, Hui H, He B, Li L, Yang X, Li H, Tian J et al (2020) Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: a multicentre study. Euro J Radiol 109041
26. Wong PK, Yan T, Wang J, Ren H, Wang H, Li Y (2020) Automatic distinction between COVID-19 and common pneumonia using multiscale convolutional neural networks on chest CT scans. Chaos, Solitons Fractals 140:110153
27. Ardakani AA, Acharya UR, Kanafi AR, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. Comput Biol Med 103795
28. Wang M, Wang S, Li X, Wu Q, Zha Y, Li W, Niu M, Qiu X, Li H, Yu H et al (2020) A fully automatic deep learning system for COVID19 diagnostic and prognostic analysis. Euro Respir
29. Pereira RM, Bertolini D, Teixeira LO, Silla Jr CN, Costa YM (2020) COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios. Comput Methods Programs Biomed 105532
30. Johnson J, Alahi A, Fei-Fei L (2016)Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the European conference on computer vision, pp 694–711
31. Ecker AS, Gatys LA, Bethge M (2016) Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2414–2423

Minimum Weight Dominating Set Using Chemical Reaction Optimization Algorithm



Pritam Khan Boni and Md. Rafiqul Islam

Abstract The set of vertices of a graph that can cover all the vertices of the graph is called the dominating set. If the dominating set contains a minimum number of vertices and the total weights of vertices are minimum then it will be the minimum weight dominating set (MWDS). It preserves many application areas encompassing data mining, social networks, and many other networks. As day-by-day the instances for networks are increasing, this problem has gained enormous attention. To resolve this issue, a new methodology using chemical reaction optimization algorithm (CRO) with a repair operator is proposed in this paper. To repair the solution, in the repair operator, we introduced scale-based searching. Here, the properties which are considered for obtaining a better solution are scaled. The method outperforms any other algorithms according to the comparative analysis for different sets of node-weight pairs from different graphs of the benchmark dataset. Besides this, the proposed method obtains the minimum weight very fast. The running time of MWDS_CRO is significantly lower than the state-of-the-art method.

Keywords Chemical reaction optimization algorithm · Graph theory · Minimum weight dominating set

1 Introduction

According to graph theory, a dominating set refers to the vertices of an undirected graph where every vertex of the graph either belongs to the set or the neighbor of the vertices of the set. The minimum weight dominating set (MWDS) refers to

Supported by ICT-Division, Ministry of Posts, Telecommunications and Information Technology, Bangladesh

P. K. Boni (✉) · Md. R. Islam
Khulna University, Khulna 9208, Bangladesh
e-mail: pritam1318@cseku.ac.bd

Md. R. Islam
e-mail: dmri1978@cseku.ac.bd

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_4

dominating set with the minimum total weight of vertices. The main concern of the MWDS problem is to find the minimum dominating set with the minimum total weight of the vertices, where there is some associated positive values with every vertex of the graph as weight.

Let, there is an undirected graph, $G = (V, E)$ where $V = \{v_1, v_2, v_3, \ldots, v_n\}$ is the set of vertices and $E = \{e_1, e_2, e_3, \ldots, e_m\}$ is the set of edges. Each edge $e = \{u, v\}$ connects two vertices u and v together and these vertices are the endpoints of e. According to this problem, we have to know the neighbor of any vertex and to include those vertices which have a maximum neighbor to get the minimum dominating set.

For getting the minimum weight dominating set the following equations are introduced.

$$\min \sum_{v \in D} \omega(v) x(v) \tag{1}$$

$$\sum_{k \in N[i]} x(k) \geq 1, \forall_i \in D \tag{2}$$

$$x(v) \in \{0, 1\} \tag{3}$$

Here, *N[i]* denotes the closed neighbor of vertex i (i.e. $N[i] = N(i) \cup \{i\}$), D denotes the dominating set and $D \in V$, ω represents the weight of each vertex and x denotes the decision variable where it is set to 1 if the vertex is included in the dominating set, otherwise, it is 0 [9].

This emerging theoretical topic has so many areas of applications including data mining, the study of social networks and influence propagation [2], protein interaction networks [3, 4], design of wireless sensor networks [1] and covering codes [5]. Though the minimum weight dominating set has been introduced to solve many real-life problems, the solution to this problem using efficient meta-heuristic is an emerging one [6]. As the MWDS problem is NP-hard [7], it is inevitable to solve this problem using a meta-heuristics algorithm to get better results. To resolve this issue in recent times a number of meta-heuristic algorithms are proposed by many researchers. Albeit of various methodologies using meta-heuristic algorithms to solve this problem, for large-scaled instances, the solutions are still not satisfactory. To cope up with this issue, in this paper, a chemical reaction optimization-based meta-heuristic algorithm has been proposed due to its superiority over many other NP-hard and NP-complete problems [8]. The dominance of CRO in solving NP-hard and NP-complete problems is proved by the comparison with many existing evolutionary approaches, some of them include CROG for 0–1 Knapsack problem [16], network coding optimization problem [17], generalized vertex covering problem [18], quadratic assignment problem [19].

2 Related Work

In recent years, MWDS showed its necessity towards many social data and network-related problems. Thus it has been drawn attention to many researchers, as it is quite impossible to work with the data of mammoth size in a graph if each vertex has to be traversed one by one using an exact algorithm. Hence to solve this problem researchers proposed different methods which include different optimization algorithms to minimize the weights of the dominating set.

Lin et al. [10] proposed a hybrid memetic algorithm to solve the MWDS problem. In their work, modified unconstrained 0–1 problem was hybridized with hybrid memetic algorithm(HMA). Another meta-heuristic algorithm based on ant colony optimization (ACO) was proposed to solve the MWDS problem, where the weights of vertices are covered or not, has been taken into account [12]. On the other hand, Potluri et al. [13] suggested a hybrid algorithm, denoted as the ACO-PP-LS algorithm. In this algorithm, ACO has been used for considering the pheromone deposit on the node and a pre-processing step immediately after pheromone initialization. CC^2FS is another algorithm by which researchers tried to cope up with the MWDS problem. In 2017, Wang et al. [14] suggested a new variant of configuration checking heuristic algorithm, named as two-level configuration checking(CC^2) and combined it with a novel scoring function based on the frequency of being uncovered of vertices to solve this problem. Albeit of all these solutions, the urge for solving this problem has not been decayed due to its characteristics of NP-hardness and evolution of many other optimization algorithms and their probability to choose a better solution in NP-hard problems. Therefore, Lin et al. [11] proposed a Binary Particle Swarm Optimization (BPSO) algorithm for solving the MWDS problem. In this work, they used BPSO for designing a new position updating rule and iterated greedy tabu search to enhance the solution quality quickly. Another hybridized method with the help of tabu search has been evolved by Albuquerque et al. [6]. In this method, they suggested a matheuristic model in which tabu search has been linked with integer linear programming (ILP), and perturbation has been used for exploring the solution space. Although the existing methodologies are quite mesmerizing to find the minimum weight dominating set the efficiencies are still not satisfactory. Thus we are intended to develop new method to obtain satisfactory efficiencies.

3 Proposed Method Using Chemical Reaction Optimization

In the proposed methodology, we have implemented the chemical reaction optimization (CRO) algorithm to solve the MWDS problem. CRO algorithm includes three stages, initialization, iteration, and the final stage. In the initial stage, all the essential parameters are initialized and the search space is created. In the iteration stage, one of the four molecular reactions is conducted. After satisfying the termination criteria the final stage begins. In this stage, the best solution is selected among all the indi-

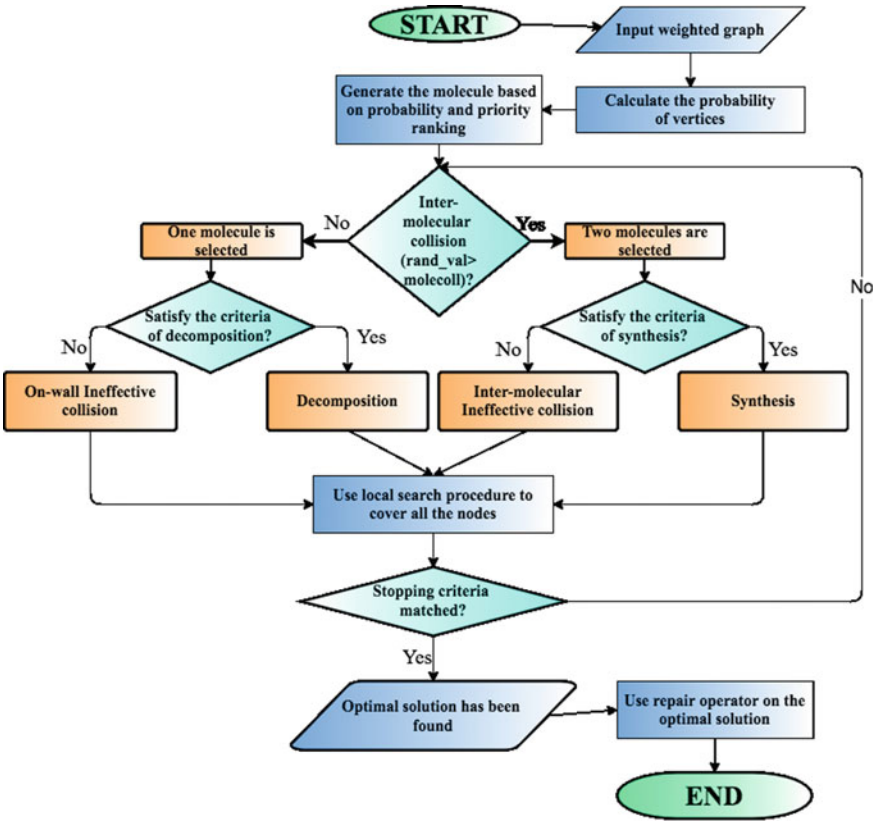


Fig. 1 Flowchart of CRO for MWDS problem

viduals and the algorithm terminates. Figure 1 shows the flowchart of our proposed method.

3.1 Population Generation

There are three main approaches to find the candidate solutions for the MWDS problem which are penalty strategy [6], preserving strategy [13] and repair strategy [15]. To generate a molecule, a penalty strategy is followed in our proposed algorithm. Different constraints are added to reduce search space and those vertices are added randomly into the candidate solution which has the maximum number of neighbors. It means the vertices, which have the most neighbors get the highest priority. The priority has been defined as the probability score. The probability score is calculated as (5).



Fig. 2 Example of a molecule

$$\text{cost}_i = \frac{\text{degree}_i \times \sum_{j \in N[i]} \text{weight}_j}{\text{weight}_i} \tag{4}$$

$$\text{probability}_i = \frac{\text{cost}_i}{\sum_{j \in N[i]} \text{cost}_j} \tag{5}$$

The probability score is the ratio of the cost of a vertex and the summation of the cost of closed neighbors of that vertex. The cost of a vertex is calculated as (4).

If any vertex is taken in the candidate solution then it is marked as 1, otherwise, it is marked as 0. Figure 2 represents the structure of a molecule (i.e. candidate solution).

To choose an active vertex (i.e. the taken vertex which is represented as 1) randomly based on priority we compared vertex-edge ratio with a constant value of 0.3. If the ratio is less than 0.3 then the percentage of vertices has been chosen according to the ratio from the priority rank. For example, if the number of vertices of a graph is 50 and the number of edges is 1000 then the vertex-edge ratio is $\frac{50}{1000} = 0.05$. So from the priority list, active vertices are selected randomly from the first 5% of high-ranked vertices. Otherwise, active vertices are chosen from 30% of high-ranked vertices by default. The priority ranking of a vertex is defined based on the probability score. The vertex with high probability score is interpreted as high priority ranked vertex and the vertex with lower probability score is interpreted as low priority ranked vertex. The priority list is, the list of probability values of vertices in descending order.

In this selection process, though chosen vertices have the dominating power most, they can not cover all the vertices of the graph. To resolve this issue and minimize the total weight the nodes which are neither the neighbor of the active nodes nor the neighbors of the neighbors are chosen as spared nodes. From this list of spared nodes, active nodes are selected based on randomness. This randomness is conditioned as, if the random value between 0 and 1 is greater than or equals 0.5 then the nodes are selected as active nodes otherwise, they are inactive. Here, if all the nodes are covered then the final molecule structure has been published for further processing.

3.2 Local Search Procedure

To solve the issue regarding uncovered vertices and to get the better molecule local search procedure has been applied. Here we search according to the greedy procedure to eradicate uncovered vertices following the technique of Potluri et al. [13]. As they showed four greedy processes and from them, the fourth process gives the best

empirical result, so we use only this one to find the proper vertices. According to this procedure, the greedy value of a vertex is calculated as (6).

$$g(v) = \deg(v) \times \frac{\sum_{i \in N(v)} W_i(v)}{\omega(v)} \tag{6}$$

Here, $\sum_{i \in N(v)} W_i(v)$ represents the summation of all uncovered neighbors of a single vertex, $\omega(v)$ denotes weight of the vertex and $deg(v)$ represents degree of the vertex.

From these vertices the vertex with maximum greedy value has been chosen and then the vertices are flipped to 1 in the solution set if $g(v) \geq \delta \times \max g(v), \forall\{v \in uv\}$. Here, uv represents the set of uncovered vertices. From this procedure, a molecule with minimum potential energy (i.e. minimum weight) has been found and the molecule is returned to do a further operation.

3.3 Neighborhood Search Operator

This is a local optimal node searching technique. In this technique, local neighborhood nodes have been searched. To initialize this process a node has been chosen randomly from active nodes. After that, the neighbors of the selected node have been searched which weight is smaller than the weight of this node. If any neighbor node is being chosen then the position of the molecule becomes 1 (i.e. the node becomes active) and the previous node becomes 0. If the position remains the same then it randomly chooses another position and again imitates the previous procedure. In the case of failure of finding any new node after trying the searching procedure for every node the process terminates. Figure 3 shows the searching procedure of the neighborhood search operator.

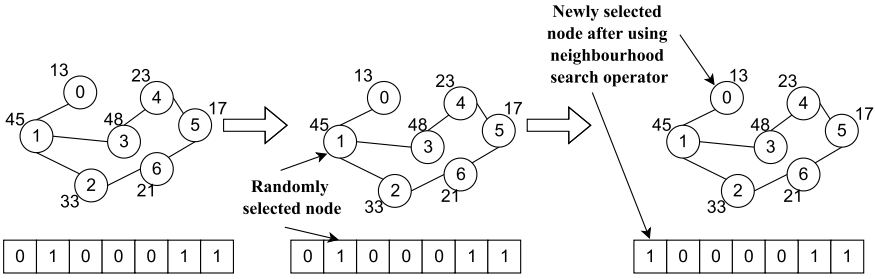


Fig. 3 Illustration of neighborhood search operator

3.4 Operator Design

We have represented four reaction operators in our proposed algorithm, which are explained in the following sub-sections.

3.4.1 On-Wall Ineffective Collision

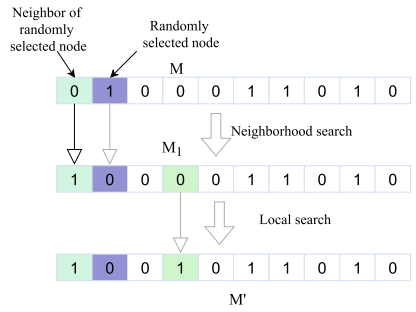
This is a molecular reaction that is used to modify the structure. Here, one position of the molecule is randomly chosen and the value of the position is changed according to the neighborhood search operator. Then to cover all the nodes (if any node remains uncovered) local search procedure has been applied to the molecule to generate a feasible solution. Figure 4a refers to the on-wall ineffective collision for MWDS problem where M represents the molecule (i.e. solution) and $M^{'}$ refers to the newly generated molecule after using both neighborhood search and local search.

3.4.2 Decomposition

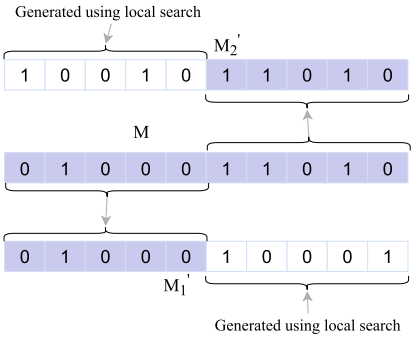
In this reaction, a molecule is divided into two parts. From the molecule, a portion (or block) is randomly selected and then the selected block and the rest are divided into two. These two portions generate two new molecules. The rest part of the new molecules (the portions do not come from the previous molecule) have been generated according local search procedure. According to Fig. 4b, the molecule is divided into two from the middle. In solution1 ($M_1^{'}$) the last portion of the molecule is the same as the first portion of the parent molecule, the rest is generated according to local search procedure and in solution2 ($M_2^{'}$) the last portion of the molecule is the same as the last portion of the parent molecule, the rest is generated according to local search procedure.

3.4.3 Inter-molecular Ineffective Collision

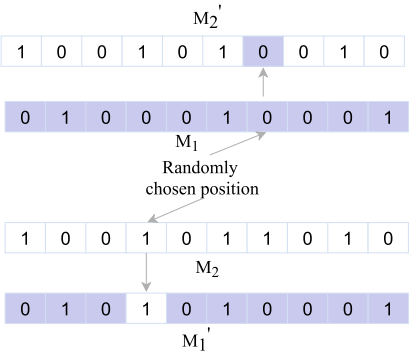
In this reaction, two structures have been used. From both structures, a position is selected randomly. Then the values of the selected positions are interchanged to generate new structures. Finally to cover all the nodes (if any node remains uncovered) local search procedure has been applied on both molecules to generate feasible solutions. Figure 4c refers the inter-molecular ineffective collision for MWDS problem where M_1, M_2 represent the molecules (i.e. solutions) and $M_1^{'}$, $M_2^{'}$ refer to the newly generated molecules.



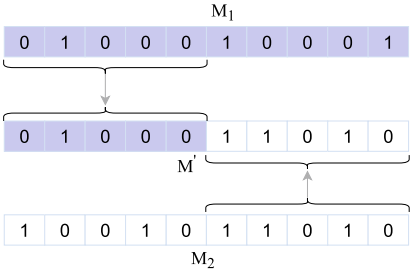
(a) Illustration of on-wall ineffective collision



(b) Illustration of decomposition



(c) Illustration of inter-molecular ineffective collision



(d) Illustration of synthesis

Fig. 4 Four rudimentary operators of CRO

3.4.4 Synthesis

Two molecules are combined to generate a new molecule in a synthesis operation. For example, two molecules M_1 and M_2 from the population are combined to form a new molecule $M^{'}$. Then if there are any uncovered vertices they are covered with the help of local search procedure. Let, ith position has been selected randomly. The values from the first position to ith position of M_1 are transferred to the first part of the new molecule. The values from $(i + 1)$th position to the last position of M_2 are transferred to the second part of the new molecule. Thus a new molecule $M^{'}$ has been generated. It is worked as the opposite of the decomposition reaction. Figure 4d shows the illustration of synthesis reaction.

3.5 Repair Operator

To find a better solution, a repair operator has been used on the molecule structures which are obtained after the implementation of CRO operators. Initially from the minimum structure 40% of nodes with maximum weight have been removed. To make the solution feasible as all nodes have to be covered, three properties of nodes have been selected to take the decision of choosing any node for molecule structure. The properties of nodes are the weight of the node, uncovered neighbor of the node, and the ratio of weight and uncovered neighbor nodes. The properties of nodes are scaled based on a standardized value. The properties are scaled as (7), (8), and (9).

$$\text{Scaled}_{\omega_i} = \frac{\omega_{\min}}{\omega_i} \tag{7}$$

$$\text{Scaled}_{\overline{\text{Cov}(N(i))}} = \frac{\overline{\text{Cov}(N(i))}}{\overline{\text{Cov}(N(max))}} \tag{8}$$

$$\text{Scaled}_{\text{ratio}_i} = \frac{\text{ratio}_{\min}}{\text{ratio}_i} \tag{9}$$

Here ω_i refers the weight of uncovered node i and ω_{min} refers the node with minimum weight from all uncovered nodes. Ratio of node i is calculated as (10).

$$\text{ratio}_i = \frac{\omega_i}{\overline{\text{Cov}(N(i))}} \tag{10}$$

The summation of these three scaled values for every nodes are calculated. The node with maximum summation is taken as active nodes if the node has more than one uncovered neighbors. Otherwise from all neighbors the node with minimum weight has been taken as active node.

3.6 Parameter Settings

As the initialization of parameters in CRO is not definite, so getting the proper parameter values is a challenging task of finding the best molecule. In the proposed work, we are trying to find the best molecule with the minimum weight by tuning parameters. The tuned values of parameters are kinetic energy loss rate = 0.1, molecoll = 0.1, buffer = 0, hit difference = 0, α = 10000, β = 1000 and kinetic energy = 100.

4 Experimental Results

To find the minimum weight dominating set, CRO-based optimization approach has been used in our proposed method and the method is used on the benchmark datasets referred to as T1 instances. This dataset is used on previous experiments based on heuristic algorithms [6, 11–14] to solve MWDS problem. The dataset is collected from (https://w1.cirrelt.ca/ vidalt/en/research-data.html). The T1 dataset encompassed small, medium, and large graph instances where SMPI class is the instances for small and medium graphs which obtains 50 to 250 nodes with 50 to 5000 edges as well as LPI class is the instances for large graphs which obtains 300 to 1000 nodes with 300 to 20000 edges. These graphs are generated with random node weights in the interval between 20 to 70 and every instance is generated based on 10 random seeds (i.e. for every nodes-weights pair there are 10 random graphs with different weights).

To experiment on this dataset python 3.7 and 2.7 GHz intel core i5 processor was used. To compare running time HTS-DS was also implemented in same environment. The experimented results of our proposed architecture are compared with the results of some state-of-the-art meta-heuristic algorithms. Tables 1 and 2 show the comparative analysis of average outcome between various algorithms with our proposed method as well as the time comparison between HTS-DS with our proposed method for different graphs of SMPI class and LPI class respectively of T1 instance.

Figure 5 shows the comparison of average weights among various proposed algorithms and Fig. 6 shows the comparison of CPU time of HTS-DS with our proposed method.

From Fig. 5 the superiority of MWDS_CRO is unambiguous as it outperforms any other methods and obtains the best-known result for almost all graphs of the experimented dataset. Alongside, Fig. 6a, b prove the dominance of MWDS_CRO over the HTS-DS algorithm regarding time issue for classes SMPI and LPI, respectively. For most of the graphs, the running time of the proposed method is far better than HTS-DS. As the flexible characteristics of CRO support both exploration and exploitation, it is comparatively easy to gain better performance using CRO in optimization problems. For better solution CRO searches in local space through on-wall and inter-molecular ineffective collision. If it fails to get a better result then it searches globally by the other two operators. These processes always make convergence to the optimal solution and thus obtain the minimum weight dominating set in limited time.

Table 1 Type T1, Class SMPI - Comparison of MWDS_CRO for various parameters with recent state-of-the-art algorithm

|V|	|E|	HMA	CC^2 FS	FBPSO	HTS-DS		MWDS_CRO (Proposed)	
		Avg	Avg	Avg	Avg	T(s)	Avg	T(s)
50	50	531.8	531.3	531.3	531.3	0.1	**531.3**	**0.1**
	100	371.2	370.9	370.98	370.9	0.1	**370.9**	**0.1**
	250	176.4	175.7	175.7	175.7	0.1	**175.7**	**0.1**
	500	96.2	94.9	94.9	94.9	0.1	**94.9**	**0.08**
	750	63.3	63.3	63.16	63.1	0.1	**63.1**	**0.07**
	1000	41.5	41.5	41.5	41.5	0.06	**41.5**	**0.04**
100	100	1064.9	1061.0	1062.99	1061.0	0.2	**1061.0**	**0.2**
	250	623.1	618.9	619.48	618.9	0.5	**618.9**	**0.4**
	500	356.8	355.6	355.74	355.6	0.6	**355.6**	**0.4**
	750	258.4	255.8	255.95	255.8	0.7	**255.8**	**0.5**
	1000	205.9	203.6	203.6	203.6	0.9	**203.6**	**0.6**
	2000	107.8	107.4	107.88	107.4	0.64	**107.4**	**0.4**
150	150	1585.3	1580.5	1585.71	1580.5	0.5	**1580.5**	**0.5**
	250	1231.8	1218.2	1223.31	1218.2	0.6	**1218.2**	**0.6**
	500	749.5	744.6	747.45	744.6	3.4	**744.6**	**1.2**
	750	550.2	546.1	548.15	546.1	7.2	**546.1**	**1.8**
	1000	435.7	432.9	433.93	432.8	8.3	**432.8**	**1.7**
	2000	244.2	240.8	241.47	240.8	8.1	**240.8**	**1.5**
	3000	168.4	166.9	167.53	166.9	7.9	**166.9**	**1.4**
200	250	1912.1	1910.4	1918.82	1909.7	0.6	**1909.7**	**0.6**
	500	1245.7	1232.8	1239.71	1232.8	2.0	**1232.8**	**1.5**
	750	926.1	911.2	918.47	911.2	7.9	**911.2**	**1.7**
	1000	727.4	724.0	727.09	723.5	7.9	**723.5**	**1.6**
	2000	421.2	412.7	415.46	412.7	8.2	**412.7**	**1.4**
	3000	297.9	292.8	294.03	292.8	7.5	**292.8**	**1.1**
250	250	2653.4	2633.4	2649.73	2633.0	1.1	**2633.0**	**1.1**
	500	1828.5	1805.9	1813.03	1805.9	2.4	**1805.9**	**2.3**
	750	1389.4	1362.2	1375.32	1361.9	8.6	**1361.9**	**2.6**
	1000	1109.5	1091.1	1099.65	1089.9	8.3	**1089.9**	**2.3**
	2000	635.3	621.9	625.41	621.6	8.6	**621.6**	**2.1**
	3000	456.6	447.9	452.45	448.0	7.8	**447.9**	**1.7**
	5000	292.8	289.5	291.18	289.5	8.2	**289.5**	**1.3**

Table 2 Type T1, Class LPI - Comparison of MWDS_CRO for various parameters with recent state-of-the-art algorithm

|V|	|E|	HMA	CC^2FS	FBPSO	HTS-DS		MWDS_CRO (Proposed)	
		Avg	Avg	Avg	Avg	T(s)	Avg	T(s)
300	300	3199.3	3178.6	3203.62	3175.4	1.3	**3175.4**	**1.3**
	500	2464.4	2438.2	2453.57	2435.6	1.6	**2435.6**	**1.6**
	750	1884.6	1854.6	1870.8	1853.8	7.7	**1853.8**	**3.6**
	1000	1518.4	1495.0	1506.56	1494.0	8.9	**1494.0**	**3.8**
	2000	878.7	862.5	871.89	862.4	9.2	**862.4**	**3.3**
	3000	640.9	624.3	628.24	624.1	9.64	**624.1**	**4.04**
	5000	411.7	406.1	409.08	406.1	8.3	**406.1**	**2.6**
500	500	5392.1	5305.7	5392.45	5304.7	2.7	**5304.7**	**2.7**
	1000	3678.3	3607.8	3659.87	3607.6	10.4	**3607.6**	**10.4**
	2000	2223.7	2181.0	2210.28	2176.8	10.8	**2176.8**	**10.8**
	5000	1074.2	1043.3	1054.88	1042.3	11.2	**1042.3**	**7.8**
	10000	595.4	587.2	594.32	587.2	10.16	**587.2**	**6.0**
800	1000	7839.9	7663.4	7771.52	7655.0	5.2	**7655.0**	**5.2**
	2000	5100.7	4982.1	5060.73	4987.3	14.6	**4985.3**	**12.3**
	5000	2495.7	2441.2	2470.38	2432.6	14.4	**2432.6**	**12.1**
	10000	1549.8	1395.6	1417.16	1393.7	14.2	**1393.7**	**10.6**
1000	1000	10863.3	10585.3	10785.37	10574.4	8.7	**10574.4**	**8.7**
	5000	3742.8	3671.8	3713.22	3656.6	16.7	**3656.6**	**12.2**
	10000	2193.7	2109.0	2132.76	2099.8	16.9	**2099.8**	**12.4**
	15000	1590.9	1521.5	1542.64	1519.7	17.2	**1519.7**	**13.1**
	20000	1263.5	1203.6	1215.98	1200.9	18.9	**1200.9**	**14.3**

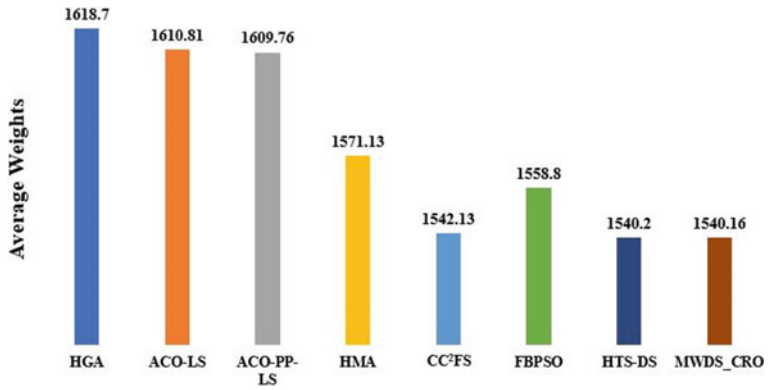
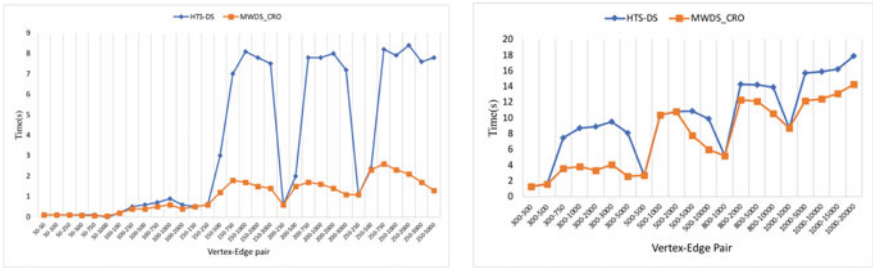


Fig. 5 Comparison of average weights of different algorithms



(a) Execution time of class SMPI (b) Execution time of class LPI

Fig. 6 Comparison of time between MWDS_CRO and HTS-DS

5 Conclusion

MWDS problem plays a vital role in solving many real-world problems including data mining, the study of social networks and influence propagation, design of wireless sensor networks, protein interaction networks, and covering codes. In short, it can be said that both mathematical and engineering fields have the importance of the study of the MWDS problem. Albeit all of these, interest in solving this problem is growing in recent times. Many researchers worked on this topic to get good results in minimum time, but further improvement is still possible. To rethink this problem, we intend to obtain better performance for every instance with help of a new algorithmic model. According to the experimented result, the average weight of the proposed architecture is 1540.16 which unquestionably glorifies our intention to achieve superior performance comparing any other algorithms. Alongside, our proposed method obtained the minimum time comparing with the state-of-the-art system. Although the methodology can prove its supremacy on a benchmark dataset, there are some other benchmark datasets to experiment with. We believe in the supremacy of the proposed method on these datasets too.

References

1. Yu J, Wang N, Wang G, Yu D (2013) Connected dominating sets in wireless ad hoc and sensor networks-a comprehensive survey. Comput Commun 36(2):121–134
2. Khomami MMD, Rezvanian A, Bagherpour N, Meybodi MR (2018) Minimum positive influence dominating set and its application in influence maximization: a learning automata approach. Appl Intell 48(3):570–593
3. Wuchty S (2014) Controllability in protein interaction networks. Proc Natl Acad Sci 111(19):7156–7160
4. Nacher JC, Akutsu T (2016) Minimum dominating set-based methods for analyzing biological networks. Methods 102:57–63
5. Östergård PR (1997) Constructing covering codes by tabu search. J Combinatorial Des 5(1):71–80

6. Albuquerque M, Vidal T (2018) An efficient matheuristic for the minimum-weight dominating set problem. Appl Soft Comput 72:527–538
7. Johnson DS, Garey MR (1979) Computers and intractability: a guide to the theory of NP-completeness. WH Freeman
8. Islam MR, Saifullah CK, Mahmud MR (2019) Chemical reaction optimization: survey on variants. Evol Intell 12(3):395–420
9. Alon N, Moshkovitz D, Safra S (2006) Algorithmic construction of sets for k-restrictions. ACM Trans Algorithms (TALG) 2(2):153–177
10. Lin G, Zhu W, Ali MM (2016) An effective hybrid memetic algorithm for the minimum weight dominating set problem. IEEE Trans Evol Comput 20(6):892–907
11. Lin G, Guan J (2018) A binary particle swarm optimization for the minimum weight dominating set problem. J Comput Sci Technol 33(2):305–322
12. Romania QS (2010) Ant colony optimization applied to minimum weight dominating set problem. In: Proceedings of the 12th WSEAS international conference on automatic control, modelling and simulation. Catania, Italy, pp 29–31
13. Potluri A, Singh A (2013) Hybrid metaheuristic algorithms for minimum weight dominating set. Appl Soft Comput 13(1):76–88
14. Wang Y, Cai S, Yin M (2017) Local search for minimum weight dominating set with two-level configuration checking and frequency based scoring function. J Artif Intell Res 58:267–295
15. Yuan F, Li C, Gao X, Yin M, Wang Y (2019) A novel hybrid algorithm for minimum total dominating set problem. Mathematics 7(3):222
16. Truong TK, Li K, Xu Y (2013) Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. Appl Soft Comput 13(4):1774–1780
17. Pan B, Lam AY, Li VO (2011) Network coding optimization based on chemical reaction optimization. In: 2011 IEEE global telecommunications conference-GLOBECOM 2011. IEEE, pp 1–5
18. Islam MR, Arif IH, Shuvo RH (2019) Generalized vertex cover using chemical reaction optimization. Appl Intell 49(7):2546–2566
19. Xu J, Lam AY, Li VO (2010) Parallel chemical reaction optimization for the quadratic assignment problem. In: World congress in computer science, computer engineering, and applied computing, Worldcomp 2010

A Hybrid Domain Fragile Watermarking Technique for Authentication of Color Images



Muzamil Hussan, Shabir A. Parah, Aiman Jan, and G. J. Qureshi

Abstract This paper presents a fragile watermarking method in which spatial, as well as frequency, domain is used for the embedding of data in color image planes. In this scheme, color images (general and medical) of size 512 × 512 are first separated into three planes. In the proposed method, two color image planes are used for embedding. The first plane, i.e., red plane is divided into four blocks, and embedding in this plane is done by using reversible watermarking based on histogram bin shifting (HBS). The second plane, i.e., the green plane is also divided into four blocks, then each block is separated into 16 × 16 blocks. Then, these blocks are further subdivided into 4 × 4 sub-blocks. Embedding in this plane is ensured by using discrete cosine transform (DCT) where coefficients are selected for embedding a fragile watermark. The third plane of an image is left unaltered. Using HBS and DCT together, the presented approach not only retrieves the data reversibly but can check the authenticity of the color images as well and, thus, validates security of the system. The presented method demonstrates an average peak signal to noise (PSNR) rate of 56.22 dB, while structural similarity index (SSIM) of 0.9985 is acquired which shows the superiority of the presented approach over previously existing colored watermarked images.

Keywords Spatial domain · Frequency domain · HBS · DCT · Fragile watermark · Authentication

1 Introduction

Due to the corona pandemic, everyone is working from home through the Internet. Consequently, transferring multimedia like video, audio, text, image is at a peak;

M. Hussan (✉) · S. A. Parah · A. Jan
Department of Electronics and Instrumentation Technology, University of Kashmir, Srinagar, India

G. J. Qureshi
Higher Education Department, Srinagar, J&K, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_5

for that reason, forgery or modification of data intentionally or unintentionally also arises at the same time because prowlers are always waiting for a moment, which may lead to economic loss or human life loss in case of a medical or defense emergency. In such situation, protection of multimedia like images and authenticating its content has become an important factor [1, 2]. For authentication of the images, digital watermarking is the paramount tool to protect the images from tampering or forgery, copyright protection, etc. Based on the embedding techniques, watermarking has been categorized into spatial, frequency, and hybrid domains. In the spatial domain which is also known as a pixel domain, embedding is done in image pixels. In the frequency domain which is also known as a coefficient domain, coefficients of the transformed image are used for the embedding process. And, the hybrid domain contains both spatial and frequency domains. For authentication purposes, digital watermarking can be sorted as robust, fragile, and semi-fragile watermarking techniques. For ownership verification and copyright applications, a robust watermark is generally used. A fragile watermark gets demolished even if a small change to the image occurs whether it is done deliberately or accidentally. To locate the tampered area in the digital images, semi-fragile watermark can be used to integrate the content [3, 4].

For authentication purpose, various methods have been proposed that is instigated in either the spatial domain or the frequency domain [5, 6]. Su et al. [7] proposed an algorithm designed for digital color images in which DC coefficients are used in the pixel domain. In this method, the color image is parted into three planes and all planes of the cover image into non-overlapping 8×8 sub-blocks. These sub-blocks are used to find DC coefficients in the pixel domain. After obtaining DC coefficients, modification of the original image pixels is done directly by incrementing the coefficients. Liu et al. [8] have proposed a watermarking algorithm for dual watermarks for color images, one watermark is used to check authentication, and the other is used for the copyright of the images. LSB substitution method is used for fragile watermarking which shows 49.75 dB and SSIM of 0.9968, and for copyright protection, DWT is used which is in the frequency domain. The average PSNR for the watermarked image after embedding dual watermarks is 40.32 dB. The et al. [9] give a method for digital color images where the optimal channel is used for improving the visual quality of the watermarked images. Robustness has also been considered against some geometrical and non-geometrical attacks. Abraham et al. [10] proposed a scheme for color images in the spatial domain. A simple image region detector has been used for the authentication of images. The logo is embedded in the blue plane which is insensitive to the human visual system. In [11], a blind digital watermarking scheme in the DWT and SVD domain has been presented. After performing a two-level DWT process on it, the watermark is embedded in all four sub-bands of the cover image. This method is not much secure and complex. In [12], an effective fragile watermarking scheme for color images has been implemented for tamper detection and correction of the image. In this scheme, OR operation and hierarchal authentication have been implemented. This scheme shows 44.55 dB PSNR of the watermarked images. In the presented paper, a fragile watermarking-based algorithm for color images has been proposed. The color image in the presented approach is partitioned into three planes

in which HBS and DCT are used in two planes separately. In the proposed scheme, reversible watermarking-based on histogram bin shifting (HBS) has been used in one plane which is reversible so that we can retrieve the image and data from this plane after extraction. Four equal blocks of the first plane are done before embedding the secret data. Second Plane is first divided into four equal blocks, then these blocks are separated into 16 × 16 blocks. These blocks are further segregated into 4 × 4 sub-blocks. DCT is applied to these 4 × 4 sub-blocks before embedding the fragile watermark, and the fragile watermark is embedded in the coefficients of the 4 × 4 blocks by using the pixel swapping method in the high-frequency coefficients, as high-frequency sub-bands of the block represent the edge information, and this band is very sensitive to lossy compression. To check the fragility of the image, we use a high-frequency band for watermark embedding. That is the way the technique is hybrid where two techniques are merged in one color image. The technique not only retrieves the secret data from the image plane but also detects whether the image has been tampered with or not by using a fragile watermark in the other plane of the color image. The proposed scheme shows an average PSNR of 56.22 dB and SSIM 0.9985.

The rest of the paper is prepared as follows. Section 2 provides the proposed work. Section 3 describes experimental results, and discussions have been explained in a detailed description. At last, the paper concludes in Sect. 4.

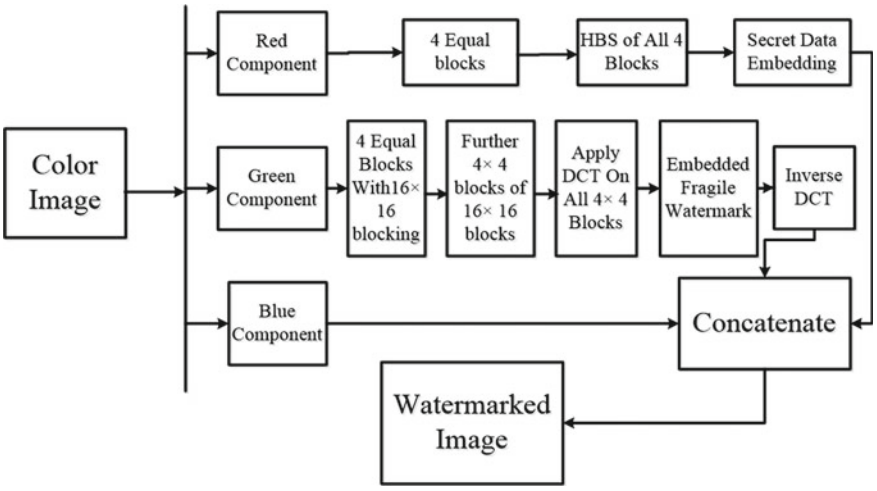


Fig. 1 Block description of the proposed work

2 Proposed Work

The block diagram of the proposed work has been shown in Fig. 1. The presented work is the hybrid of two domains, i.e., pixel as well as coefficient domain. Where in the spatial domain, we use a reversible watermarking algorithm based on histogram bin shifting (HBS), and in the frequency domain, we use discrete cosine transform-based embedding in the coefficients. In the proposed scheme, one plane is embedded with secret data and the other plane with a fragile watermark so that if any type of altering occurs in the image, we can detect the legitimacy of the image by checking the extracted watermark.

2.1 Embedding

The steps included in the embedding areas are:

(i) Distribute the digital color image having 512 × 512 size into three planes, i.e., red, green, blue.

(ii) Two planes are used for embedding wherein one plane, secret data are embedded and in another, fragile watermark is embedded.

(iii) Divide the first plane into four equal blocks, then embedding of secret data is done by using HBS. In HBS, the pixel value at minimum or zero frequency is said to be zero point value (Zv) and the pixel value at the maximum frequency is said to be peak point value (Pv). The pixel value in between those two points is shifted by one toward zero point value. It depends on whether the zero point value is greater or lesser than the peak point value. The bin is created near the peak point value and is used for embedding [4].

(iv) Divide the second plane into four equal blocks, then these four blocks are separated into 16 × 16 blocks. These blocks are further partitioned into 4 × 4 non-overlapping blocks. Perform DCT on each 4 × 4 sub-block. Let the coefficients of four DCT blocks be denoted by K1(g, h), K2(g, h), K3(g, h), and K4(g, h) where ‘g, h’ in a 4 × 4 sub-block gives the row and column positions of a DC coefficient where ‘g’ runs from 1:4 and ‘h’ runs from 1:4. To insert a watermark bit into two high-frequency DCT coefficients of a 4 × 4 block, two conditions are checked for each block. First, if K33 is less or equal to K44, then ‘0’ will be embedded, and if K33 is greater or equal to K44, then coefficients will swap, i.e., K44 will take K33’s place and vice versa. Then watermarked bit ‘1’ is embedded. This process will continue for all 4 × 4 blocks of the image plane.

(v) After performing the above-mentioned steps, all the planes of the color image are concatenated to form the watermarked color image.

2.2 Extraction

The watermarking extraction procedure comprises the same steps as that implicated at the embedding way until (iii) as described in an embedding way. In the first plane of the color image, the image blocks are scanned for peak point and zero point intensities, and accordingly, data bits are extracted. To extract data bits, image blocks peak point value and zero point value intensities are scanned, and if peak point value comes in the existence, then bit 1 is extracted, and for zero point value, 0 is extracted. After getting data from peak and zero intensities, original image plane is recovered back. The second plane of the color watermarked image is divided into four equal blocks. These four blocks are divided into 16 × 16 blocks, and these blocks are further subdivided into 4 × 4 sub-blocks. DCT procedure is performed the same way as mentioned in the embedding procedure. Only those coefficients are extracted which are changed throughout embedding, and we get this plane back and concatenating these two planes with the plane which is not embedded to get the color image back. If any malicious attack happens to this plane watermark which is not in its original form, it will be destroyed. That indicates that the image has tampered, and we can say that it is unauthentic.

3 Experimental Results

This section presents the investigational results of the presented methodology. General color images and medical images (OPENi database for medical images) with size 512 × 512 have been used in the presented method and are shown in Fig. 2. The parameters like PSNR and SSIM, mean square error (MSE) have been used to check the quality of the watermarked images [2]. To show the effectiveness of our scheme, PSNR and SSIM values are presented in Table 1. A comparison of the suggested scheme with other techniques has been shown in Table 2 and Fig. 3 individually. PSNR values show the superiority of the presented method when compared

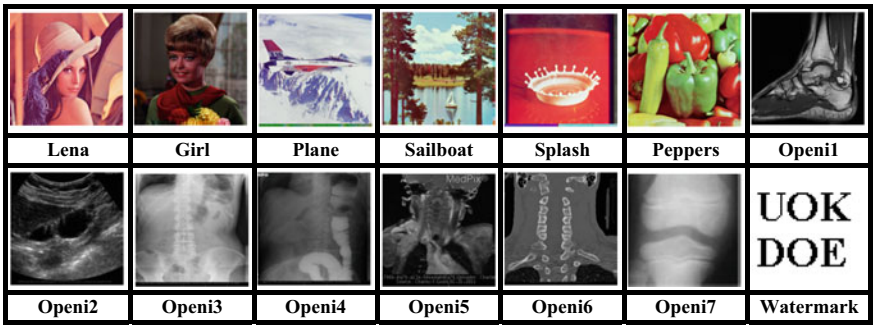


Fig. 2 Test images used in the proposed method

Table 1 Quality analysis of general and medical images

Image	PSNR	SSIM
Girl	55.32	0.9994
Sailboat	57.68	0.9999
Plane	58.89	0.9993
Lena	54.35	0.9965
Splash	59.81	0.9999
Peppers	58.87	0.9999
Openi1	54.88	0.9980
Openi2	56.79	0.9989
Openi3	56.88	0.9987
Openi4	53.35	0.9978
Openi5	53.30	0.9970
Openi6	56.97	0.9988
Openi7	53.78	0.9971
Average	56.22	0.9985

Table 2 Comparison of the PSNR (dB) values of the proposed method with other existing schemes

Image	Singh et al. [6]	Dadkhan et al. [5]	Garcia et al. [12]	Proposed
Lena	37.90	44.13	44.60	54.35
Sailboat	37.90	44.12	44.61	57.68
Peppers	37.79	44.06	44.54	51.23
Splash	37.84	44.08	44.47	59.81

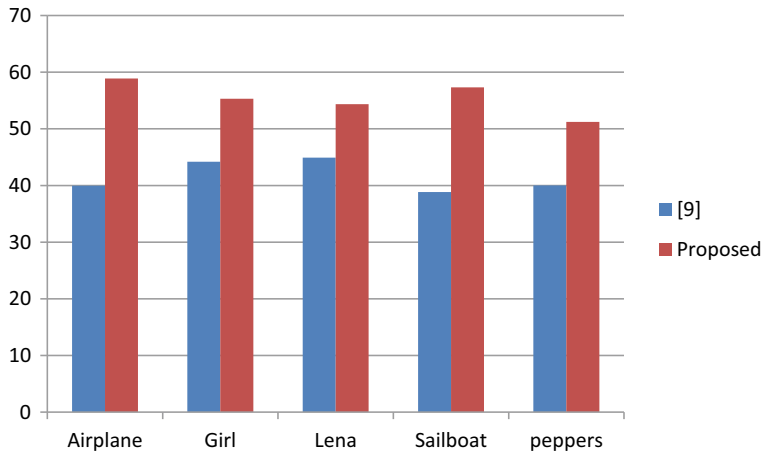


Fig. 3 Comparison of PSNR (dB) with other existing techniques

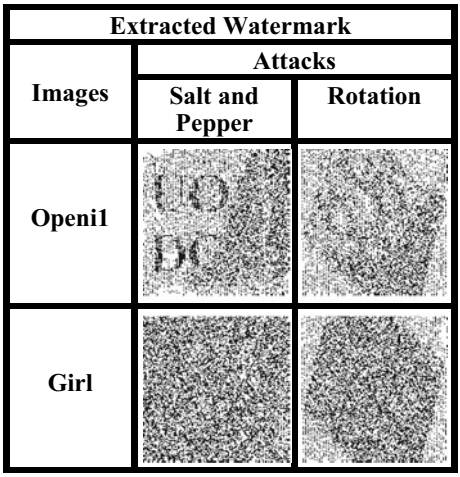


Fig. 4 Extraction of fragile watermark after subjected to various attacks

to further approaches. To verify the authenticity of the presented method, the received color images has been set under many attacks like salt and pepper, rotation, Gaussian noise among which two are shown in Fig. 4, and it can be effortlessly envisaged that the watermark acquired after the attack is unrecognizable that demonstrates that the approach can authenticate the data.

4 Conclusion

In the presented paper, a fragile watermarking approach has been implemented for color cover images authentication. In this scheme, two domains of an image have been used for embedding data into color cover image (general and medical). Single domain embedding is done in separate planes of the color image. The proposed method uses HBS for secret data embedding in one plane and DCT for fragile watermark embedding in another plane. The technique retrieves the secret data from the image plane reversibly using HBS and also detects whether the image has been tampered with or not using a fragile watermark. The watermarked color images show high perceptual quality, after embedding data (watermark and secret data) in the two planes of an image. The proposed scheme shows an average PSNR of 56.22 dB and SSIM is 0.9985. In the future, the presented method can be enhanced in tamper localization domain of the color images and can be implemented on other planes of the color images like YC_bC_r and HVS.

References

1. Hussan M et al (2021) Tamper detection and self-recovery of medical imagery for smart health. Arab J Sci Eng 1–17. https://doi.org/10.1007/s13369-020-05135-9AJSE-D-20-03588R1
2. Bhalerao S, Ansari IA, Kumar AA (2020) Secure image watermarking for tamper detection and localization. J Ambient Intell Human Comput 1057–0168. https://doi.org/10.1007/s12652-020-02135-3
3. Agrawal S, Kumar M (2016) Improved reversible data hiding technique based on histogram bin shifting. In: Proceedings of 3rd international conference on advance computing networking and informatics. Springer
4. Hussan M, Parah SA, Qurashi GJ (2019) Realization of a reversible data hiding framework using histogram bin shifting for color images. In: Fifth international conference on image information processing (ICIIP), Shimla, India, pp 221–224. https://doi.org/10.1109/ICIIP47207.2019.8985824
5. Dadkhah S, Manaf AA, Hori Y, Hassanien AE, Sadeghi S (2014) An effective SVD based image tampering detection and self-recovery using active watermarking. Signal Process 29(10):1197–1210. https://doi.org/10.1016/j.image.2014.09.001
6. Singh D, Singh SK (2016) Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. J Vis Commun Image Represent 38:775–789. https://doi.org/10.1016/j.jvcir.2016.04.023
7. Su Q, Niu Y, Wanga Q, Sheng G (2013) A blind color image watermarking based on DC component in the spatial domain. 124(23):6255–6260
8. Liu X, Lin C, Yuan S (2016) Blind dual watermarking for color images' authentication and copyright protection. IEEE Trans Circuits Syst Video Technol 28(5):1047–1055
9. The TH, Banos O, Lee S, Yoon Y, Tien TL (2016) Improving digital image watermarking by means of optimal channel selection. Expert Syst Appl 62:177–189
10. Abraham J, Paul V (2017) An imperceptible spatial domain color image watermarking scheme. J King Saud Univ-Comput Inf Sci 31(1):125–133
11. Furqan A, Kumar M (2015) Study and analysis of robust DWT-SVD domain based digital image watermarking technique using MATLAB. In: IEEE international conference on computational intelligence and communication technology (CICT) pp 638–644
12. Garcia JM et al (2020) An effective fragile watermarking scheme for color image tampering detection and self-recovery. Sign Proces Image Commun. https://doi.org/10.1016/j.image.2019.115725

IoT-based Carbon Monoxide Monitoring Model for Transportation Vehicles



I. Etier, A. Anci Manon Mary, and Nithiyananthan Kannan

Abstract The main objective of this work is to develop a prototype model for IoT-based carbon monoxide monitoring in a Raspberry Pi environment for transportation vehicle. Carbon monoxide (CO) is a hurtful gas conveyed by mostly consuming of various carbon-based stimulates. It can cause cerebral agony, ailment, regurgitating and chaos for individuals and essential to condition. From now on, this work finds a response for measure and stalls the level of carbon monoxide spread in the vehicle used for transportation on boulevards. This structure uses MQ-7 gas sensor to follow the substance of carbon monoxide that goes about as a toxic substance in the barometrical air. MQ-7 sensor is used to measure the carbon monoxide level and its clothing to the vehicle. If the carbon monoxide level is accessible, the message will be sent normally to the Pollution Control Board demonstrating the appearance of this level from the vehicle. Hence, the work environment has the ability to record the collection of proof against the vehicle that produces the over pollution. MQ-7 sensor measures the current estimation of the transmitting carbon monoxide from every vehicle, Wi-Fi modules are related with every Raspberry Pi-3, and it will send the message to the PHB-based IoT checking. It might be used in all around for seeing of pollution.

Keywords Internet of Things · Raspberry Pi-3 · CO sensor · Relay · LCD display

I. Etier
Electrical Engineering Department, The Hashemite University, Zarqa, Jordan

A. Anci Manon Mary
Department of EEE, Karpagam College of Engineering, Coimbatore, India

N. Kannan (✉)
Electrical Engineering Department, Faculty of Engineering, Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: nmajaknap@kau.edu.sa

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_6

1 Introduction

Continuous progress in advancement fundamentally focuses on controlling and seeing of different normal activities in order to show up at the human needs. Profitable watching structure is required to screen and assess the pollution levels if there ought to be an event of outperforming the embraced level of parameters (e.g., disturbance, CO and radiation levels). Air tainting infers the introduction of solid buildup particles, liquid globules or gases into earth's air which causes afflictions, hypersensitivities and downfall to the living animals in natural framework [1–10]. The atmosphere is a vaporous structure fundamental to help life on the planet earth. Smithy Institute announced world's worst polluted places; it records distinctive indoor and air quality issues. A toxic substance can be of trademark or man-made. Air harms are major released from motor vehicle exhaust, and assembling plants conveyed enormous proportion of flotsam and jetsam, carbon monoxide and sulfur dioxide independently. Exactly, when nature is outfitted with sensor devices, microcontroller and distinctive programming applications, it transforms into a self-making sure about and self-watching condition and it is in like manner called as sharp condition. In a sharp space, if there is a vehicle exhaust, the reprimand message is sent to customer and Air Pollution office alerts normally. The effects in light of the lacking consuming of fuel from the vehicle can be checked and obliged by means of air sullying watching structure. Utilization of sagacious condition targets using of embedded knowledge into nature which makes the earth shrewd with objectives. A human need demands different sorts of watching structures these depend upon the kind of data gathered by the sensor contraptions [11–14]. Sharp condition is of two sorts: event detection-based and spatial process estimation. Sensor devices are placed in the vehicle to accumulate the data to foresee the lead of toxic substances. Point behind this examination is to design and execute a capable checking structure which screens remotely using web. The data amassed from the sensors are taken care of in the cloud and evaluated using web program. If the assessed regard crosses beyond what many would consider possible, by then we close the proximity of the defilement in the particular condition. In this paper, by using remote embedded preparing structure, we in like manner present a slanting delayed consequence of assembled or identified data with respect to the normal or decided extents of explicit parameters. Similarly, many software tools have played a major role in the analysis of the applications of electrical and reported [15–17].

2 Existing System Model

The activity of BLDC motor is comprised of three sections, namely (i) modeling, (ii) controller and (iii) the transcendent embedded device for watching CO levels in the vehicle radiation to make the earth canny or instinctive with the articles through

remote correspondence. The proposed model is shown up in Fig. 2 which is progressively flexible and distributive in nature to screen the environmental parameters. The proposed model is shown up in Fig. 2 which is progressively flexible and distributive in nature to screen the environmental parameters. The proposed building is discussed in a four-level model with the components of each individual modules made for air tainting checking. Level 1 is nature, sensor devices in Level 2, sensor data acquiring and dynamic in Level 3 and astute condition in Level 4. The existing designing is shown up in Fig. 1; Level 1 gives information about the parameters under the district which is to be checked for noise and air defilement control. Level 2 courses of action with the sensor devices with proper characteristics, features, and all of these sensor contraptions are worked and controlled subject to its affectability similarly as the extent of recognizing. In Level 2 and Level 3, basic distinguishing and controlling moves will be made depending on the conditions, for example, fixing the edge regard, periodicity of identifying, messages (alert or chime or LED, etc.). Considering the data assessments acted in Level 2 and Level 3 and moreover from the past experiences, the parameter edge regards during essential conditions or conventional working conditions are settled. Level 3 depicts about the data getting from sensor devices and moreover consolidates the dynamic, which demonstrate the condition that the data are addressing on which parameter. In the proposed model, Level 4 game plans with the sagacious condition which suggests it will perceive the assortments in the sensor data and fix the edge regard dependent upon the recognized level of CO.

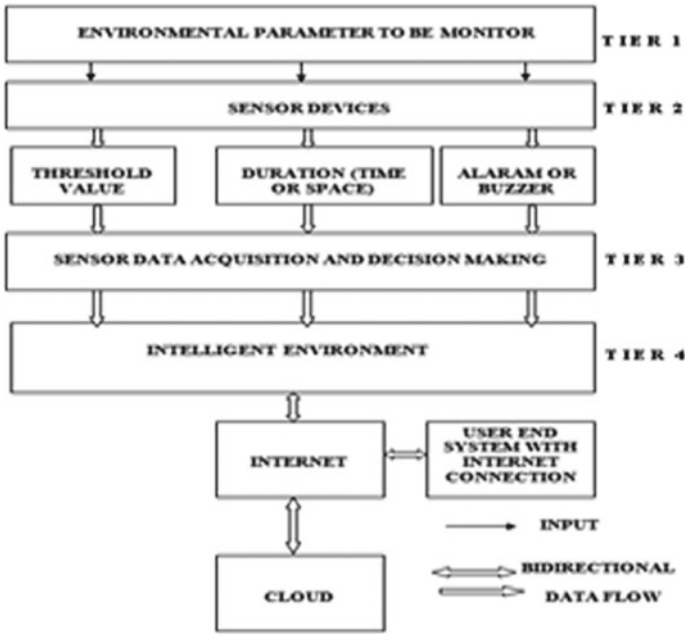


Fig. 1 Existing model

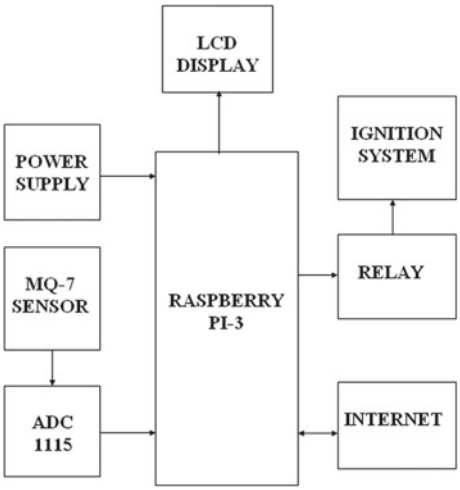


Fig. 2 Proposed model

3 Literature Survey

The present strategy uses the shrewd condition which perceives the assortments in the sensor data and fixes the edge regard dependent upon the recognized level of CO. The level distinguished data which were got together with Google spreadsheets and shows an example of the recognized parameters concerning the foreordained characteristics. The end customers can examine the data using PDAs, PCs. In the existing model, Level 4 courses of action with the cunning condition. It will recognize the assortments in the sensor data and fix the edge regard dependent upon the perceived level of CO. In this level, distinguished data will be taken care of and set aside in the cloud for instance in to the Google spreadsheets, and moreover, it will show an example of the recognized parameters concerning the predefined values. The current embedded contraption is for checking racket and CO levels in the atmosphere to make the earth sharp or savvy with the things through remote correspondence. The present model is shown up in Fig. 1 which is progressively adaptable and distributive in nature to screen the biological parameters. The present plan is discussed in a four-level model with the components of each individual modules created for air pollution watching. The enduring model includes four-levels. Level 1 is the earth, sensor contraptions in Level 2, sensor data acquisition and dynamic in Level 3 and shrewd condition in Level 4 for watching nature.

4 Proposed CO Monitoring Model

The objective of this paper is to give human prosperity and control the air defilement by using IoT. The structure decreases the CO level made by the vehicle, so impacts

on the people can be dodged. Table 1 shows the different effects of CO hurting on individuals. This endeavor is used for consistent seeing of CO made by the lacking devouring of fuel. It is done with the help of Raspberry Pi-3, CO sensor LCD show and hand-off. CO sensor should be acquainted at close with (10 cm) the channel of exhauster. The contraption has three standard potential circumstances. These are whether the CO center around the district is in safe condition, notice or danger state. The LCD will print those characteristics unfailingly that are surveyed of what might be finished. The fumes framework found in vehicles joins carbon monoxide and oxygen to shape less harmful carbon dioxide (CO_2). Fumes framework lessens the high concentration in the ventilation framework (normally, 30,000 ppm in outside and 400 ppm in inside the vehicle) to low obsession (customarily, underneath 1000 ppm after the fumes framework in outside and 100 ppm in inside the vehicle). If the center is under 1000 ppm, the LCD will print the 'spare' status. In case the center is inside the extent of 10,000 ppm in outside and 200 ppm in inside the vehicle, the LCD will print the 'cautioning' status and lit the exhortation LED. Else, the obsession is more significant than 30,000 ppm that the LCD will print the 'peril' status, by then the hazard LED will lit up if after a couple of times if vehicle driver could not think less about the CO level, the message will be sent normally to sullying control load up and show the conveying of the CO level against the vehicle to make the verification of over defilement.

Table 1 Physiological effects of CO in closed area

Parts per million	Time of exposure	Response
50	-	Threshold limit non-toxic
100	Several hour	No symptoms
200	2–3 h	Headache
400	1–2 h	Headache and nausea
800	45 min	Headache, dizziness and nausea
800	2 h	Unconsciousness
1600	20 min	Headache, dizziness and nausea
1600	2 h	Possible death
3200	5–10 min	Headache and dizziness
3200	10–15 min	Unconsciousness and possible death
6400	1–2 min	Headache and dizziness
6400	0 min	Unconsciousness and possible death
12,800	Immediate	Unconsciousness
12,800	1–3 min	Danger of death

Table 2 Physiological effects of vehicle CO emission in open air

Parts per million	Time of exposure	Response
Below 1000	-	Threshold limit no symptoms
10,000	9–12 h	Possible headache
30,000	1–3 h	Unconsciousness and possible death
Above 45,000	1–3 min	Danger of death

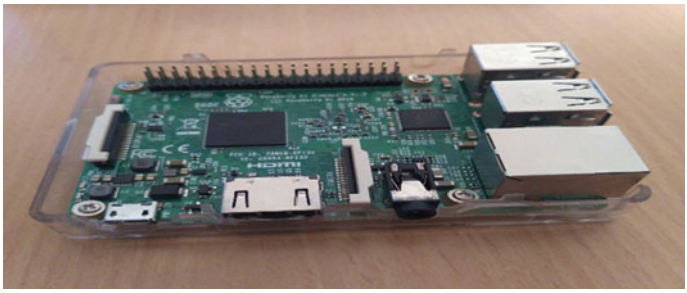


Fig. 3 Raspberry Pi-3 (Model B)

Table 1 shows that the physiological impacts of carbon monoxide present inside the vehicle. Here, we think about the three territories for CO recognition. There are 100, 200 and 400 ppm. If the CO level surpasses the 400 ppm methods, the people endured by threat sicknesses like wooziness. So, we set the most extreme recognition level as 400 ppm (Table 2).

At the point when the client clears the fine, the pollution specialists send the OTP to the transfer by utilizing IoT. So, the hand-off is working, and afterward, vehicle can be begun. The proposed model is appeared in Fig. 2: This model comprises of CO sensor, Raspberry Pi-3, LCD show and hand-off. It is the most recent adaptation of Raspberry Pi-3 Model B with 40-pin GPIO and force charger.

Raspberry Pi is having the inbuilt web associating port. It goes about as the little PC. Model B is the propelled form in Fig. 3, having the constructed Wi-Fi module. This Model B is further developed in all application. Here, the Raspberry Pi-3 goes about as processor. This associates all the parts in the proposed framework.

4.1 CO Sensor

The CO sensor is utilized to quantify the CO delivered by the vehicle. The one adaptation of CO sensor is MQ-7 (Fig. 4).

Assume the deliberate CO level is surpass the limit esteem it will naturally showed on the LCD show. In case after a few times the driver could not care less about the

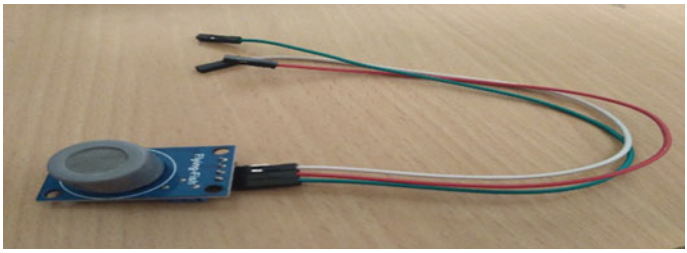


Fig. 4 MQ-7 sensor

CO level methods, the message will be consequently sent to pollution specialists and showed the creating of the CO level.

4.2 Relay

A hand-off is an electrically worked switch. Transfers associated together are utilized as an electromagnet to precisely work the switch and give electrical isolation between two circuits. This eight-channel 5 V transfer has both optical and attractive segregations, which protect the contributions from electrical deficiencies, work well with 3.3 and 5 V rationale (Raspberry Pi.), requiring minimal flow drive. For this situation, the workplace will document body of evidence against the vehicle to deliver the proof of over contamination. Presently, the hand-off is opened and afterward bolts the vehicle. At the point when the client clears the fine, the pollution specialists send the OTP to the hand-off by utilizing IoT. So, the hand-off is working, and afterward, vehicle can be begun.

4.3 Internet of Things (IoT)

This framework is additionally checked utilizing IoT, the web of things. At whatever point the parameters cross the limits, the qualities are refreshed. These refreshed qualities can be seen anyplace and whenever by opening the connection given through web. The contributions from sensors given are interfaced with IoT and made accessible online constantly, so any individual who has the connection can see the state of the parameters. An IP address is made so as to get to the information. A Web site page shows up when the connection is locked.

Job of IoT: The sensors yields are interfaced with web of things. A page is made which contains the readings or perceptions of sensors. The yield of sensors is constantly refreshed and can be found in site page at Air Pollution office. The Web site page is facilitated on the web, and a connection is given through which we can get to the page and view the sensors yield.

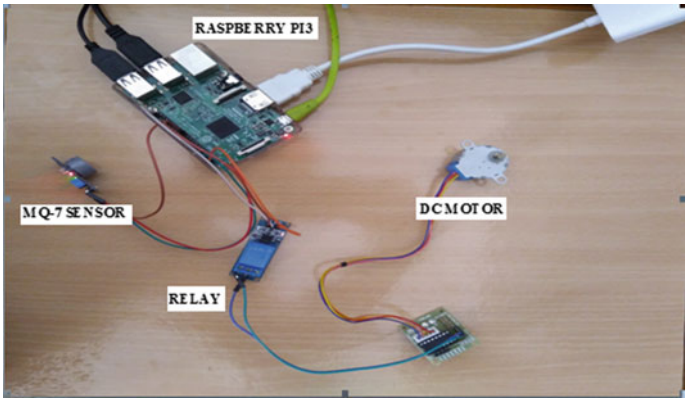


Fig. 5 CO monitoring and control in vehicle by using Raspberry Pi-3 system

5 Implementation

In this execution model, we utilized Raspberry Pi-3 Model B with inbuilt web association port. Raspberry Pi-3 comprises of 40 general purpose input output (GPIO) pins. The execution model is appeared in Fig. 5. This model shows implanted framework with its segments utilized for perusing and putting away detected data. MQ-7 sensors are associated with Raspberry Pi-3 for observing the CO level. MQ-7 sensor has four pins to be specific analog output (AO), digital output (DO), GND and VCC. The DO stick is associated with the contribution of Raspberry Pi-3 of every 13 pin, and the yield is got at Pin 20. The sensor information can be moved to contamination specialists by utilizing web association. Also, the relay is associated with the Raspberry Pi-3 for (ON/OFF) controlling the vehicle.

At the point when the CO level surpasses the most extreme range, the hand-off will disengage the association between start framework and engine. So, the vehicle is halted where the vehicle client at long last off the Key. At the point when the contamination specialists send the OTP to transfer by utilizing IoT, at that point, the hand-off will build up the association.

After the culmination of information detecting, it will be prepared through IoT. The conclusive outcome is appeared in Fig. 6. The points of interest of this task are to dodge the age of CO because of the vehicles that are subserviced. So, this undertaking makes people to administrate their vehicle appropriately.

6 Conclusion

The proposed framework which is planned shows the yield of detecting the carbon monoxide gas and afterward showed through LCD. Certain moves are made by

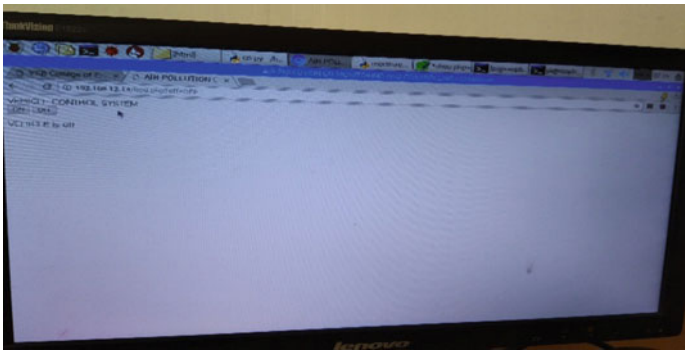


Fig. 6 Web page result

Pollution Control Board when the client does not find a way to control the outflows of CO in vehicle much after the two notice messages are shown on LCD. The third time sensor yield is pushed to cloud and can be seen by Pollution Control Board through web and lock the vehicle by utilizing IoT. This is a strong framework which is valuable for people as a result of the expanding contamination because of increment in vehicles. This framework is easy to understand, and cost of the item is reasonable. The consequences of the task are precise. This model demonstrates a productive contamination control in vehicle through consistent checking.

References

1. Goth A (2016) Implementation of an efficient noise and air pollution monitoring system using Internet of Things (IOT). ISSN 2319 5940 Int J Adv Res Comput Commun Eng ISO 3297:2007 5(7)
2. Culler D, Strain D, Srivastava M (2004) Overview of sensor networks. IEEE Comput
3. Zucatto FL, Biscassi CA (2007) ZigBee for building control wireless sensor networks. IEEE
4. James JQ, Li VOK, Lam AYS (2007) Sensor deployment for air pollution monitoring using public transportation system. 1024–1031
5. Kwon J-W, Park Y-M, Kook S-J, Kim H (2007) Design of air pollution monitoring system using ZigBee networks for ubiquitous-city. In: Proceedings of the international conference on convergence information technology, pp 1024–1031
6. John Wiley and Sons Ltd, The Atrium, Southern Gate, Chi Chester, West Sussex, England, 2005 (2007) An atmosphere environment monitor system based on wireless sensor network Journal 26: 47–51
7. Gutierrez JA (2004) On the use of IEEE 802.15.4 to enable wireless sensor networks in building automation. IEEE
8. Karl H, Willing A Protocols and architectures for wireless sensor networks
9. Mainwaring A, Culler D, Plaster J, Szewczyk R, Anderson J (1998) Wireless sensor networks for habitat monitoring. In: de Bert M (ed) Facing the air pollution agenda for the 21st century, in air pollution, priority issues and policy, Netherland. Schneider, Elsevier Science B.V. pp 3–8
10. Martinez K, Hart JK, On R (2004) Environmental sensor networks. IEEE Comput J 37(8):50–56

11. Sethy PK, Behera SK, Kannan N, Narayanan S, Pandey C Smart paddy field monitoring system using deep learning and IoT. Concurrent Eng: Res Appl 1–9. https://doi.org/10.1177/1063293X21988944
12. Van Edmond ND, Schneider T (1998) Historical perspective and future outlook. In: Air pollution in the 21st century, priority issues and policy. Elsevier, Netherland, pp 35–46
13. Masilamani R, Sureshkumaar G, Kannan N et al (2021) PIC microcontroller based development of air quality improvement system for automobiles. Mater Today: Proc. https://doi.org/10.1016/j.matpr.2020.09.541
14. Zhang Q, Yang X, Zhou Y, Wang L, Guo X (2007) A wireless solution for greenhouse monitoring and control system based on ZigBee technology. 8:1584–1587
15. Etier I, Murugan CA, Kannan N, Venkatesan G (2020) Measurement of secure power meter with smart IOT applications, J Green Eng 10(12):12961–12972
16. http://www.epa.gov/OCEPAterms/pterms.html. EPA Website
17. http://en.wikipedia.org/wiki/Air_pollution. Wikipedia

TWEERIFY: A Web-Based Sentiment Analysis System Using Rule and Deep Learning Techniques



Oluwatobi Noah Akande, Enemuo Stephen Nnaemeka, Oluwakemi Christiana Abikoye, Hakeem Babalola Akande, Abdullateef Balogun, and Joyce Ayoola

Abstract Social media platforms have ceased to be a platform for interaction and entertainment alone, and they have become a platform where citizens express their opinions about issues that affect them. In recent years, it has become a powerful platform where elections are won and lost. Therefore, organizations and governments are increasingly interested in citizen's views expressed on social media platforms. This research presents a novel approach to carry out aspect-level sentiment analysis of users' tweets using rule and convolutional neural network (CNN)-based deep learning technique. The rule-based technique was used to detect and extract sentiments from preprocessed tweets, while the CNN-based deep learning technique was employed for the sentiment polarity classification. A total of 26,378 tweets collected using "security" and "Nigeria" keywords were used to test the proposed model. The proposed model outperformed existing state-of-the-arts GloVe and word2vec models with an accuracy of 82.31%, recall value of 82.21%, precision value of 82.75% and $F1$ score of 81.89%. The better performance of the proposed techniques could be as a result of the rule-based techniques that was introduced to capture sentiments expressed in slangs or informal languages which GloVe and word2vec have not been designed to capture.

O. N. Akande (✉) · J. Ayoola
Computer Science Department, Landmark University, Omu-Aran, Kwara State, Nigeria
e-mail: akande.noah@lmu.edu.ng

O. N. Akande · E. S. Nnaemeka
Landmark University SDG 09: Industry, Innovation and Infrastructure, Omu-Aran, Kwara State, Nigeria

O. C. Abikoye
Computer Science Department, University of Ilorin, Ilorin, Kwara State, Nigeria

H. B. Akande · A. Balogun
Department of Telecommunication, University of Ilorin, Ilorin, Nigeria

A. Balogun
Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, 32610 Bandar Seri Iskandar, Perak, Malaysia

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_7

Keywords Data security · Aspects extraction · Sentiment analysis · Deep learning

1 Introduction

The need to capture the opinion of citizens or customers expressed in response to products or government policies is increasing on a daily basis. It serves as a feedback mechanism that can be used to determine how well a product is accepted in the market or how bad a government policy affects her citizens. Literarily, a sentiment is a general feeling, opinion, disposition or perception of a situation or thing that can be expressed via sign languages, written words or spoken words. Sentiment analysis or opinion mining is a field of research that detects, extracts and analyzes peoples' sentiments about a product, service or governments' policies. It has grown to become an active area of research in natural language processing (NLP) with application areas in intelligent recommender systems, data mining, Web mining, text mining, information retrieval, summarization, etc. [1, 2]. Its use has cut across different field of study such as computer science, political science, marketing and health sciences [3–6] communications. Furthermore, the increasing penetration and use of the internet coupled with the wide acceptability of social media platforms have also contributed to the rapid development of this research field. Nowadays, individuals and organizations employ the content of these media for decision making. Companies use consumers' opinion or feedback from existing products to improve new products, while consumers also prefer to scan through product reviews in order to use the opinion of people who have purchased a product to make informed decision whether they should go ahead to buy the same product or not. Therefore, organizations may not necessarily need to conduct surveys or opinion pool to harvest customers' feedback due to the abundance of such information on products' Web sites. Opinion could be categorized as regular or comparative opinion [7, 8]. Regular opinion provides a direct or indirect sentiment about an entity. For instance, "the drug is powerful" is a direct opinion, while "my headache increased after using the drug" is an indirect and negative opinion about the entity "drug". On the contrary, comparative opinion expresses a sentiment about an entity with respect to another entity. For instance, "I prefer Panadol to Paracetamol for the treatment of headache" is a comparative opinion that gives a positive opinion about the usage of Panadol for headache treatment. However, extracting and summarizing these huge volumes of opinions remain a hectic task especially from social media networks.

Most tweets are not necessarily written in formal English language; therefore, the use of acronyms, symbols, emoticons and abbreviations which varies from one country to another is acceptable. According to authors in [8], sentiment analysis can be carried out in three levels: document, sentence [9] and aspect-based. Document-level sentiment analysis examines the entire opinion document and then posits if the document expresses a negative or positive sentiment. This shows that document-level sentiment analysis gives an overall sentiment polarity associated with one entity. However, in sentence level, the opinion document is broken down into sentences,

and the sentiments of each sentence is then extracted and categorized into negative or positive. This shows that sentence level gives a more detailed sentiment polarity than document level. However, aspect-level sentiment analysis attempts to carry out a more comprehensive and in-depth examination of expressed thoughts hidden in each sentence. Aspect level looks beyond the language constructs such as documents, sentences, paragraphs, phrases or clauses; it directly examines the opinion itself. Furthermore, aspect-level sentiment analysis involves two subtasks: aspect detection and sentiment classification [2]. While aspect detection focused on the extraction of sentiments, sentiment classification majorly focused on categorizing the extracted sentiments into respective polarity (i.e., positive, neutral or negative); this can be achieved using knowledge-based techniques (lexicon approach), machine learning techniques (statistical approach) and hybrid methods [10]. Therefore, this study explores the use of deep learning and rule-based machine learning techniques for aspect-based sentiment detection, extraction and classification. With emphasis on Twitter data, the rule-based technique was used to detect and extract sentiments from preprocessed tweets, while the CNN-based deep learning technique was employed for sentiment polarity classification.

2 Methodology

To ensure that no aspect of a tweet is missed due to the sensitive nature of tweet data, rule-based and deep learning techniques were employed for the aspect detection, extraction and classification. Rule-based technique was employed for the sentiment detection and extraction, while deep learning technique was employed for the sentiment classification. The stages employed include tweet data collection and preprocessing, model development, model evaluation and model data visualization.

2.1 Tweet Data Collection and Preprocessing

Two categories of tweet data were used in this study: the training and testing data. The training data are 1.6 million annotated security-related tweets made available during Kaggle Sentiment140 event, while the testing data are 26,378 tweets with keyword “security” and “Nigeria” that reflects the state of security in Nigeria. The tweet data used for testing were primarily collected with Twitter API; however, it was discovered that the Twitter API could not collect tweets that are older than a week, some were incomplete, truncated, repeated or redundant. Therefore, a Web scraping script was incorporated to supplement data gotten via the Twitter API. The Web scraping technique was used to extract tweets from the Twitter Web site using the specified keywords. Furthermore, the tweet data contain several URL links, @mentions, usernames, hashtags, numbers, acronyms, unnecessary symbols, tags,

tabs, tweet_id, user details, language attributes, emoticons, whitespaces, media entities like images, videos, etc. All these are unwanted features which were removed during the preprocessing stage. This was achieved with the help of a script that could recognize and identify regular expressions and/or patterns matching elements that are modeled to be unnecessary and need to be removed. With the help of an inbuilt Python RegEx library, the pseudocode below was implemented for the preprocessing task:

```
Input: Tweet sentences
Output: Preprocessed Tweets
for each sentence in tweet_sentences do
    a. sentence = RegEx.remove(r' http\S+', sentence)
    b. sentence = RegEx.remove(r' pic.\S+', sentence)
    c. sentence = RegEx.remove(r' @\s\w+', sentence)
    d. sentence = RegEx.remove(r' < [^>] +>', sentence)
    e. sentence = RegEx.remove(r' [^a-zA-Z] ', sentence)
    f. sentence = RegEx.remove(r' \s+[a-zA-z] \s+ ', sentence)
    g. sentence = RegEx.remove(r' \s+ ', sentence)
    h. sentence = sentence.trim()
```

Where + represents strings containing a pattern appearing one or more times, \S represents strings containing no whitespace character, \w represents matches where the string contains any word characters (characters from a to Z, digits from 0 to 9, and the underscore character), and \s represents matches where a string contains a whitespace character. Also, in step b, URLs, i.e., strings beginning with pic.twitter, were removed, in step c, @mentions in the tweets are removed, in step d, contents in HTML tags were removed, in step e, elements that are not alphabetical, i.e., symbols, punctuations, etc., were removed, in step f, single characters were removed, in step g, multiple concurrent spaces were removed, while in step h, spaces and white spaces were removed from the input tweet data.

2.2 Model Development

The framework of the developed aspect-based sentiment extraction and classification model using rule-based and deep learning techniques is presented in Fig. 1. It has two modules:

(i) Aspect identification and extraction module
(ii) Sentiment polarity identification module.

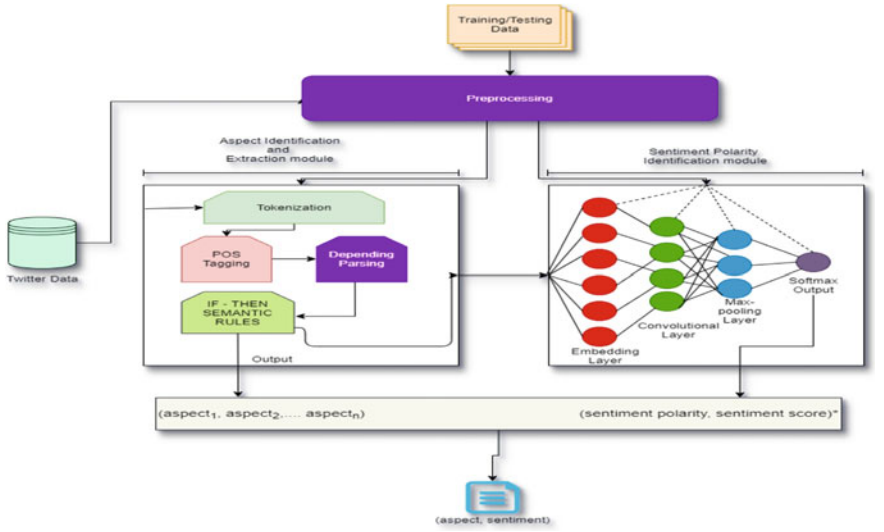


Fig. 1 Research framework

2.2.1 Aspect Identification and Extraction Module

The aspect identification and extraction module consist of components for identifying and extracting aspect terms from sentences. Rule-based technique that uses natural language processing rules was used to extract relevant aspects from the tweets. SentiWordNet which is a sentiment lexicon in which words are assigned numeric sentiment values was used as the opinion word lexicon. Of the 26,378 tweets considered, 51,949 total words was worked with after words having zero sentimental values have been removed. SpaCy Python library was used to implement the aspect extraction. This involves:

(i) Tokenization: This breaks sentences into smaller elements called tokens for easy analysis.
(ii) Part-Of-Speech (POS) Tagging: This step identifies and classifies tokens on the part of speech they belong to.
(iii) Dependency Parsing: A variety of aspects dependencies are collected after the part-of-speech tagging and defined using syntactic tokens.
(iv) Implementing IF–THEN rules: Rules in the form "IF–THEN" are used to extract aspects based on identifiable semantic relationships between tokens. These rules are inferred from studying sentence structure, grammatical and lexical syntax and semantics of sentences. The pseudocode of the if…then rules employed goes thus:

```
for each sentence in tweets do
   POS_Tag(sentence)
   Dep_Parse(sentence)
   for each word in sentence do
      If opinion word in sentence then
         If POS_Tag_of(word) = NOUN then
If Dep_relationship_between(opinion word, word) =
ADJECTIVAL MODIFIER or ADVERBIAL MODIFIER then
                Aspects.add(word)
            End if          End if          End if
      If Dep_relationship_of(word) = CONJUCTION in sentence then
         If word.neighbor.left in Aspects then
            If POS_Tag(word.neighbor.right) = NOUN then
               Aspects.add(word.neighbor.right)
            End if          End if
         sentence_split_leftt = sentence.split(word.index).left
         sentence_split_right = sentence.split(word.index).right
      if Dep_relationship_of(word) = NOMINAL SUBJECT in sentence_split_left
         or sentence_split_right then
            if POS_Tag_of(word) = NOUN then
               if word mot in Aspects then Aspects.add(word)
                        End if    End if
     If Dep_relationship_to(word) = DIRECT OBJECT in sentence_split_left or
   sentence_split_right then Aspects.add(Dep_relationship(DIRECT OBJECT).word)
          End if
       End if
       If POS_Tag(word) = VERB then
   If Dep_relationship_to(word) = DIRECT OBJECT and
   POS_Tag(Dep_relationship_to(word))  = NOUN then Aspects.add(word)
    End if
       End if
        If POS_Tag(word) = VERB then If Dep_relationship_to(word) = DIRECT
   OBJECT and POS_Tag(Dep_relationship_to(word)) = NOUN then As-
   pects.add(word)
       End if
            End if
       If word in Aspects and Dep_relationship_to(word) = NOUN
          COMPOUND then Aspects.add(word)
       End if
          End for
               End for
```

2.2.2 Sentiment Polarity Identification and Classification Module

A six-layer convolutional neural network (CNN) was employed for the sentiment polarity identification and classification. The network consists of an embedding

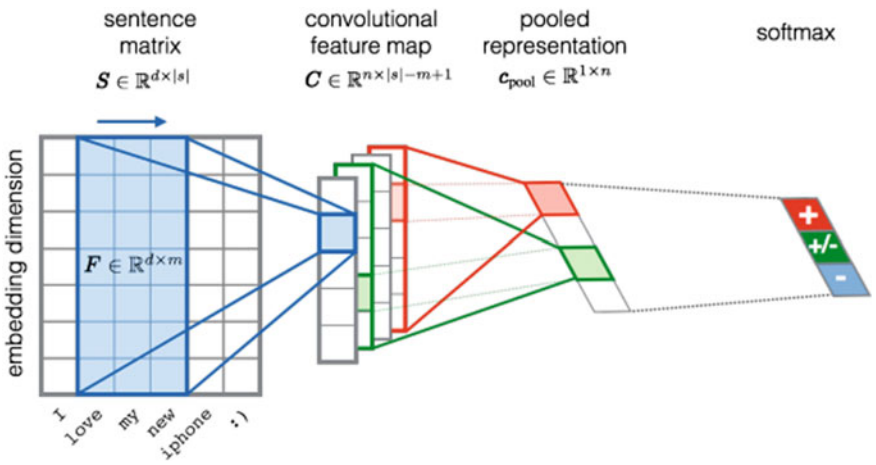


Fig. 2 CNN-based sentiment polarity identification and classification module

layer, two convolutional layers, two max-pooling layers and a fully linked layer with Softmax output. This is illustrated in Fig. 2:

2.3 Performance Evaluation

The performance of the proposed rule and deep learning-based model for sentiment analysis was evaluated using accuracy, precision, recall and *F*-score such that:

(i) Accuracy measures the proportion of correctly labeled tweets in comparison with the total number of tweets used for the testing. Accuracy is more efficient when the distribution of the class is fairly equal. Equation (1) was used for the accuracy.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positive} + \text{False Positive} + \text{True Negatives} + \text{False Negatives}} \quad (1)$$

(ii) Precision is also called the positive predictive value. It measures the percentage of correctly classified tweets that are actually correct. Equation (2) was used for calculating the precision:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

(iii) Recall is also known as sensitivity. It measures the completeness of the techniques used in categorizing the polarity of a tweet. It was calculated such that:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

(iv) F-Score: This is an extrapolation metric that combines precision and recall as the harmonic mean of both. It gives equal weight to precision and recall. It is defined mathematically as:

$$F = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

3 Results and Discussion

The results were discussed under the following subsections:

3.1 Model Evaluation

The proposed rule-based and machine learning model was compared with the state-of-the-art word2vec and GloVe word embedding models. During the training stage, the proposed model with its indigenously trained word embedding outperforms the baseline pre-trained embedding models in all performance metrics. The training results are presented in Table 1. However, during the testing stage as presented in Table 2, the proposed model outperformed the existing models in terms of the accuracy, recall and F-score.

Table 1 Training performance of the proposed model

Performance metric/model	Accuracy (%)	Recall (%)	Precision (%)	F-score (%)
Word2vec model	84.82	84.20	85.20	84.28
GloVe model	83.62	80.35	85.91	82.55
Proposed model	87.91	87.55	88.15	87.49

Table 2 Testing performance of the proposed model

Performance metric/model	Accuracy (%)	Recall (%)	Precision (%)	F-score (%)
Word2vec model	82.22	81.69	82.73	81.72
GloVe model	82.18	78.92	84.57	81.14
Proposed model	82.31	82.21	82.75	81.89

Table 3 Aspect characteristics of extracted tweets

Item	Value
Unique appearing aspects	79,318
Total number of aspects appearing more than once	17,147
Highest appearing aspect	Nigeria
Times appeared	15,702
Maximum polarity associated with an aspect	Negative
Polarity breakdown for aspect	Negative: 7547 neutral:4566 positive: 1549
Total number of aspects appearing once	62,171
Mean aspect occurrence per tweet	7.2092
Tweets with aspects less than mean aspect appearance	11,156
Number of tweets with one aspect	385
Polarity breakdown for tweets with one aspect	Negative: 167, neutral: 138, positive: 80

3.2 Characteristics of Aspects Extracted

As mentioned in the previous section, 26,378 tweets used to test the proposed model were gathered using a Web scraping script and Twitter API. Table 3 gives an overview of the aspects extracted from the tweets captured.

3.3 Sentiment Polarity Module

The sentiment polarity identification and extraction model were used on the collected data. Although the proposed model showed impressive results in training and validation, the model still showed a few noticeable discrepancies in classifying the polarity of tweets. It is therefore deduced that collected tweets do not share a resemblance in form and structure to the training data; i.e., the training data still lacked words that seem to appear consistently in the collected tweets. Examples of some other discrepancies include tweets containing substrings or appearing entirely in Yoruba, Pidgin or Hausa (which are our native languages) and unnatural abbreviations (e.g., collect to mean collect) which is common in this part of the world. Hence, both models trained with pre-trained word embedding were used to supplement the proposed model. It should be noted that the models trained with pre-trained word embedding

have no difference in process architecture with that of the proposed model. The only difference is the embedding used. In the proposed model, the embedding is trained from the training data, which have 272,944 unique words only. On applying all three models, the final polarity score of a tweet is a normalized value of all three model values. This normalization involves calculating a mean of the three values, calculating the median of the three and finding the mean of these two values. If the final polarity score of the model is denoted by f and the models' score denoted by p, w, g, respectively, then the final polarity is computed using Eq. (5) such that:

$$f = \frac{\text{mean}(p, w, g) + \text{median}(p, w, g)}{2} \tag{5}$$

This deals with outliers, considers the score of all three models and gives a value close to a range (± 0.046) of values common to two of the models. For instance, if the proposed model scores a tweet as 0.565, the word2vec model scores 0.264, and the GloVe scores 0.248. The polarity of the first model is neutral, and the majority polarity of the classified tweet would be negative (using the aforementioned scale for classifying polarity) as the value common to two models is negative. The normalized value is 0.3115 which still falls on negative but closer to neutral, considering the score of the first model. Using this method, Fig. 3 shows the distribution of polarity over the collected tweets.

Results obtained revealed that there are more negatively classified tweets than other categories. The total number of neutral tweets came as a surprise. However, when the testing tweets were examined, it was observed that the tweets which were categorized as neutral consist of statements, suggestions and questions.

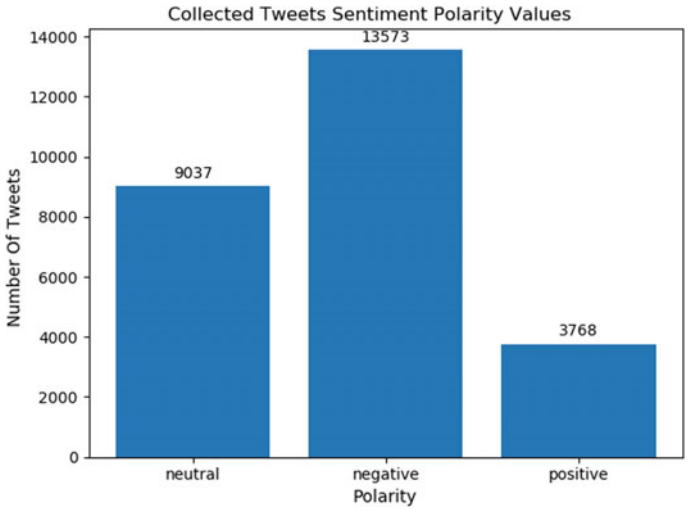


Fig. 3 Value distribution of sentiment polarity over the collected tweets

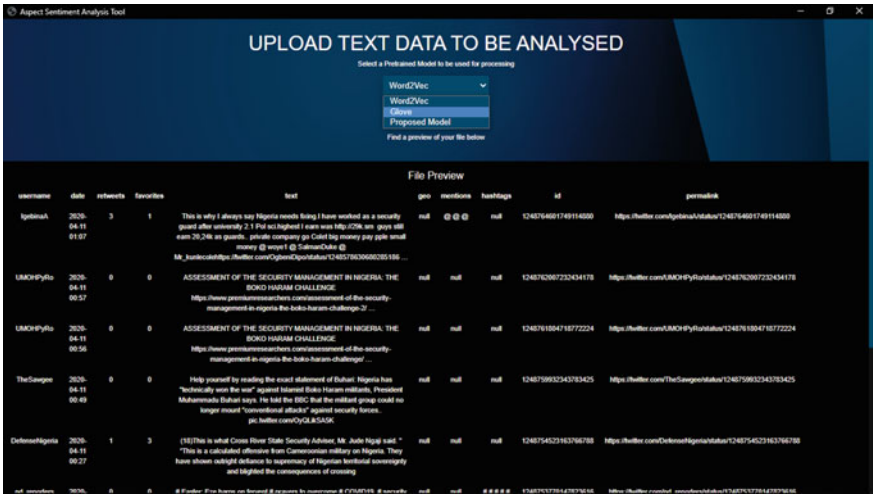


Fig. 4 File upload preview window

3.4 Web-Based Graphical User Interface

For easy prediction, a Web-based graphic user interface (GUI) shown in Fig. 4 was developed. This makes it easier for users to interact with the sentiment prediction system. Once the tweets to be analyzed have been extracted into a spreadsheet file, the file can easily be uploaded via the GUI. This displays an overview of the file uploaded in terms of the username, date, number of retweets, the actual text, mentions, hash tags, tweet id and permalinks. One important feature of this interface is the section that allows the user to choose the desired pre-trained model to be used for processing. This could either be the GloVec, word2vec or the proposed model. Once the preferred pre-trained model has been selected, the results will be presented in the interface shown in Fig. 5. The classification results are presented in terms of the aspects extracted from each tweet with their respective polarity, insights summary and the performance evaluation results in terms of accuracy, precision, recall and F-score.

4 Conclusion and Future Work

This research has extensively discussed a novel technique for collecting and analyzing sentiments expressed in tweets using rule and CNN-based deep learning technique. The rule-based technique was used to detect and extract sentiments from the processed tweets, while the CNN-based deep learning technique was employed to identify and classify sentiment polarity of the tweets. The data used to test the proposed model are 26,378 security-related tweets that were collected using

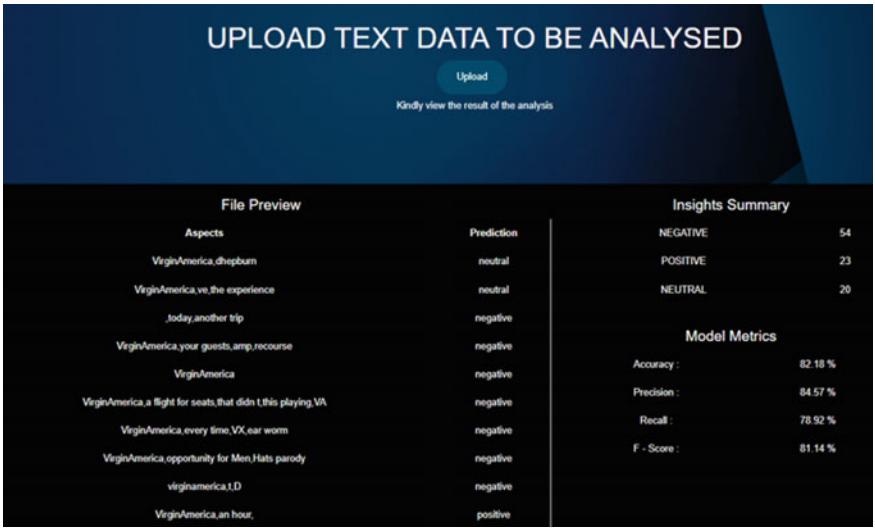


Fig. 5 Classification results preview

keywords: security and Nigeria. The total number of negative aspects extracted from the tweets was 7547, neutral aspects were 4566, while positive aspects were 1549. The proposed model yielded a better performance than the state-of-the-art word2vec and GloVe models in terms of accuracy, recall, precision and F-score. Furthermore, a Web-based graphic user interface was designed to make the sentiment analysis task easier. It enables users to upload tweets that have been saved in spreadsheet formats. Afterward, the preferred word embedding model will be selected, and the semantic analysis of the uploaded tweets in terms of the statistics of the positive, neutral and negative polarities, accuracy, recall, precision and $F1$ score values will be displayed. A real-time version of the semantic analysis system with more features will be explored in subsequent research.

Acknowledgements Authors appreciate Landmark University Centre for Research and Development, Landmark University, Omu-Aran, Nigeria, for sponsoring the publication of this article.

References

1. Zhang L, Wang S, Liu B (2018) Deep learning for sentiment analysis : a survey. https://doi.org/10.1002/widm.1253
2. Yang C, Zhang H, Jiang B, Li K (2019) Aspect-based sentiment analysis with alternating coattention networks. Inf Process Manage 56(3):463–478. https://doi.org/10.1016/j.ipm.2018.12.004
3. Akande NO, Abikoye CO, Adebiyi MO, Kayode AA, Adegun AA, Ogundokun RO (2019) Electronic medical information encryption using modified blowfish algorithm. In: International

conference on computational science and its applications. Springer, Cham, pp 166–179. https://doi.org/10.1007/978-3-030-24308-1_14
4. Abikoye OC, Haruna AD, Abubakar A, Akande NO, Asani EO (2019) Modified advanced encryption standard algorithm for information security. Symmetry 11:1–17. https://doi.org/10.3390/sym11121484
5. Oladipupo ET, Christianah AO, Oluwatobi AN, Aderonke KA, Kehinde AJ (2020) Comparative study of two divide and conquer sorting algorithms: quicksort and mergesort. Procedia Comput Sci 171:2532–2540. https://doi.org/10.1016/j.procs.2020.04.274
6. Oluwatobi AN, Ayeni TP, Arulogun TO, Ariyo AA, Aderonke KA (2020) Exploring the use of biometric smart cards for voters' accreditation: a case study of Nigerian electoral process. Int J Adv Sci Eng Inf Technol 10(1):80–89. https://dx.doi.org/10.18517/ijaseit.10.1.8459
7. Jindal N, Liu B (2006) Mining comparative sentences and relations. In: Proceedings of national conference on artificial intelligence Aaai
8. Liu B (2012) Sentiment analysis and opinion mining. https://doi.org/10.2200/S00416ED1V01Y201204HLT016
9. Kim Y (2011) Convolutional neural networks for sentence classification. ArXiv: 1408.5882
10. Li W, Qi F, Tang M, Yu Z (2020) Bidirectional LSTM with self-attention mechanism and multi-channel. Neurocomputing 387(28):63–77. https://doi.org/10.1016/j.neucom.2020.01.006

Machine Learning Based Fake News Detection on Covid-19 Tweets Data



Vishesh Mehta and Ram Krishn Mishra

Abstract The importance of social media has seen a huge growth in last few years as it lets folks from any part of the world remain in contact. Due to COVID-19 epidemic, social media is used extensively and has turned to be more pertinent than the past years. Alongside, fake news dissemination has revived and also dissemination of tweets has taken the attention. For the present study, we used various machine learning models to detect the fake news on COVID-19 related tweets. Due to millions of active social media users, identifying the fake news has become a crucial task. The models we applied are Gradient Boosting Classifier (GBC), Logistic Regression (LR), Random Forest Classifier (RFC), Decision Tree Classification (DT). All these models detect if the tweet relating to COVID-19 is "Fake News" or "Not a Fake News". Hence we conclude that Logistic Regression is best among all other models.

Keywords COVID-19 · Machine learning · TF-IDF · Fake news

1 Introduction

Fake news portrays the media which disseminates fake information and deceptions through traditional platforms along with the online platforms, popularly known as Social Media. On the social media platforms, a growing interest has been seen in false news, most probably because of politics today, and few more factors [1–3]. Tracking down wrong information on social media is very crucial and at the same time full of challenges. The trouble is partially because of the reality that even mankind can't correctly differentiate between the fake news and the real news, mostly for the reason that it requires tiresome proof collection and checking the facts warily. Due to the

V. Mehta (✉) · R. K. Mishra
Department of Computer Science, BITS Pilani,Dubai Campus, Dubai International Academic City, Dubai, United Arab Emirates
e-mail: f20180153@dubai.bits-pilani.ac.in

R. K. Mishra
e-mail: rkmishra@dubai.bits-pilani.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_8

emergence of new technology, more and more advancements in the technology and accelerating spread of fake articles, it's becoming more essential to devise automated infrastructure for detecting fake news [4].

Communicating electronically has assisted to eradicate distance and time barriers to distribute and transmit the information. In 2019, COVID-19 virus was noticed and it's been more than a year now in April, 2021. There are millions of cases till date and millions of deaths. Mostly all the countries in the world are facing the fatal COVID-19 epidemic. In these times, we are also surrounded by fake news concerning the disease, healing strategies, how we can avoid the virus, and reasons for getting affected. The dissemination of false news and the wrong information at the time of such dangerous situations, can have outstanding repercussions resulting in prevalent fear and increasing the risk of the epidemic itself. Hence it is most important to restrict the circulation of fake news and make sure that correct knowledge is spread among the people [5].

Social Media allows communication and spreading of fresh ideas [6]. Therefore, users of social media can promote thoughts and disseminate the news in the form of retweets, and etc. Thus, the users are always vulnerable to ungovernable information, particularly news from individualistic authors. It has been revealed that social media is a powerful contrivance for disseminating significant amounts of non-filtered content, allowing wrong information and therefore aggravating the chance of controlling the people's insights of actuality through the spreading of false news [7]. Circulating fake news in the current digital environment has become uncontrollable. Few government officers even involve in the spreading of wrong information to many people for their own benefit [8].

The primary objective of this research is to classify the fake information on Tweets tweeted by the users as fake or not fake. We have proposed various methods by which fake news can be discovered easily. We also get the F1-scores for all the methods and we observe that Logistic Regression performs better with F1-score 0.93.

2 Related Work

Nistor [9], propose a guide learning technique to identify the true information from bogus information in the course of COVID-19 epidemic. Their guide insinuates a model for differentiating fake news in the social media platforms. The model, like Machine Learning algorithm, may have accuracy around 95% and thus can help in detecting fake news. They have tried to detect fake news in much simpler ways and have employed Naive Bayes Method. Gundapu and Mamidi [10], propose a method to examine the dependability of info shared on online platforms relating to COVID-19 epidemic. Their method is predicated on grouping of 3 transformer models "BERT, ALBERT, and XLNET" to identify false news. The result they received on the test set was 0.9855 F1 score and their ranking was fifth among one hundred and sixty crews.

Mookdarsanit and Mookdarsanit [11], have put forward a modern approach for detecting fake news regarding COVID in Thai. Their work is predicated on transfer learning. Their paper includes preparing Thai text, translating the dataset from English to Thai language, feature shifting and Thai-NLP technique on the basis of transfer learning, respectively.

Manzoor and Singla [12], mention different Machine Learning methods for identifying false and fictitious news. They also mention the constraints of such kinds of approaches and improvising done through applying deep learning. They come to a conclusion that, in future, plenty of studies in this domain will apply deep learning models, such as Convolutional Neural Network (CNN), Deep Neural Network (DNN), Deep Autoencoder in Natural Language Processing (NLP), and etc.

Gupta et al. [13], resolve the issue of identifying antagonistic and false information in the "Devanagari (Hindi) script" as multilabel and multiclass problem. With the help of the NLP method, they developed a model which refers to a foul language detecting device. Their model attains 0.97 F1-score. Moreover, they developed models to detect false news in relation to COVID in tweets (in English). They attain 0.93 F1-score on false news (in english) identification work and they leverage entity info pulled out from tweets in addition to texts educated from word embedding.

Bang et al. [14], take two different methods to attain a powerful model for COVID-19 false news identification. The first method they use is "fine tuning transformer based language model" with strong loss function. The second method they use is eradicating dangerous train cases with the help of influence computation. Additionally, they assess the strength of their models by assessing the various COVID wrong information testset to know the capability of model generalisation. Using the first method, they attain 98.13% for weighted F1-score for common tasks, while 38.13% for weighted F1-score on Tweets-19 maximum. They carried out their work on two different COVID-19 testsets. Also, they propose the significance of model generalization capability in this particular work to come ahead to address the COVID-19 false information issue in social media platforms.

3 Methodology

Figure 1 depicts the block diagram of our approach. We first collected the COVID-19 twitter data. Then we cleaned the dataset and preprocessed it. Further, we classified the data into fake and real and then we merged the classified data. Then we partitioned the data in such a way that we kept 0.7 for the train and 0.3 for the test. Furthermore, we converted the data from text to vector form for model training. Finally we feed in the tweets from the dataset and we get the predicted result whether the news is fake or not.

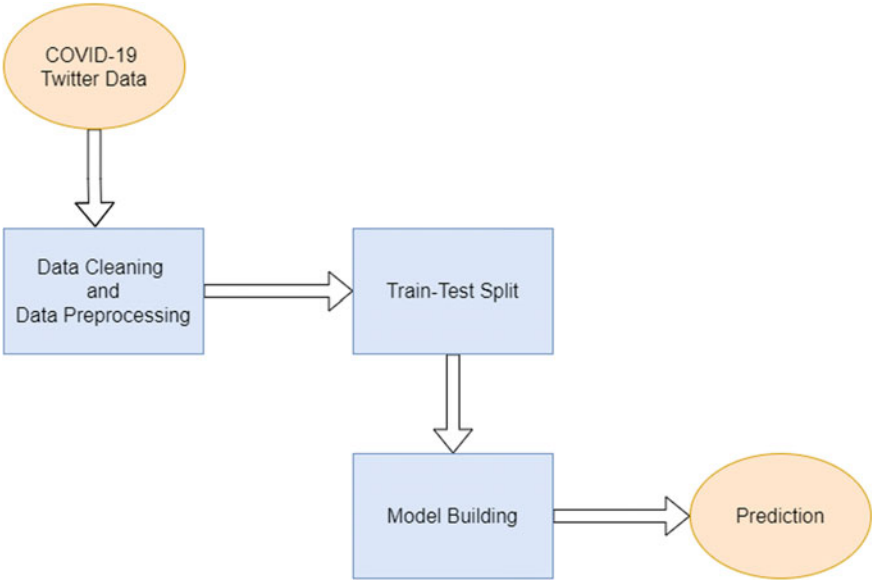


Fig. 1 Block diagram

3.1 COVID-19 Twitter Data

We used dataset from fact verification sources [15] and the data consists of the tweets related to COVID-19. Some tweets are fake while some are true. We divided the dataset into three parts. One dataset contains only fake tweets and another contains only real tweets. The third dataset consists of random tweets from real and fake dataset. We will use this dataset for training. The two types of tweets are: Real - Tweets from trusted resources and provides beneficial info on COVID-19. Fake - Tweets, public posts, news that speculates regarding the virus are checked to be not real.

3.2 Data Cleaning and Data Pre-processing

As we know that twitter data contains so much extra information, we removed extra spaces, special characters like "@" and urls. Even keeping all words in lowercase gives a more accurate result, so we converted all uppercase letters to lowercase letters.

For converting the text to vector, we use TF-IDF, which stands for "Term Frequency-Inverse Document Frequency". It is employed to calculate the significance of a word in a document provided. It is also employed whenever we attempt to fetch certain common data from text or review's data. This technique gives a weight

to a word depending on its incidence in the document. The word's frequency gives a method for the data's score or rank and finds the similarity relation between the words.

It is used extensively if we want to pull out a few customary pieces of information from the data [16]. In Tf-IDF, the documents are delineated as vectors. But rather, a vector of 1's and 0's, the document consists of a score for every word [17].

We classified the dataset in two categories. First: real and Second: fake. This classification was already provided with real and fake values and we found that 3050 were the fake tweets and 3350 were the real tweets.

3.3 Train Test Split

Once our dataset is ready we partition our original dataset into two parts, train and test dataset. Train dataset contains 70% of the whole dataset and 30% is being allotted for test purpose, model predictions, and validating model predictions.

3.4 Models

We have applied four different models on our data. Those models are as follows.

3.4.1 Gradient Boosting Classifier (GBC)

This model is a group of machine learning algo's which unite several weak learning methods collectively to produce a strong prediction method. GBC methods are becoming famous due to their efficacy in the classification of complicated data sets, and have lately been employed in winning several "Kaggle data science" contests. The concept behind Gradient boosting is to take a "weak learning" or "weak hypothesis" algorithm and make a number of adjustments which will help in improving the robustness of the hypothesis or learning.

3.4.2 Logistic Regression (LR)

It is a model which employs a logistic function to form a "binary dependent variable", even though several complicated extensions are present. It is employed to achieve the ratio of odds in existence of multiple variables.

3.4.3 Random Forest Classifier (RFC)

Random Forest is a "meta estimator" which suits several Decision Tree Classification on different sub-samples of the data set and it makes the use of average to enhance the prediction accuracy and also it regulates over-fitting.

3.4.4 Decision Tree Classifier (DT)

They use more than one algorithm to determine to divide a node into multiple nodes. Building the new nodes gives rise to "homogeneity" of consequent subnodes. It divides the nodes on each variable which are obtainable followed by selecting the split that gives rise to most homogenous subnodes.

4 Experimental Result

The architecture (shown in Fig. 2) contains Training dataset and Testing dataset. Model training process is applied to the training dataset and then we apply four different models to this dataset. We apply "Gradient Boosting Classifier (GBC)", "Logistic Regression (LR)", "Random Forest Classifier (RFC)", and "Decision Tree Classification (DT)". Also, we feed in a test dataset to the models to get the prediction

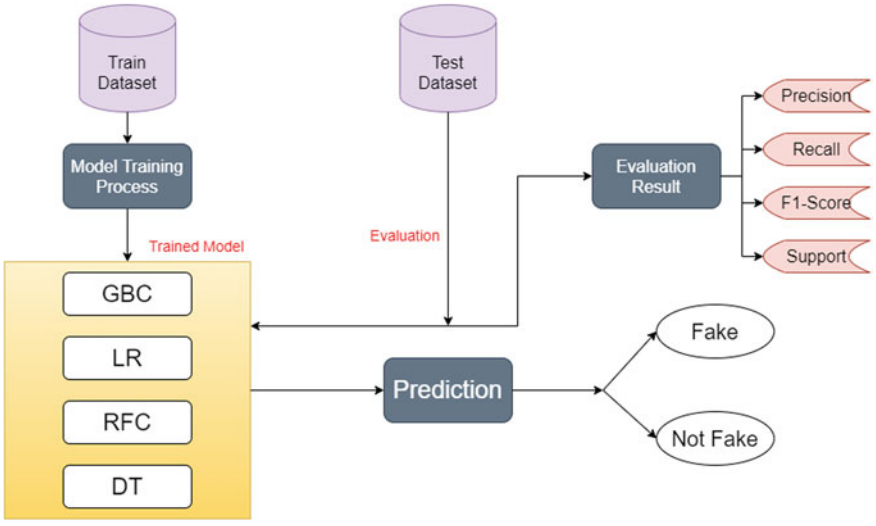


Fig. 2 Model architecture

Table 1 Evaluation of Different Models

Model Performance				
Model	Precision	Recall	F1-Score	Support
GBC (Fake)	0.90	0.86	0.88	759
GBC (Not Fake)	0.88	0.91	0.89	841
LR (Fake)	0.93	0.92	0.93	759
LR (Not Fake)	0.93	0.94	0.93	841
RFC (Fake)	0.87	0.90	0.89	759
RFC (Not Fake)	0.91	0.88	0.89	841
DT (Fake)	0.86	0.83	0.85	759
DT (Not Fake)	0.85	0.88	0.87	841

of whether the tweets are fake or real. After passing the test dataset to the model, we verified the results of "precision, recall, F1-score, and support".

Table 1 shows the models and their values of "precision", "recall", "F1-score", and "support". All models have two results: results for fake news and results for real news. From the results, we can see that Logistic Regression performs better than other models as its F1-score is highest- 93%.

5 Conclusion

Presently, in the information era, individuals have endless access to information, as tech progresses and communication platforms grow, yet amidst all this info, which is advantageous to us, is fake news. The fake news causes numerous difficulties in the present world, and it is of utmost importance that we make an attempt to detect them. Various researchers have attempted in the last few years ways to identify false information on different platforms of social media, such as Instagram, LinkedIn, etc.

Research done till now demonstrates that machine learning provides great outcomes in respect to the accuracy of fake information revelation, and hence the model can be a nice resolution for a considerable count of users of social media platforms. For the present study, we present several models for identifying bogus tweets pertaining to COVID-19. These models can eradicate the dissemination of false information on this topic. We trained our data and tested it using different models in order to get the best result. We found the Logistic Regression is best among all, as it has the highest F1-score, that is 93%.

References

1. Tucker J et al (2018) Social media, political polarization, and political disinformation: a review of the scientific literature. SSRN Electron J
2. Calvillo DP, Ross BJ, Garcia RJB, Smelter TJ, Rutchick AM (2020) Political ideology predicts perceptions of the threat of COVID-19 (and susceptibility to fake news about it). Soc Psychol Personal Sci 194855062094053
3. Monti F, Frasca F, Eynard D, Mannion D, Bronstein MM (2019) Fake news detection on social media using geometric deep learning. arXiv [cs.SI]
4. Das SD, Basak A, Dutta S (2021) A heuristic-driven ensemble framework for COVID-19 fake news detection. arXiv [cs.CL],
5. Vijjali R, Potluri P, Kumar S, Teki S (2020) Two stage transformer model for COVID-19 fake news detection and fact checking. arXiv [cs.CL], pp 1–10
6. Zhou X, Zafarani R, Shu K, Liu H (2019) Fake news: fundamental theories, detection strategies and challenges. In: Proceedings of the twelfth ACM international conference on web search and data mining
7. Lazer DMJ et al (2018) The science of fake news. Science 359(6380):1094–1096
8. Apuke OD, Omar B (2021) Fake news and COVID-19: modelling the predictors of fake news sharing among social media users. Telematics Inform 56. https://doi.org/10.1016/j.tele.2020.101475
9. Nistor A (2021) Fake news detection about SARS-cov-2 pandemic using neutral networks and detection algorithms. EcoForum 10(1)
10. Gundapu S, Mamidi R (2021) Transformer based automatic COVID-19 fake news detection system. http://arxiv.org/abs/2101.00180
11. Mookdarsanit P, Mookdarsanit L (2021) The covid-19 fake news detection in thai social texts. Bull Electr Eng Inform 10(2):988–998. https://doi.org/10.11591/eei.v10i2.2745
12. Manzoor SI, Singla J (2019) Fake news detection using machine learning approaches: a systematic review. In: 2019 3rd international conference on trends in electronics and informatics (ICOEI), pp.230–234. https://doi.org/10.1109/ICOEI.2019.8862770
13. Gupta A, Sukumaran R, John K, Teki S (2021) Hostility detection and Covid-19 fake news detection in social media. http://arxiv.org/abs/2101.05953
14. Bang Y, Ishii E, Cahyawijaya S, Ji Z, Fung P (2021) Model generalization on COVID-19 fake news detection. http://arxiv.org/abs/2101.03841
15. Patwa P et al (2021) Fighting an Infodemic: COVID-19 fake news dataset. In: Combating online hostile posts in regional languages during emergency situation. Springer International Publishing, Cham, pp 21–29
16. Mishra RK, Urolagin S, Jothi AAJ (2019) A sentiment analysis-based hotel recommendation using TF-IDF approach. In: Proceedings of 2019 international conference on computational intelligence and knowledge economy, ICCIKE 2019, pp 811–815. https://doi.org/10.1109/ICCIKE47802.2019.9004385
17. Das B, Chakraborty S (2018) An improved text sentiment classification model using TF-IDF and next word negation. arXiv [cs.CL]

Adverse Effect of Electromagnetic Frequency on Human Being Health Under Real-life Scenario: A Comparative Study



Gaurav Gandhi and Sharad Sharma

Abstract The increasing use of electronic gadgets like mobile phones, laptops, smart television has led to a corresponding increase in public attention to their potential health hazards. These issues involve the impact of electromagnetic frequency (EMF) radiation on the health of human beings, and it is a multi-factor problem, which needs to be studied, and related research must be published in research journals. But, apart from Russia, no other country has taken real-time safety precautions from EMF exposure. In this study, a survey is presented that covers multiple research articles that have used the concept of machine learning (ML) for the prediction of possible damage from EMF radiations. From research, we observed that artificial neural network (ANN) and random forest (RF) are the two classification approaches that performed better with higher accuracy.

Keywords Electromagnetic frequency · Plants · Animals · Human · Machine learning approaches

1 Introduction

Electromagnetic radiation is the fluctuation of electric and magnetic fields. The speed of propagation in a vacuum is equal to light (about 300,000 km/s). In other environments, the propagation speed of radiation is slower. Electromagnetic radiation is classified by frequency range. The boundaries between the ranges are somewhat arbitrary; there are no abrupt transitions [1].

Visible light: This is the narrowest range in the entire spectrum. Visible light combines the rainbow's colors that are violet, indigo, blue, green, yellow, orange,

G. Gandhi (✉)
Maharishi Markandeshwar Deemed To Be University, Mullana, India

S. Sharma
Electronics and Communication Engineering Department, Maharishi Markandeshwar Deemed University, Mullana, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_9

and red. Behind the red color, there is infrared radiation, and behind the violet-ultraviolet, but they are no longer distinguishable by the human eye. Visible light waves are very short and high frequency. The length of such locks is one-billionth of a meter or one billion nanometers.

Ultraviolet radiation: This is the part of the spectrum between visible light and X-rays. Ultraviolet radiation is used to create lighting effects on the theater stage, discos; banknotes of some countries contain protective elements visual only in ultraviolet light [2].

Infrared radiation: This is the part of the spectrum between visible light and short radio waves. Infrared radiation generates more heat than light: Every heated solid or liquid body emits a continuous infrared spectrum. The higher is the heating temperature, the shorter the wavelength and higher the radiation intensity.

X-ray radiation (X-ray): X-ray waves have the property of passing through matter and not being absorbed too much. Visible light does not have this ability. Due to X-rays, some crystals can glow.

Gamma radiation: This is the shortest electromagnetic waves that pass through matter without being absorbed: They can overcome a one-meter wall of concrete and a lead barrier several centimeters thick [3].

As shown in Fig. 1, the electromagnetic frequency (EMF) wave is categorized into two main frequencies, namely ionizing and non-ionizing. The main sources of EMFs are the transmission towers/lines. The issues related to this have been observed for the last couple of years. Other sources of EMFs are the mobile phone, smart communication devices, and the recent fourth-generation (4G) and coming fifth-generation (5G) technologies [5].

Neither 4G nor 5G has been tested against real scenarios with incredible security. Shockingly, many studies conducted in milder environments have shown that such radiation can have harmful effects. This article outlines the medical and biological research done up to date with respect to the effects of EM radiations and explains

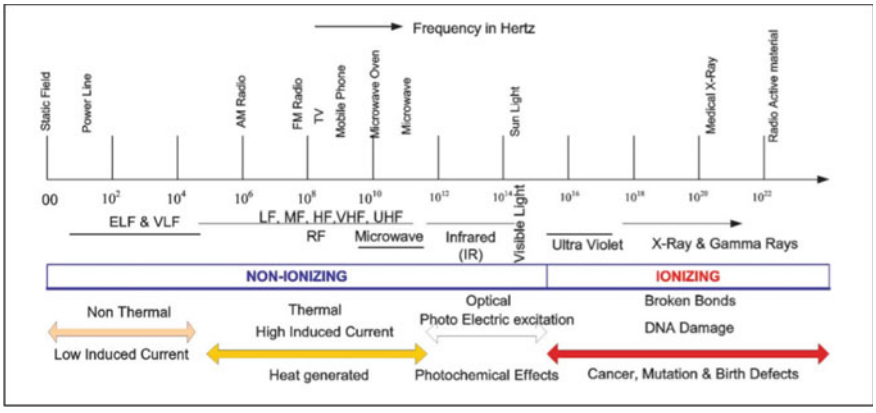


Fig. 1 Electromagnetic spectrum [4]

why these studies are still insufficient relative to safety [6]. From state of the art, one can easily say that EMF has a diverse effect on human health compared to real-life toxic chemicals. The research that shows effects on the health of living beings from the use of wireless radiation devices has been reported in the literature [7].

Rest of the paper is organized as followed: In Sect. 2, sources of EMF generation along with modern wireless radiation exposures. In Sect. 3, a review of literature is provided that explained the effects of EMF on human as well as animal health. In Sect. 4, performance parameters along with mathematical expression are described. The conclusion is presented in Sect. 5 followed by references.

2 Sources of EMF Generation

The propagation of both electric and magnetic fields through space is referred to as EMF. It is measured in Tesla. The range of frequencies in which the EMF spectrum is divided includes γ-rays, X-rays, ultraviolet rays, visible, infrared, microwaves, radio frequency, and ELF range. The waves having a frequency less than 300 GHz are referred to as EMF [8].

The various sources of EMF generation are shown in Fig. 2. These include transmission line, wireless communication tower, television (TV), radio, mobile phone, laptop, hairdryer, wireless fidelity (Wi-Fi), electrical types of equipment. How the frequency of the EMF spectrum is used by these sources is listed in Table 1.

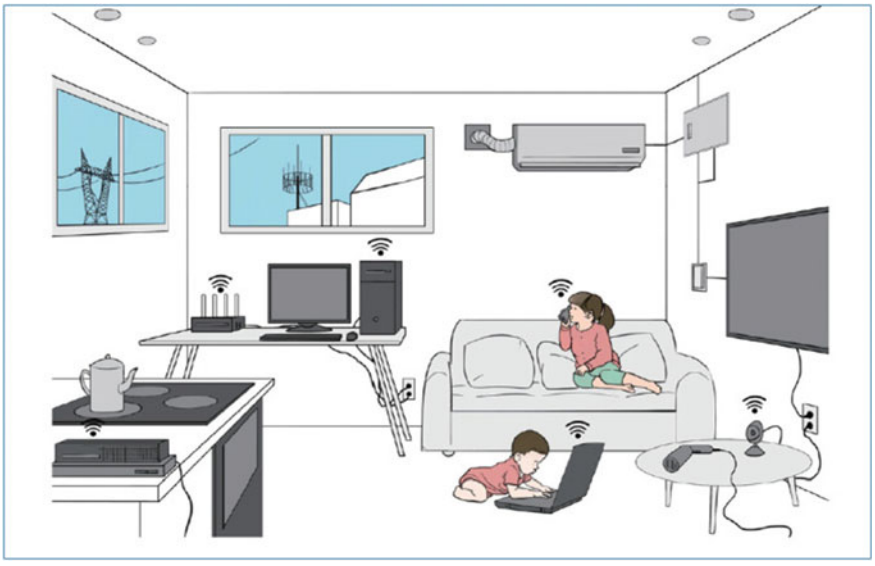


Fig. 2 Various sources of EMF exposure [9]

Table 1 Uses of EMF [10]

Frequency type	Frequency range	Uses
Low	3 Hz–300 kHz	Transmission line
Medium	300 kHz–900 MHz	AM, FM, and TV broadcasts
Lower microwave	900 MHz–5 GHz	Telecommunication
Higher microwave	5–300 GHz	Radar and later for 5G
Terahertz	300–3000 GHz	Health and security

In the current article, effect of EMF on the health that used the spectrum from 3 Hz to 300 GHz is discussed.

2.1 Modern Wireless Radiation Exposures

In the distant past, sunlight and its reflections on the moon provided most of the visible spectrum for human beings. But, nowadays, many kinds of artificial light such as incandescent and fluorescent lamps and light emitting diodes (LEDs) have now replaced the sunlight and become the main supplier of visible or natural light during working hours. In addition, EMF radiation from other parts of the non-ionizing invisible spectrum has become everywhere in daily life, such as from telecommunications. In the last couple of years, the explosive growth of the cellular phone industry has resulted in many residential areas located less than a mile from the cellular base station. The future realization of next-generation 5G mobile network technology will increase the density of base station signal towers by order of magnitude. Health problem arises due to (i) mobile communication equipment, (ii) occupational and residential exposure, (iii) wireless communication system used in homes, commerce, and schools, (iv) use of vehicles, and (v) wireless radiation generated by other non-ionizing EMF radiation resources, including Internet of Things (IoT) [11].

3 Related Work

Some studies have shown that potential health hazards may appear due to unnecessary exposure of living beings, humans, or animals to high power density radio waves. These health hazards include:

- Cancer
- Tumors
- Headaches
- Fatigue.

Most of the world region is covered by the EMF radiation sources, such as radio, mobile phone communication, and television transmitters, but there is no survey paper till now that presents the effect of EMF radiation on living things, including plant, human, and animals. The work performed to know the effect of EMF radiation on living beings is discussed in the following section.

Erogul et al. (2006) have investigated the effect of EM radiations on human sperm. To perform this, sample of 27 males has been taken, and the sample of sperm has been divided into two parts. One part of which is exposed to EM radiation generated from mobile phone around 900 MHz, whereas the other kept without exposure to EM radiation. From the test, it has been observed that EM radiation slightly decreases in the rapidly progressive and slow progressive sperm movement [12].

Halgamuge et al. (2015) have studied the impact of radiation emitted from mobile phones on soybean seeds. In the presence of mobile phone radiation and transmission base stations, the growth rate of soybeans has been analyzed. Four days of soybean plant has been exposed to gigahertz (GHz) and continuous wave (CW) with the signal magnitude of 5.7 and 41 V and frequency of 900 MHz. The high CW results in the decreased outgrowth of the roots in addition to this smaller exposure of CW results in the decreased outgrowth of the hypocotyls. The result shows that the growth of soybean is influenced depending upon the amplitude of EM frequency [13].

Shehu et al. (2016) have examined the effect of exposure of EM radiations generated from the mobile phone. The effect of radiation during ringing and vibration mode on the rat behavior has been analyzed. Five groups of Wistar rats have been taken, each comprised of five animals. Each group is exposed to a mobile phone, while the phone is operated in a diverse mode like Group 1 in switch off mode, Group 2 (silent mode), Group 3 (vibration mode), Group 4 (ringtone mode), and Group 5 (vibration and ringtone both mode). Each group is exposed to a 10-min call, for 30-min missed call per day. This process is repeated for 1 month. After 1 month, neuro-behavior analysis has been performed on the basis of the collected samples of the brain. The result shows that ringtone and vibration, both in combinations as well as individually, affect the behavior of Wistar rat [14].

Liptai et al. (2017) have studied the influence of Wi-Fi radiation on the growth of plants preferably including the garden cress. The researchers have taken two samples one of which was exposed to EM radiation for long time in the surrounding area of Wi-Fi router. Another sample has been exposed to the same radiation but for smaller duration. The result indicates that variation in the growth of plants as well as in its appearance has been observed, while the cress was exposed for longer duration. The results show that with the passage of time, the growth of plants reduces and maximum reduction in the growth has been observed after twelfth day of the experiment [15].

Using ML and big data analysis, high-level knowledge can be extracted from raw data and later on used to enhance the performance of automation tools that contribute to the health field. ML is a key tool in analysis, where the system learns from data to find out hidden insights. For experts, it is very challenging to ignore the important details of huge data (Kononenko 2001) [16]. Therefore, instead of using automated tools to process raw data and extract valuable information, it is essential for decision-makers. ML techniques have been employed for big data analysis; however,

the challenge is to construct predictive models for data with several variables. Raw data captures key information, such as patterns and movement that can be used to precede decision-making and optimize performance. This section presented a brief discussion about ML techniques followed by the research articles presented by various researchers in the last decade (Cortez and Alice 2008) [17].

3.1 Machine Learning

ML is a type of artificial intelligence (AI) that enables the framework to receive and improve information without explicit automatic programming. This focuses on the development of computer programs that can access data and use it to 'learn' on its own. ML algorithms learn from past experience and provide reliable solutions for the considered problem [16, 18, 19]. ML is a part of AI, which instructs the computer to acquire knowledge by iteratively extracting information and identifying patterns from the raw data, and also known as a learning algorithm [20].

All living things, including plants and animals, contain the same basic building blocks of DNA. The plant kingdom provides a series of assessment endpoints, and these endpoints are frequently overlooked when looking for basic biological results related to exposure to EMF radiation. Although, there is still controversy surrounding the effect of EMF explored on the human body.

Halgamuge (2017) has studied the effect of EM frequency radiated from mobile phones on plants using ML algorithm. The raw data was collected from 45 peer reviews published in between the years from 1996 to 2016, and experiments of 169 cases have been conducted to predict the potential influence of radio frequency electromagnetic fields (RF-EMF) on plants. The eight ML techniques have been used to identify (i) the condition above which the plants cannot be exposed to EM frequency, (ii) the frequency which is safe for plants, and (iii) which plant species are influenced by the EM frequency. The highest classification, about 95.26% with a minimum error of 0.084%, was attained by random forest (RF) and the K-means algorithm [21].

Parsaei et al. (2017) have used multilayer perceptron neural network (MLP-NN) algorithm to design a prediction model to predict EMF radiation's effect on human health, while the human body is exposed to mobile base station. The dataset of 699 human beings, which includes 363 men and 336 women, has been considered for the experiment. From the total 699 data, 70% of data has been used for training, remaining 30% of data is used for validation (15%) and testing (15%). The experiment shows that symptoms like headaches, fatigue, sleep disturbance, dizziness, and vertigo have been seen in humans subjected to EMF radiation. The experiment shows that the designed system is suitable for predicting the symptoms in the human body that were exposed to the mobile base station. Using the MLP-NN scheme, parameters like sensitivity and specificity have been observed [22].

Halgamuge and Davis (2019) have presented an ML approach to identify the effect of EMF on plants. The data has been collected from previous papers that cover

the year from 1996 to 2016, which consists of 169 experimental cases. About six different attributes, namely frequency, specific absorption rate (SAR), power flux density, electric field strength, exposure time, and plant type (species) have been taken into consideration. Using K-nearest neighbor (KNN) and RF as ML approach, the dataset is classified. The performance of the work has been analyzed in terms of prediction accuracy and computation time [23].

Singh et al. (2020) have studied the effect of EMF on the human brain. The brain features like volume and shape are automatically divided by acquiring microscopic images, and distinguishable geometric features are extracted from the microscopic images to determine the impact of EMF exposure. The geometrical properties of the brain images of Drosophila micro-segmentation have been analyzed and found to have discriminative properties consistent with the ML method. The most popular discriminate function has been assigned to four different classifiers: support vector machine (SVM), naive, artificial neural network (ANN), and RF, which are used for performing exposure/unexposed microscopic images classification. Using the method based on feature selection, the classification accuracy is as high as 94.66%, and low time complexity has been attained [24].

Kiouvrekis et al. (2020) have studied the effect of SRM 3006 spectrum analyzer with a frequency range of (27 MHz–3 GHz) on school children in Greece. The exposure of EMF has been analyzed using the unsupervised ML technique. Using the ML approach, mean value for EMF radiation exposure has been analyzed [25].

Maurya et al. (2020) have presented an automatic detection of EMF radiation-affected brain image using image processing technique. Brain features like volume and size have been extracted using the brain morphology scheme. These features have been optimized using a genetic algorithm as an evolutionary-based approach. These optimized features are then fed to the ANN as an ML approach. From the result, the experiment shows that the designed model distinguished between the EMF exposed brain image and the non-exposed brain image, respectively [26].

Tafakori et al. (2020) studied the effect of RF-EM signal on the rat brain. The rat is exposed to the 900 MHz frequency radiated by the dipole antenna for at least 3 h a day. The radiation exposure time was initially recorded for 7 days, and then exposure time was increased to 1 month. The behavioral parameters of rats were analyzed after 1 week and 1 month by implanting microelectrodes on the rats. The test results show that the classification accuracy is as high as 84.2%. When the rat was exposed to the θ zone, the rat's task was observed to be delayed, and when exposed to the RF-EMR wave, the behavior was normal. The results concluded that when the brain is exposed to RF-EMR signals, abnormal functions may occur. Using SVM and linear discriminant analysis (LDA) as an ML approach, the difference between the true normal trial and true delayed trials have been performed [27] (Table 2).

Table 2 Comparative analysis

Ref. no.	Number of experiments	Type of exposure	PCC	MAE/RMSE	Precision/recall/*F*-score	Computation time (s)	Exposure time
[13]	12	GSM/CW					2 h
[21]	169	RF-EMF	94.08	MAE(0.0824) RMSE(0.2242)	p(95), R(95.3), fs(0.2)	0.2	–
[22]	699	Mobile base station			Headache (p(7.18), R(90.9), fs(83.8)), sleep disturbance (p(82.1), R (83.3), fs(82.9)), dizziness (p(65.2), R(85.4), fs(81.0)), vertigo (p(65), R(84.7), fs(81.0)), fatigue (p(8.3), R(98.9), fs(88.6)		
[23]	169	RF-EMF	RF (91.17%), KNN (89.41%)	–	–	KNN (3.38 s), RF ((248.12 s)	Maximum of (3,153,600 min) and minimum (0.013 min) for tomato
[24]	155 sample image	EMF radiations	SVM (94.66), naïve Bayes (93.33), ANN (87.10), RF (84)		SVM (p(88.47), R(95.83), 91.17), naïve Bayes (p(96.15), R(86.21), fs(90.90)), ANN (p(78.79), R(89.66), fs(82.63)), RF (p(100), R(68.42), fs(81.3))		

(continued)

Table 2 (continued)

Ref. no.	Number of experiments	Type of exposure	PCC	MAE/RMSE	Precision/recall/*F*-score	Computation time (s)	Exposure time
[25]	SRM 3006 spectrum analyzer with frequency range of (27 MHz–3 GHz)	EMF radiations	–	–	–	–	3500 times below the Greek exposure
[26]	158 total brain images	1800–2700 MHz EMF source	Naïve Bayes (86.11), SVM (87.50), ANN (97.91), and RF (96.52)	–	Naïve Bayes (p(91.9), R(81.70), fs(86)), SVM (p(88.57), R(86.48), fs(87)), ANN (p(98.6), R(97.26), fs(97)), RF (p(98.55), R(94.67), fs(96))	–	–
[27]	12 male rats	900 MHz	SVM (84.2), LDA (81.6)	–	–	SVM (0.22) and LDA (0.94)	3 h a day

4 Comparison of Result

Using ML technology, collected data is divided into two parts, one is training, and the other is testing. Using testing block, trained data is tested based on a number of evaluation parameters. Some of the evolution parameters discussed by LeCun et al. [21] and Parsaei et al. [23] are discussed below.

(i) **Percentage of correct classification (PCC)**
This parameter is used to estimate the accuracy of the classification algorithms, which is expressed by Eq. (1).

$$\gamma(i) = \left\{ \begin{array}{l} 1 \text{ if } b_i = q_i \\ 0 \text{ else} \end{array} \right\} \tag{1}$$

where b_i and q_i represent the actual and predicted value using classification algorithm at ith instance. Using Eq. (1), PCC can be defined by Eq. (2)

$$\text{PCC} = \sum_{i=1}^{n} \frac{\gamma_i}{n} \times 100 \tag{2}$$

(ii) **Computation time**
It is defined as the time taken by the designed model to predict event using classification algorithm.

(iii) **Mean absolute error (MAE)**
This parameter is used to measure the average of absolute error. It is calculated by subtracting the actual value from the predicted value. If the actual values are $b_1, b_2, \ldots b_n$ and prediction instances are $q_1, q_2, \ldots q_n$, then the MAE for n diverse prediction instances can be represented by Eq. (3).

$$\text{MAE} = \frac{q_i - b_i}{q_n - b_n}/n \tag{3}$$

(iv) **Root-mean-square error (RMSE)**
It is an essential parameter need to be evaluated while determining the performance of ML approach. A classifier with 100% PCC is considered as accurate classifier. But, a classifier with small RMSE near to zero is also necessary to design an accurate classification approach of a prediction model. Mathematically, RMSE is represented by Eq. (4).

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} (q_i - b_i)^2/n} \tag{4}$$

(v) **Confusion matrix**
This parameter is used to evaluate the classifiers rate to predict the actual sensitivity of the designed prediction model. It is calculated by measuring precision, recall, and F-score.

(vi) **Precision**
This parameter represents the percentage of correctly predicted items from the positive class, which is given by Eq. (5).

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

(vii) **Recall**
This parameter represents the percentage of correctly predictive items from the total positive class, which is given by Eq. (6)

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

(viii) ***F*-score**
It is defined as the harmonic means of P and R, which can be calculated using Eq. (7)

$$F = \frac{2PR}{P + R} \tag{7}$$

Figure 3 represents the comparison of PCC examined by many researchers to predict the health of human being using ML as a classification approach. From the graph, it has been analyzed that among multiple classification algorithms, namely RF, SVM, KNN, naïve Bayes, ANN, and LDA, ANN as ML approach performed better. This is because ANN can work with incomplete knowledge of data. After training the system, the system can provide output with a small knowledge.

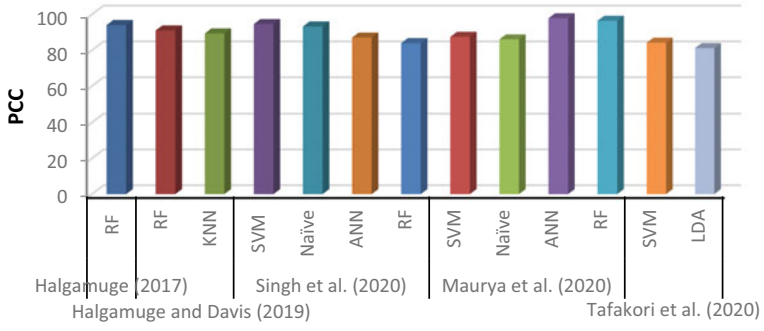


Fig. 3 Comparison of PCC

The second highest PCC has been obtained by the RF classifier. This is due to its ensemble behavior, which results in the minimization of the overfitting problem and reduces the variance that enhances the accuracy of the system. From the existing result, it has been concluded that using the RF approach maximum PCC.

Figure 4 represents the comparison of three different parameters, including precision, recall, and *F*-score. Each parameter is represented by the blue, the red, and the green bar, as shown in Fig. 4. From Fig. 5, it has been analyzed that the RF classifier used by Halgamuge and Davis [24] has performed with the highest precision. On the other hand, highest recall of (97.26%) has been observed by the ANN algorithm used by Kiouvrekis et al. [26] work.

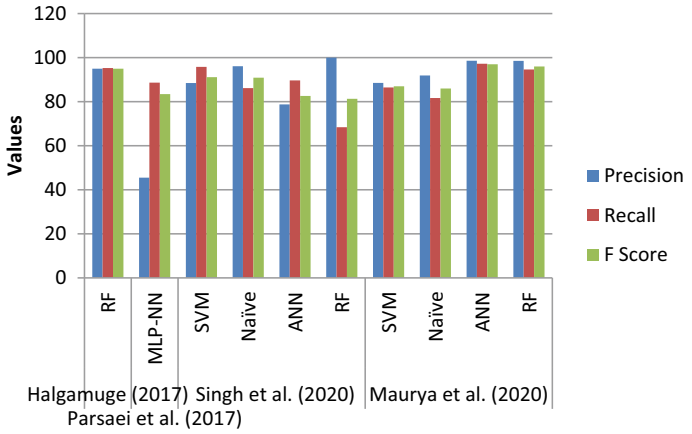


Fig. 4 Comparison of precision, recall, and *F*-score

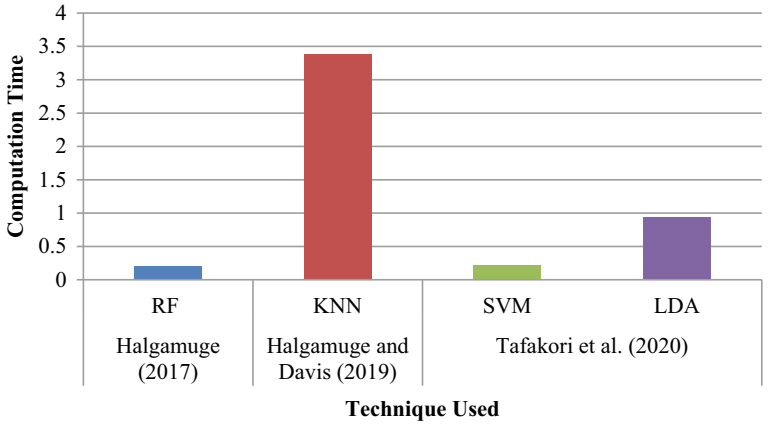


Fig. 5 Comparison of computation time

Figure 5 represents the comparison of computation time analyzed by various ML techniques, namely RF, KNN, SVM, and LDA by three authors [21, 23, 27], respectively. From the graph, it has been observed that using the RF approach, minimum computation time of 0.22 s has been observed compared to other existing ML approaches.

5 Conclusion

Accurately predicting the risk of living beings' health among residents living near EMF radiated types of equipment, including mobile phones or other electronic gadgets, can improve their quality of life by providing appropriate medical care and suggesting efficient methods to minimize the severity of these symptoms. The use of mobile phones has become an essential part of our society because it has become a social and informational device, achieving great success. ML technology can perform advanced extraction from raw data, thus providing an excellent opportunity to predict future trends and results of electronic devices and their effects on living things. No single research can fully cover the concept of ML in the field of electromagnetic domain. However, this article has surveyed a number of the predictive model and its accuracy to determine the best classification algorithm for analyzing data that shows the environmental impact of EMF radiations on animals, plants, and humans. This analysis helped us to understand the diverse types of attributes that affect the lives of human beings. Among all classification algorithms, it has been observed that the RF and ANN algorithms have produced better predictions.

References

1. Siegert AJF (1937) Note on the interaction between nuclei and electromagnetic radiation. Phys Rev 52(8):787
2. Gotsis A, Papanikolaou N, Komnakos D, Yalofas A, Constantinou P (2008) Non-ionizing electromagnetic radiation monitoring in Greece. Ann Telecommun—annales des télécommunications 63(1):109–123
3. Sowa P, Rutkowska-Talipska J, Sulkowska U, Rutkowski K, Rutkowski R (2012) Ionizing and non-ionizing electromagnetic radiation in modern medicine. Polish Ann Med 19(2):134–138
4. Zamanian A, Hardiman CJHFE (2005) Electromagnetic radiation and human health: a review of sources and effects. High Freq Electron 4(3):16–26
5. Calvente I, Fernandez MF, Villalba J, Olea N, Nuñez MI (2010) Exposure to electromagnetic fields (non-ionizing radiation) and its relationship with childhood leukemia: a systematic review. Sci Total Environ 408(16):3062–3069
6. Gultekin DH, Siegel PH (2020) Absorption of 5G radiation in brain tissue as a function of frequency, power and time. IEEE Access 8:115593–115612
7. Moskowitz JM (2019) We have no reason to believe 5G is safe. Scientific American
8. Deruelle F (2020) The different sources of electromagnetic fields: dangers are not limited to physical health. Electromagn Biol Med 39(2):166–175

9. Moon JH (2020) Health effects of electromagnetic fields on children. Clin Exp Pediatr 63(11):422
10. Kostoff RN, Lau CG (2017) Modified health effects of non-ionizing electromagnetic radiation combined with other agents reported in the biomedical literature. In: Microwave effects on DNA and proteins. Springer, Cham, pp 97–157
11. Kostoff RN, Heroux P, Aschner M, Tsatsakis A (2020) Adverse health effects of 5G mobile networking technology under real-life conditions. Toxicol Lett 323:35–40
12. Erogul O, Oztas E, Yildirim I, Kir T, Aydur E, Komesli G, Irkilata HC, Irmak MK, Peker AF (2006) Effects of electromagnetic radiation from a cellular phone on human sperm motility: an in vitro study. Arch Med Res 37(7):840–843
13. Halgamuge MN, Yak SK, Eberhardt JL (2015) Reduced growth of soybean seedlings after exposure to weak microwave radiation from GSM 900 mobile phone and base station. Bioelectromagnetics 36(2):87–95
14. Shehu A, Mohammed A, Magaji RA, Muhammad MS (2016) Exposure to mobile phone electromagnetic field radiation, ringtone and vibration affects anxiety-like behaviour and oxidative stress biomarkers in albino wistar rats. Metab Brain Dis 31(2):355–362
15. Liptai P, Dolník B, Gumanová V (2017) Effect of Wi-Fi radiation on seed germination and plant growth-experiment. Ann Fac Eng Hunedoara 15(1):109
16. Kononenko I (2001) Machine learning for medical diagnosis: history, state of the art and perspective. Artif Intell Med 23(1):89–109
17. Cortez P, Silva AMG (2008) Using data mining to predict secondary school student performance
18. VoPham T, Hart JE, Laden F, Chiang YY (2018) Emerging trends in geospatial artificial intelligence (geoAI): potential applications for environmental epidemiology. Environ Health 17(1):1–6
19. Patel J, Shah S, Thakkar P, Kotecha K (2015) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst Appl 42(1):259–268
20. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
21. Halgamuge MN (2017) Machine learning for bioelectromagnetics: prediction model using data of weak radiofrequency radiation effect on plants. Mach Learn 8(11):223–235
22. Parsaei H, Faraz M, Mortazavi SMJ (2017) A multilayer perceptron neural network–based model for predicting subjective health symptoms in people living in the vicinity of mobile phone base stations. Ecopsychology 9(2):99–105
23. Halgamuge MN, Davis D (2019) Lessons learned from the application of machine learning to studies on plant response to radio-frequency. Environ Res 178:108634
24. Singh A, Singh N, Jindal T, Rosado-Muñoz A, Dutta MK (2020) A novel pilot study of automatic identification of EMF radiation effect on brain using computer vision and machine learning. Biomed Signal Proces Control 57:101821
25. Kiouvrekis Y, Alexias A, Filipopoulos Y, Softa V, Tyrakis CD, Kappas C (2020) Unsupervised machine learning and EMF radiation in schools: a study of 205 schools in Greece. arXiv:2007.13208
26. Maurya R, Singh N, Jindal T, Pathak VK, Dutta MK (2020) Machine learning–based identification of radiofrequency electromagnetic radiation (RF-EMR) effect on brain morphology: a preliminary study. Med Biol Eng Compu 58:1751–1765
27. Tafakori S, Farrokhi A, Shalchyan V, Daliri MR (2020) Investigating the impact of mobile range electromagnetic radiation on the medial prefrontal cortex of the rat during working memory. Behav Brain Res 391:112703

Consistency Driven Cross-Modal Twitter Sentiment Analysis and Opinion Mining



P. Agalya, S. Ashvathe, R. Harsini, and S. Chitrakala

Abstract Nowadays, social media is being widely used by the public to convey their views and opinions on various topics. Among various social media platforms, Twitter is used to a greater extent for these purposes. Users of social media are increasingly utilising text and images to express their opinions and offer their insights. To overcome the challenges of effectively exploiting the content from both the textual and visual part of multi-modal posts, we propose a consistency based sentiment classification (CBSC) approach. The work presented here is a multi-modal sentiment analysis approach based on image-text consistency. The approach proposed here examines the image-text consistency before applying the adaptive multi-modal sentiment analysis algorithm. Finally, from the classified tweets, the overall public opinion in a particular domain is summarised.

Keywords Mutimodal tweets · Sentiment analysis · Consistency · Opinion mining.

1 Introduction

With the remarkable and ubiquitous rise of social media, people are more likely to share their thoughts on social media platforms, simplifying the decision making. Sentiment analysis of social media especially twitter data plays a pivotal role in understanding people's feelings, opinions, and attitudes on a particular incident. It has risen in prominence in the blink of an eye for tracking people's moods or perceptions by analysing social data. Traditional/Conventional sentiment analysis approach concentrates on a particular modality and hence become inadequate when dealing with the massive amounts of data flowing from social networks. Multi-modal data including both text and image has been enormously available in social websites

P. Agalya (✉) · S. Ashvathe · R. Harsini · S. Chitrakala
Department of Computer Science and Engineering, College of Engineering Guindy Anna University, Chennai, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_10

as social networks have grown in popularity. Even still, combined analysis of multi-modal data on social networks is accorded only a secondary role.

Unlike the conventional sentiment analysis of single modal data, multi-modal sentiment analysis must take into account variety of features. It should be able to successfully connect the space between various modes. It also needs to distil the comprehensive and discriminative information present in each modality. Opinion polls and surveys serve as a bridge between public opinions and politicians. Thus, collecting and analyzing Twitter data remains a cost-effective way for a large number of participants in a short period of time, unlike developing surveys.

A novel image-text consistency based sentiment classification (CBSC) approach is proposed, followed by opinion mining, to address the challenges of successfully exploiting information from both visual and textual part of multi-modal posts. The CBSC technique begins by ensuring that the text and image are consistent, and then employs an adaptive multi-modal sentiment analysis approach. Explicitly, the consistency of the verbal and visual characteristics retrieved from the multi-modal tweet is verified first. Then, two separate sentiment analysis models are built, one for the consistent image-text tweets and the other the inconsistent image-text tweets. Finally, from the classified tweets, the overall public opinion in a particular domain is summarized.

2 Related Work

Sentiment analysis approach proposed in [7] inspects consistency between the text and the image and then a multi-modal adaptive sentiment analysis is performed. The Senti-Bank method used to retrieve the mid-level characteristics and the image tag features, representing the visual features are integrated with the textual features to carry out the consistency based sentiment analysis. To utilize the internal correlation and the discriminative features between semantic and visual contents, an image text sentiment analysis model, Deep Multi-modal Attentive Fusion (DMAF) [1] is presented for sentiment analysis with a mixed fusion architecture. The image feature space is regarded jointly in [3]. Then, the textual and social features are plotted into a joint feature space using the genetic algorithm. Finally, to detect the semantic link between the text and image, SVM model is utilised. The cross-modal consistent regression model constructed in [6] is trained on the textual and visual features to learn the sentiments. Then, to learn textual characteristics, a distributed paragraph vector model is trained on the related titles and descriptions of the pictures. In [2] an image sentiment detection model is constructed with Convolutional Neural Networks to analyse the sentiment for the inconsistent image-text. To execute transfer learning, this model, in particular, is pre-trained on a huge dataset for object identification. Madhuri and Ganesh Kumar [4] presented a statistical approach for extractive text summarization on a document. Weights are assigned to sentences to rank them. The document's highly scored sentences are extracted, resulting in a better and concise summary of the document. Ullah and Islam [5] presented a method for generating

an extracted summary that takes into account the semantic linkages between the document's phrases.

All known multi-modal sentiment analysis algorithms essentially aggregate sentiments from multiple modalities, resulting in poor classification of sentiments. The consistency between image and text modalities is not taken into consideration. It does not bridge the gap between the visual and textual part from multi-modal tweets. Also, overall public opinions are not summarised after performing the sentiment analysis.

3 System Architecture

The general architecture diagram for the proposed approach is depicted in Fig. 1. The proposed approach performs consistency driven adaptive sentiment analysis on the multi-modal tweet using the CBSC Algorithm incontrast to the existing approaches. In addtion to the sentiment analysis of the tweets, our proposed approach performs opinion mining from the classified tweets, where the overall opinions are summarised by constructing a graph followed by ranking the tweets based on page rank algorithm.

3.1 Text Preprocessing

The text preprocessing includes removal of URLs, Twitter handles, digits, extra spaces, expressions pertaining to the start of the tweet, and replacement of shorthand included in the tweets with its abbreviated forms are all part of the text preprocessing. Finally, the tweet's characters are all lowercased. Once the tweets are cleaned, the stop words are removed and tokenization is carried out to convert the tweets into tokens. Lastly, lemmatization is done to group together the different inflected forms of a word.

3.2 Textual Feature Extraction

Word2vec is used for textual feature extraction. It improves the feature quality by considering contextual semantics of words in a text. The tweets are converted into vectors using Google-News-vectors-negative 300. The word addition method is used to represent the full tweet in vectors once each word has been converted into 300-D vectors. In this method, the addition of each dimension is calculated to get 300-D vectors as a text representation, which may be represented as

$$5[v_1, v_2, v_3, \ldots, v_{300}] = \sum_{i=1}^{n} \frac{[v_1, v_2, v_3, \ldots, v_{300}]}{n} \tag{1}$$

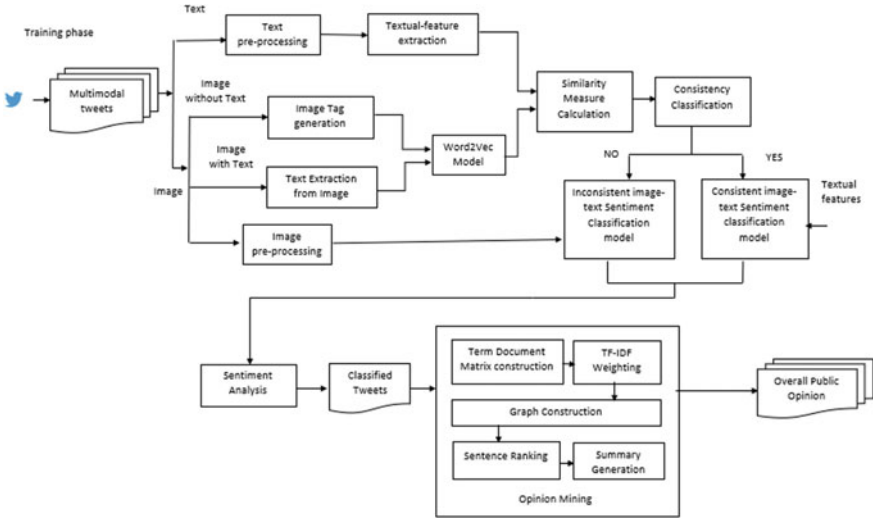


Fig. 1 System architecture

In addition to this, the tf-idf features are also generated. The Unigrams and Bigrams are extracted for SVM based sentiment analysis. The feature vector has a positive integer at indices of unigrams (and bigrams) which denotes the frequency of that particular unigram (or bigram) in the tweet and 0 elsewhere. To assign greater values to significant words, a matrix of such term-frequency vectors is generated, and each tf is scaled by the term's inverse-document-frequency (idf). The term's inverse-document-frequency is expressed as

$$\text{idf}(t) = \log\left(\frac{(1+n_d)}{1+d_f(d,t)}\right) + 1 \tag{2}$$

where n_d represents the total count of documents and $d_f(d, t)$ is the count of documents in which the term t occurs. So the TF-IDF is expressed as

$$\text{tf idf}(t, d, D) = \text{tf}(t, d).\text{idf}(t, D) \tag{3}$$

3.3 Image to Tag Generation

Image annotation playing an important role in content-based image understanding, is capable of describing an image with a set of tags. The sample of tags generated for the given image is shown in Fig. 2.

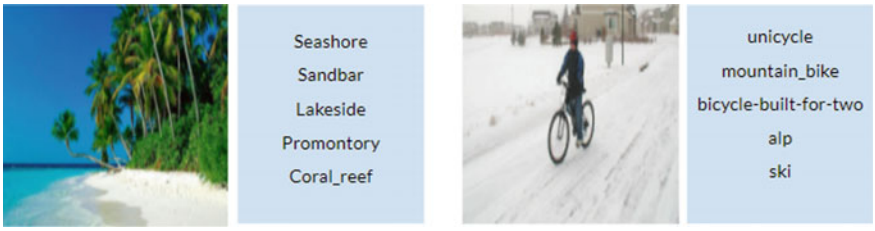


Fig. 2 Sample output of the tags generated

To extract the top tags from individual pictures, we utilised three previously trained CNN models. VGG-16, Inception V3, and Resnet, which are image classification models, were employed. The image-net database has pre trained these models. The tags are then sent to the w2v model, which generates 300-D vectors that are used in later stages.

3.4 Text Extraction from Image

In general, the text contents present in the images are detected using Optical Character Recognition (OCR) method. Text detection, text extraction, and text recognition are the three stages involved in retrieving text from image. Text detection seeks to separate text from background pixels, or to extract text pixels from background pixels. The input image is first pre-processed where processes like grayscale conversion, binary segmentation, contour detection takes place. From the preprocessed image, the text is extracted by passing through an OCR for recognition.

3.5 Consistent Image-Text Sentiment Classification

Initially the consistency between the given image and the text is calculated. For this purpose, the vectorised image tags generated and the vectorised textual tokens are mapped in the common space and checked for their consistency. If the consistency measure value is greater than the threshold i.e. 0.5, the image is consistent with the text. On the other side, if the value is lower than 0.5, the image is said to be inconsistent with the text. The method for performing consistent based sentiment analysis is shown in the Algorithm 1.

Once the image is consistent with the text, it is then fed to the consistent sentiment classification model for determining the sentiment of the tweet (see Fig. 4). Here, a SVM classifier is used to train the model. For such consistent image and text, the sentiment is determined from the textual part as both image and the text convey the same sentiment. Hence, the textual features extracted i.e., the Tf-idf vectors are fed

Algorithm 1 Consistency Based Sentiment Classification (CBSC)

```
Input: Textual tweet, Image tags
Output: Sentiment Polarity of the tweets
procedure SENTIMENT_ANALYSIS(text,image_tags)
    model=word2vec.load()
    text_vec=model(text)
    tag_vec=model(image_tags)
    consistency_val=model.similarity(text_vec,tag_vec)
    if consistency_val>threshold then
        feature_vec=apply_tfidf(preprocessed_tweets)
        classifier=load(consistent_sentiment_classification_model)
        sentiment=classifier.predict(feature_vec)
    else
        pre_processed_img=preprocess(image)
        model=load(inconsistent_sentiment_classification_model)
        sentiment=model.predict(pre_processed_img)
    end if
    Print the sentiment
end procedure
```

as input to the classifier. The model gets trained with the help of these features and finally predicts the polarity of the tweet i.e. either 1(positive) or 0 (negative).

3.6 Inconsistent Image-Text Sentiment Classification

If the image tag features and text features are inconsistent, only the image is considered for the sentiment classification. Here, DenseNet architecture consisting of a chain of Dense Block and transition layers is used. It has 121 layers with each layer connected to one another and ensures maximum information/gradient flow between them. Initially, all the images are fitted to size $224 \times 224 \times 3$ before being fed into the DenseNet model. All image pixel intensity values are normalized to be in the range 0 to 1.

Once preprocessed, a DenseNet121 model with the output layer removed and replaced by the following set of layers namely Flatten layer, 30% dropout layer, Dense layer, Output SoftMax layer is used. The neural net outputs confidence scores at the output layer for each of the sentiment polarities and the polarity with highest score is considered as the output prediction. Here, the sentiment is obtained from the image irrespective of the textual tweet since textual tweets are of smaller length (only 280 chars). The sample output is shown in Fig. 5.

3.7 Opinion Mining

Opinion mining is the technique of determining people's opinion based on reviews. It entails categorising the opinion expressed by the user as positive or negative polarity, followed by opinion summarization using Algorithm 2, which is the process of presenting review data in a concise and summary style.

Algorithm 2 Opinion Mining

```
Input Classified tweets(positive and negative tweets)
Output Summarised opinion of tweets
procedure GENERATE_SUMMARY(files)
    infiles = get_infiles(files)
    sentence_list = tokenise_document(infiles)
    cv=countvectorizer()
    tdm=cv.fit_transform(sentence_list)
    norm_tdm=tfidf().fit_transform(tdm)
    g=construct_graph(norm_tdm)
    ranks = nx.pagerank(g)
    sorted_sentences=sort(ranks)
    write_output(sorted_sentences)
end procedure
positive_opinions=generate_summary(positive_tweets)
negative_opinions=generate_summary(negative_tweets)
```

The tweets are converted into a term document matrix which is a collection of all words in the given document. The TDM splits each term in the document and assigns weightage to each word according to TF-IDF weights. Then, a graph is constructed to proceed to the text ranking algorithm as in Fig. 3. In the graph, each node represents a sentence and each word represents the words that they have in common. The weight of the edges is the number of words they have in common between the two nodes. The sentences are then ranked based on the page rank algorithm. Thus, by spotting the important sentences, the summarised opinions of positive and the negative tweets are generated.

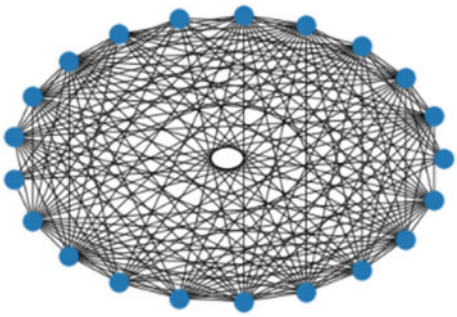


Fig. 3 Output of the graph constructed



Fig. 4 Sample output of the consistent image-text sentiment analysis phase

4 Experimental Results

4.1 Dataset

Our approach tackles the problem of multi-modal sentiment analysis, using both textual and image information. For this, we used the B-T4SA dataset, which is a dataset consisting of Twitter information, in which every sample has both text and image.

4.2 Performance Metrics

For the Image Text Sentiment Analysis, the polarity of tweets may be predicted correctly or may not be predicted correctly. So, the evaluation of this process is carried out using the measures namely Accuracy, Recall, Precision, F1-score. Performance evaluation of all the sentiment classification models used is referred in Table 1.

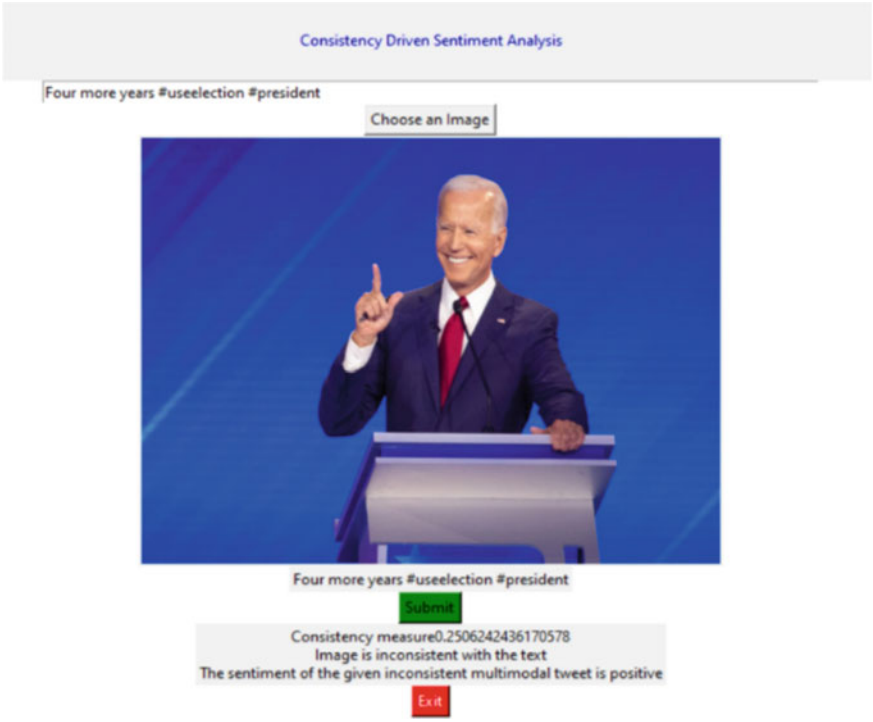


Fig. 5 Sample output of the inconsistent image-text sentiment analysis phase

Accuracy: It is the number of correctly predicted sentiments divided by the total number sentiment predictions.

$$\text{Accuracy} = \frac{\text{True_Positive} + \text{True_Negative}}{\text{All_Samples}} \tag{4}$$

Precision: The percentage of tweets that are correctly predicted as positive to that of the total number of positively classified tweets.

$$\text{Precision} = \frac{\text{True_Positive}}{\text{True_Positive} + \text{False_Positive}} \tag{5}$$

Recall: The percentage of tweets accurately predicted as positive to the total positively classified tweets.

$$\text{Recall} = \frac{\text{True_Positive}}{\text{True_Positive} + \text{False_Negative}} \tag{6}$$

Table 1 Performance evaluation of the models (values in percentage)

Model	Precision	Recall	F-score	Accuracy
Consistent image-text sentiment classification model	90	80	85	84
Inconsistent image-text sentiment classification model	64	68	65	64

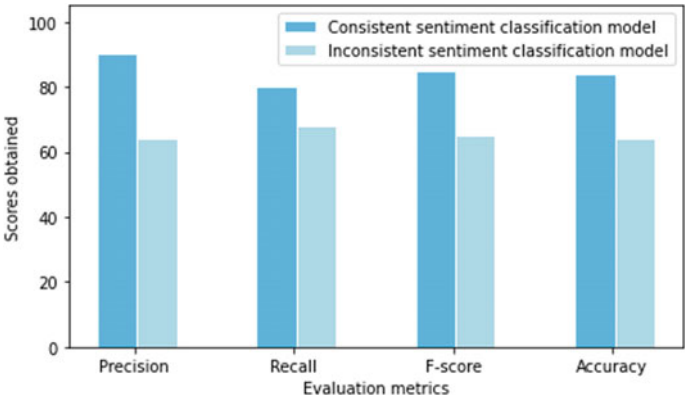


Fig. 6 Result analysis of sentiment classification models

F1-score: The F1 score is defined as the weighted harmonic mean of the test's recall and precision.

$$F1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

The values obtained on evaluating the consistent image-text sentiment analysis and inconsistent image-text sentiment analysis models are analysed in the Fig. 6.

Performance of the summarised opinions are measured using ROUGE 1 (Refer Table 2)

ROUGE 1: ROUGE-1 alludes to the unigram overlap amidst the summary generated by the system and the reference summary.

P—the amount of words that overlap in the system, as well as a summary of the references.
Q—the amount of words in the summary generated by the system.
R—the amount of words in the reference summary.

Recall: The amount of the reference summary that is recovered or captured is measured by Recall.

$$\text{Recall} = \frac{P}{R} \tag{8}$$

Table 2 Performance evaluation of opinion mining

	ROUGE-1-P	ROUGE-1-R	ROUGE-1
Opinion mining	0.750	0.462	0.571

Precision: Precision is defined as the amount of the system summary that was useful or relevant.

$$\text{Precision} = \frac{P}{Q} \tag{9}$$

5 Conclusion

Thus the proposed consistency driven cross modal twitter sentiment analysis and opinion mining has been implemented. The proposed approach is able to analyse the multi-modal tweets and summarise them in an accurate manner. The system performs consistency based adaptive sentiment analysis using consistency based sentiment classification model(CBSC). In addition to this, the overall opinions are summarised. The proposed work would provide ample room for future work. The obtained accuracy can be further increased using different approaches for different sentiment classification modules.

References

1. Huang F, Zhang X, Zhao Z, Xu J, Li Z (2019) Image-text sentiment analysis via deep multi-modal attentive fusion. Knowl-Based Syst, Elsevier 167:26–37. https://doi.org/10.1016/j.knosys.2019.01.019
2. Jindal S, Singh S (2015) Image sentiment analysis using deep convolutional neural networks with domain specific fine tuning. In: International conference on information processing (ICIP). Pune, India. https://doi.org/10.1109/infop.2015.7489424
3. Liu M, Zhang L, Liu Y, Hu H, Fang W (2017) Recognizing semantic correlation in image-text weibo via feature space mapping. Comput Vis Image Understand 163:58–66. https://doi.org/10.1016/j.cviu.2017.04.012
4. Madhuri JN, Ganesh Kumar R (2019) Extractive text summarization using sentence ranking. In: International conference on data science and communication (IconDSC). Bangalore, India. https://doi.org/10.1109/IconDSC.2019.8817040
5. Ullah S, Islam ABMAA (2019) A framework for extractive text summarization using semantic graph based approach. In: Proceedings of the 6th international conference on networking, systems and security, pp 48–56. https://doi.org/10.1145/3362966.3362971
6. You Q, Luo J, Jin H, Yang J (2016) Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In: Proceedings of the ninth ACM international conference on web search and data mining (WSDM), pp 13–22. https://doi.org/10.1145/2835776.2835779

7. Zhao Z, Zhu H, Xue Z, Liu Z, Tian J, Chua MCH, Liu M (2019) An image-text consistency driven multi-modal sentiment analysis approach for social media. In: Information processing and management, 56(6). https://doi.org/10.1016/j.ipm.2019.102097

A Blockchain-Based Document Verification System for Employers



Kshitij Dhyani, Jahnavi Mishra, Subhra Paladhi, and I. Sumaiya Thaseen

Abstract Academic institutions often maintain records and documents of the students enrolled within the institute. These records are greatly regarded by employers and often verified before giving employment offers. While all the other sectors in academia are moving towards automation, this specific use case is still quite primitive in its functioning. Even when this is a regular activity undergone by academic institutes, methodologies with vulnerabilities and reasonable overhead are employed. There is a need for a tamper-proof, reliable, and faster mode of document verification for employers which can be used on the go with complete reassurance for the authenticity of the provided documents. This paper puts forward a blockchain-based platform that allows academic institutes to offer a secure, dependable, cost-effective, and scalable verification of documents via a Web interface. The platform is built upon the logic written in solidity and utilizes the ethereum blockchain to enforce immutability of the document, and the backend is linked to the user interface using the web3.js library. Additionally, the platform offers a statistical comparison of these records to other students and assists the employer to assess the performance of the student being recruited.

Keywords Blockchain · Ethereum · Record verification

1 Introduction

Most processes in the modern world have started using automation and digital architectures. This has improved efficiency and reduced the cost and time for these processes. Employers require academic institutions to share documents for verification purposes when a student is joining their organization after being recruited on the campus of the institution. The employers always verified due to the possibility of false documents being presented by the students. Hence, as an authenticity check, they ask the academic institution for the record of the hired student. In most cases,

K. Dhyani (✉) · J. Mishra · S. Paladhi · I. Sumaiya Thaseen
School of Information Technology and Engineering, Vellore Institute of Technology (VIT), TamilNadu, Vellore 632014, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_11

the transfer of academic documents and records is done using primitive methods like postal mail and e-mails, and then, they need to be verified manually. Postal mails require a person to physically send and receive the mail. This is followed by a manual verification procedure. This makes postal mails a time taking, costly, and less reliable method for such important tasks. E-mails require a person to physically scan the documents, send the e-mail, and it depends on the person on the other side to check the documents and verify them. It is prone to problems like entering the wrong e-mail address, e-mail going into the spam folder, or sender attaching wrong attachments.

Considering the drawbacks of the above-mentioned methods, machine learning algorithms like SVM have been used for identifying whether the documents are original or not [1]. Also, various institutions have designed Web interfaces where the documents are uploaded by the concerned authorities, verified by the system using digital signatures, or QR code scanners, or using hashing algorithms. These systems comprise external universities, recruiters, students, and the current universities who play their respective roles for the smooth functioning of the system.

These approaches provide some benefits over the primitive method of postal mail and e-mails but still have their limitations. For few models, the type of documents to be uploaded was limited to PDF format only, and other text documents and image-based documents were not acceptable [2]. The main issue with QR code scanners was that it is not cost-effective and faking QR codes and digital signatures have a high possibility [3]. Hacking and data getting deleted from DB are also possible when it comes to Web interfaces.

To resolve many of these limitations a blockchain-based approach can be used. Blockchain is a decentralized, immutable digital ledger that can be used to store information. Being an immutable ledger, once a verified document is put into the blockchain, the trustworthiness of the document being genuine is guaranteed. A blockchain-based approach provides full immunity from situations such as a student's record being deleted by error or someone hacking into a database to change records. The information of the students will be available to employers on-demand over the blockchain, and they can be sure about the data being un-tampered and verified by the academic institution. Most of the steps in this process can be automated and will not require the interference of a human. The purpose of our proposed system is to provide a user-friendly, fast, reliable, and secure system for the verification of academic records.

The structure of the rest of the paper is as follows: Sect. 2 is about related work, Sect. 3 is the background which explains the working of blockchain, in Sect. 4, we present our proposed solution, Sect. 5 is about evaluation and results of the experiments, and the finally, Sect. 6 presents the conclusion and expected future work.

2 Related Work

There have been various possible solutions for the transfer and the validation of academic records and self-sovereign documents which have been adopted by various institutions and individuals worldwide for various purposes varying from academia to health care to legal deeds.

Garain and Halder [1] proposed a model which uses both genuine and duplicate bank cheques as the reference which are scanned into colour images, and security features are then extracted to generate feature vectors. A classification model is designed using SVMs which is trained using a set of labelled samples, and a testing set is used to determine accuracy. The analysis of error cases is not discussed in the paper. Also, this approach is limited to bank cheques only and needs to be extended to lottery tickets, legal documents, etc.

Brdesee [4] from King Abdulaziz University proposed a unified e-portal for the university to standardize the verification of all the documents of the students when the recruiters and other universities demand it. The recruiters, external universities, and students need to log in and then create a verification request or trace the previous request. They also get to choose the type of document to be verified and the means by which they wish to receive the verification.

IDStack provides a platform [2] where the documents provided by the owners are digitized by the extractor and converted into a machine-readable document using IDStack stating that the provided documents and the digitized document are 100% matching. Each machine-readable document is then digitally signed by the validators. Those signed documents are then viewed by relying upon third parties and confidence, and correlation scores are provided to the document. One major issue with digital signatures is that these documents are self-signed and there is no way to verify the identity of the signer. Another drawback is that the model only accepts PDF documents and not other text documents and image-based documents.

A possible update to this technique would be supporting image processing and using OCR to extract data from image files.

Putro and Luthfi [3] provide a system to produce authentic and safe documents with a combination of perceptual hash and OCR. Public and private keys are determined to be used in both, document creation and verification phases. SHA-256 is used to hash the contents in the document, and a digital signature is created using hash value and keys. This along with perceptual hash and OCR hash is used to form QR code which will be inserted in the document to be printed. The QR code reader will read the QR code given in the document and test its authenticity and integrity. The major issue with this scheme is that devices used to scan QR codes are costly and faking a QR code is also a possibility nowadays.

The above methods are very well tested and are quite efficient for mainly document authentication and transfer if required, while blockchain-based approaches provide extra competence to achieve immutability through its distributed continuous ledger architecture.

The MIT Digital Credentials Consortium [5] provides a model where the learner can register their digital documents to a bitcoin blockchain. The learner is issued digital credentials which they can associate with various self-sovereign IDs. The credentials are under the learner's control, and the decision of sharing or exchanging them depends on him/her. These credentials once shared with the parties are verified, and the status of verification is returned in the form of success or failure with the reason behind the failure. Few drawbacks of bitcoin blockchain include very high processing time, immutability, harder to scale, and high energy consumption.

The ShoCard app [6] which is designed to prove one's identity rather than verify someone's identity, uses blockchain technology as a public, unchangeable ledger that allows third parties to verify that the original data or certifications have not been tampered with. It provides a wallet to store all validated identities such as driver's license and school transcripts. Every user will be provided with a credit score, and based on certain conditions, the identities will be validated. Temporary access to the private sections of blockchain can be provided to few organizations for validation.

EduCTX [7] which is a blockchain-based higher education credit and grading platform provides an approach that aims to form single merit for judgement for all students/employees. It is based on a P2P network system. Home universities referred to as HEI are also a part of the blockchain network. When the student enrols in HEI, a student ID is issued, and a new blockchain address is generated. At the point when an organization needs to confirm the students' credit record and course completion, the student needs to send his/her blockchain address, 2–2 multi-signature blockchain deliver and reclaim content to the verifier—organization. Utilizing the blockchain Web API to get to blockchain information, the organization checks the measure of ECTX tokens in the 2–2 multi-signature address, which addresses the students' credit count.

Permissioned blockchain-based system [8] is an approach for the verification of academic records. It consists of a Web interface for registering and demanding the transfer of academic records with a backend utilizing hyper ledger fabric and hyper ledger composer to preserve the hash of the archives on the blockchain for validation. The usage of blockchains such as ethereum and bitcoin function on open networks is therefore not trustworthy and is unsafe. This calls for the introduction of permissioned blockchain such as hyper ledger with functions on a private network, and the users can be identified and hence make the functioning safe and trustworthy. As compared to the other approaches discussed above, this model provides flexibility to upload documents in any format, and it provides an automated solution that is more scalable to assist the load of possible users and institutions.

3 Background

In this section, we talk in some detail about blockchain and the tools that will be required to create the blockchain for our proposed model.

3.1 Blockchain

Blockchain was first presented in the famous bitcoin whitepaper by an anonymous person with the pseudonym Satoshi Nakamoto. The paper proposed to use blockchain technology to create bitcoin which is a decentralized, permissionless cryptocurrency [9]. A trust-based system has a central authority that verifies the transactions which is a bottleneck in the whole system. If the central authority gets corrupted or goes down, then the whole system goes bust. Blockchain being a decentralized system does not have to deal with such problems. Blockchain is suitable for usage in a system that requires distributed data storage with immutability [10].

In a blockchain, each block has its own unique hash, the previous block's hash, the sender's, and the receiver's encrypted identifier. If something is changed in the block's data, then the hash value is also changed [9].

3.2 Proof of Work

For the creation of a block, proof of work is required. To get the proof of work, a hash needs to be calculated which has the required number of zeros at the beginning of the hash value. For generating the hash, a nonce is used which is incremented until the required hash is generated. The proof-of-work calculation is a one-CPU-one vote system [9].

3.3 Immutability

Immutability is essential in a blockchain as it provides a guarantee that the information in the chain is valid. To obtain immutability, blockchain depends on a computationally hard problem. The nodes consider the longest chain to be the correct chain and perform all transactions on that. To change one block on the chain, the work for that block and for all the blocks in the chain after that needs to be computed, and this new chain needs to outpace the original chain. Because of this, once the chain becomes long enough, it is almost impossible to change any block on the chain, hence making the chain immutable.

3.4 Ethereum

Ethereum is a cryptocurrency that was developed by Vitalik Buterin and launched in 2015 [11]. The ethereum chain allows to run many different instructions and not just send and receive transactions like on bitcoin. This allows the development of more

complex applications. For all types of transactions on the ethereum chain, the Ether (Ether) cryptocurrency is required [12].

3.5 Smart Contracts

Smart contracts are computer protocols that verify and enforce a contract between multiple parties on a blockchain. Ethereum was the first blockchain to implement complex smart contracts. Ethereum has ethereum virtual machine (EVM) to create run-time for executing smart contracts. A transaction can trigger a contract, and the transaction is successful if it is a valid transaction according to the contract [13].

3.6 DApps

DApps stand for decentralized applications. These are applications that are built over a blockchain and use the blockchain for its data storage and processing requirements. This kind of application is implemented using smart contracts. Ethereum is the most widely used blockchain to build DApps [12].

3.7 Metamask

Metamask provides a secure way to connect to Web applications based on ethereum and several test networks like Rinkby, Georli, etc. It manages the account and connects the user to the blockchain [14].

3.8 Rinkby

Rinkby is an Ethereum test net. Test nets are used to test blockchain applications. The currency used on the test nets is valueless [15].

3.9 Remix

Remix IDE is an open-source Web and desktop application that is used for building DApps on ethereum [16].

3.9.1 Solidity

Solidity is an object-oriented, high-level language that is used for implementing smart contracts. Solidity is influenced by C++, Python, and JavaScript programming languages [17].

3.9.2 Web3 JS

web3.js is a collection of libraries that are used to interact with a local or remote ethereum node using HTTP, IPC, or WebSocket [18].

4 Proposed Solution

This section attempts to explain the proposed solution of academic and employment document verification systems powered by blockchain. To best grasp the overview of the proposed technology, we will start by understanding the underlying assumptions which were axiomatically assumed before building the platform, followed by the description of the architecture of the system. Then, the implementation process is discussed along with the scope of scalability. Finally, some challenges and solutions are discussed which were encountered during the development life cycle.

4.1 Assumptions and Scope

Some non-trivial assumptions are to be made, for the proper functioning of the platform.

1. It is assumed that the documents, transcripts, student information and credentials, history of academic performance, and other such details are provided exhaustively by the participating university/entity/organization.
2. It is assumed that the documents and details provided by the participating organization are accurate and true. The integrity of the details provided by the participating organization during the system initialization is attested by the organization itself.
3. The records and details provided to the system are valid for the indefinite foreseeable future unless particularly specified by the participating organization, in which case it will be explicitly mentioned on the interface.

The proposed technology offers a comprehensive solution to documentation and detailed verification for academic institutes and employment houses, provided that the above assumptions are met.

4.2 Architecture

The system architecture can be broken down into three main modules. Namely the (1), smart contract, the actual (2) ethereum private blockchain network, and finally the (3) Web interface module. In the rest of this subsection, we will start to discuss each module, for the best comprehension.

Ethereum smart contracts are a methodology developed and provided by the ethereum blockchain community, which allows any blockchain idea to become highly scalable. This is achieved by introducing a programming language called solidity [19], where the logic to be followed by the blockchain is specified. With this algorithmic set of rules written in solidity, the associated ethereum blockchain network knows exactly how to function. The ethereum community additionally offers an integrated development environment (IDE) called the Remix for the solidity programming language, which offers features such as solidity linting, syntactical checking, and easy compilation of the code. For defining the data within the blockchain, structs data types are most popularly used. For our proposed technology, the logic for the document verification system is written in solidity, where the struct data type defines the attributes to be stored for the student, all within the Remix IDE.

Ethereum blockchain network is the second layer of the proposed architecture, and it is essentially a network of nodes/participating devices working together to validate transactions and maintain a ledger based on the norms of the ethereum community. There are multiple ethereum blockchain networks within the community, for different purposes. This blockchain network is where the data of the students will be stored once the smart contract is deployed. Once the contract has been deployed and the data has been pushed and validated by the network, it fundamentally becomes tamper-proof. The deployment is done using the Remix IDE.

The last and the topmost layer of the proposed architecture is the Web interface, where the employer or academic authorizes can come to verify information and documents. This Web interface is connected to the blockchain using a JavaScript framework web3.js. This Web interface abstracts and encapsulates the other two layers and makes them invisible to the user (see Fig. 1). The user simply searches the desired candidate's information and verifies it.

4.3 Implementation

As discussed in the previous section, the proposed technology was implemented through three modules. This section will elaborate more on each of these three modules and their implementation in real time.

Smart Contract Module
The smart contract is written in the solidity programming language, which is considered to be a mild representation of C++ and JavaScript. The smart contract is essentially the logic behind the functioning of the blockchain. For our particular use case,

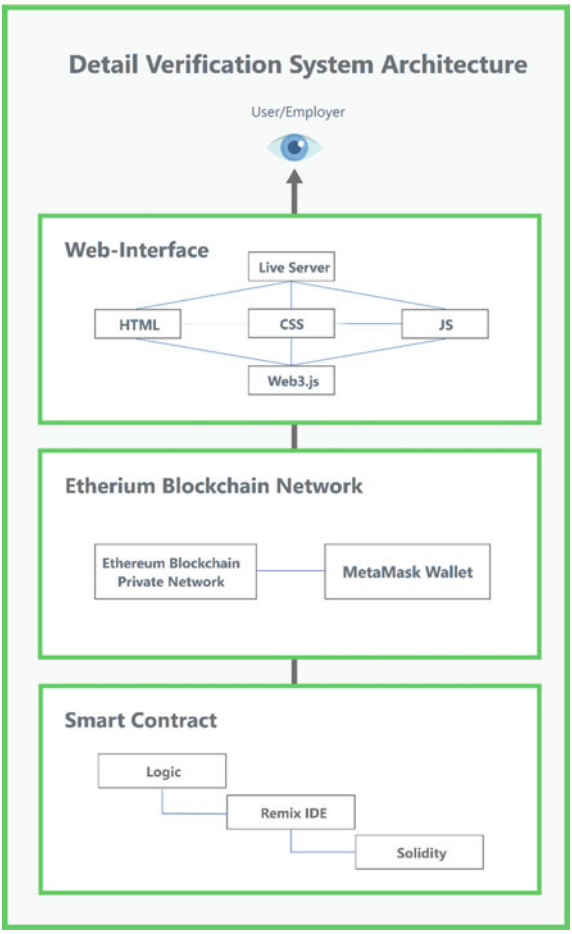


Fig. 1 System architecture for the proposed technology

our smart contracts implement the logic behind document and detail verification. The solidity program a.k.a the smart contract defines functions within the code to perform functions such as creating a student and reading student details with functions createStudent() and readStudent(), respectively. The details within each student's block are determined by defining a struct, which is essentially a user-defined data type that helps represent complex real-world entities in the code easily. This contract is written within the Remix integrated development environment which facilitates solidity development. The smart contract for our proposed application is written in the solidity version 0.5.0 and is further augmented with abicoder v2 which allows us to use structs and dynamic variables along with functions. The code for the smart contract can be found and downloaded at https://github.com/wimpywarlord/ISM-PROJECT.git.

Ethereum Blockchain Network

The ethereum blockchain network used for the proposed technology is Rinkby private network [20]. This network is a typical ethereum network that utilizes the proof-of-work (PoW) algorithm to verify transactions to and hence limits the user from tampering with the data. In order to deploy the smart contract to the Rinkby network, the metamask cryptocurrency wallet is used to pay the gas money [21] (see Fig. 2).

To attain ether in the Rinkby network wallet, public faucets were used. Upon linking the metamask wallet with the Remix IDE, the contract is deployed on the network. With this, the blockchain is set up and additionally equipped with the smart contract which is the logic of functioning.

Web Interface

The Web interface is the topmost layer of the whole architecture. This is the only layer that is visible to the user, and other layers are of no significance for the user. The Web interface needs to be connected with the blockchain. This is done using the *JavaScript* framework *web3.js* [22], which is developed to enable interactions between ethereum networks and nodes over *HTTP/IPC/WebSocket* protocols. Additionally, the *application binary interface* (ABI) for the smart contract [23], along

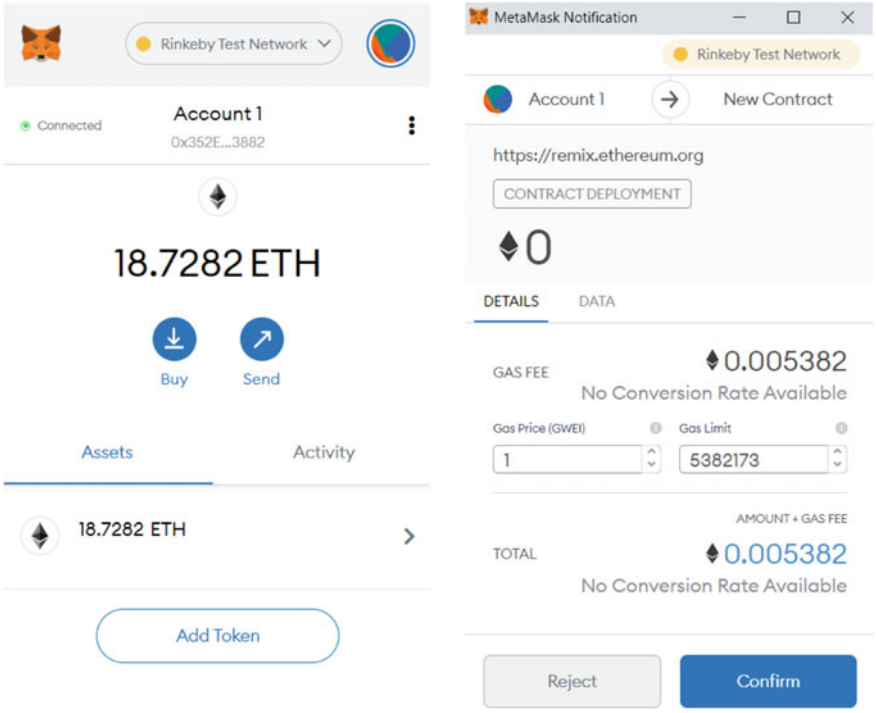


Fig. 2 Metamask cryptocurrency wallet being used for payment of gas fee while the deployment of smart contract on the ethereum Rinkby test network

with the deployed contract *to address,* is fed to the web3.js function, to perfectly establish the connection. Once connected the interface is structured using HTML, styled using CSS, and manipulated using JavaScript.

Scalability
The platform is built upon the ethereum blockchain, and hence, scaling is a very straightforward process. As per the current standard, the chain can store up to 13 TB of data. Additionally, if the data requirements increase, a simple enhancement of memory of the full nodes will vastly increase the ledger capacity. Moreover, the simple Web interface for the users allows them to use the service without being connected to the ethereum blockchain directly. Therefore, even in the state of storage issues, the platform is fully functional.

Challenges and Solutions
Currently, the ethereum blockchain can support only up to 30 transactions per second which acts as a bottleneck in some edge cases where rapid update queries in data are issued for multiple participating organizations at the same time. However, the ethereum 2.0 is expected to be released by the end of this year, which will push the threshold to around 100,000 transactions per second.

5 Experimental Evaluation and Results

The performance evaluation for the proposed platform was done using Google lighthouse. The device used for performance check is an emulated desktop, with network throttling of 40 ms TCP RTT, 10,240 Kbps throughput (simulated). User agent (host) is Mozilla/5.0 (Windows NT 10.0; Win64; $\times$64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36, and the user agent (network) is Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4143.7 Safari/537.36 Chrome-Lighthouse. The Web interface received an overall performance rating of 92. The website was assessed for the following metrics: First contentful Paint, Time to interactive, Total Blocking time, Cumulative Layout Shifts, and Speed Index, Largest Contentful Paint and got fast rating under 4 out of 6 metrics and moderate rating for 2 out 6 metrics (see Fig. 3).

In contest against the available alternatives, the proposed model provides greater set of features, proves to be more scalable and facilitates a great deal of automation as shown in Table 1.

The platform built offers a faster, more reliable, tamper-proof, and fraud-free solution to the document verification process of universities and employers. Once deployed, the data within the blockchain cannot be altered even by the university employees, therefore ensuring the authenticity of the displayed data. Moreover, the Web interface has a straightforward and intuitive user interface, and it offers single tap verification of data. The blockchain-powered architecture offers impeccable security, and this, in turn, builds integrity and trust for the documents within the system.



Fig. 3 Performance metrics for the Web interface

Table 1 Comparative analysis with alternative solutions present in the industry

	Blockchain	Record	Automated	Immutable	Scalable
SVM model	N/A	Bank cheques	Yes	No	No
King Ab-dulaziz University	N/A	Academic documents (any format)	Yes	No	Yes
IDStack	N/A	Only PDF documents	Yes	No	Yes
Hash and OCR	N/A	Physical documents	No	No	No
MIT DCC	Bitcoin	Any	No	Yes	No
ShoCard	Bitcoin	Any	Yes	Yes	Yes
EduCTX	ARK	Student credit record	Yes	Yes	Yes
Proposed solution	Ethereum	Any	Yes	Yes	Yes

This solution outperforms manual transmission of data and other alternatives, by all standards. Not only does it offer absolute security against counterfeit but also curtails wait time by a very significant amount.

The Web interface gives a thorough overview of a student's profile by giving information about the academic record, extracurricular record, industrial experience, research experience, and much more (see Fig. 4).

Additionally, the Web interface offers searching candidates based on their skill, which helps employers to shortlisted desirable students for their job role and moreover provides an analytical overview of the student's abilities while comparing it to the average of all the students in the university/organization (see Fig. 5).

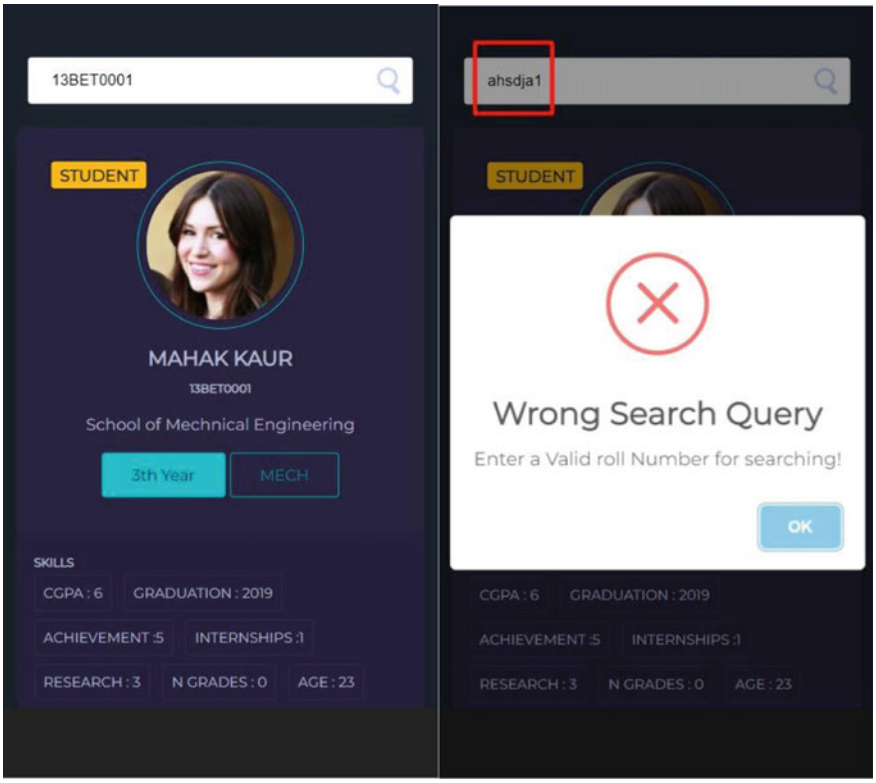


Fig. 4 Web interface displaying the student's information and detecting faulty queries

6 Conclusion and Future Work

This paper puts forward a platform that offers a secure and comprehensive alternative for employers to verify student data given by the university in a faster and more reliable way. The architecture is powered by an ethereum blockchain and is offered as a Web interface to the users build upon on JavaScript. The proposed platform has many verticals that could be exploited to produce a complete tool kit for academic institutions. The platform is being worked upon to offer charts and graphs as an analytical overview of the student's performance and aptitude. Moreover, features such as offering to store the hash of the official academic transcript for verification and authentication purposes could be looked upon further. Moreover, the performance of the Web interface can be further improved by using the industry best practices and removing certain unused libraries and styling in the code.

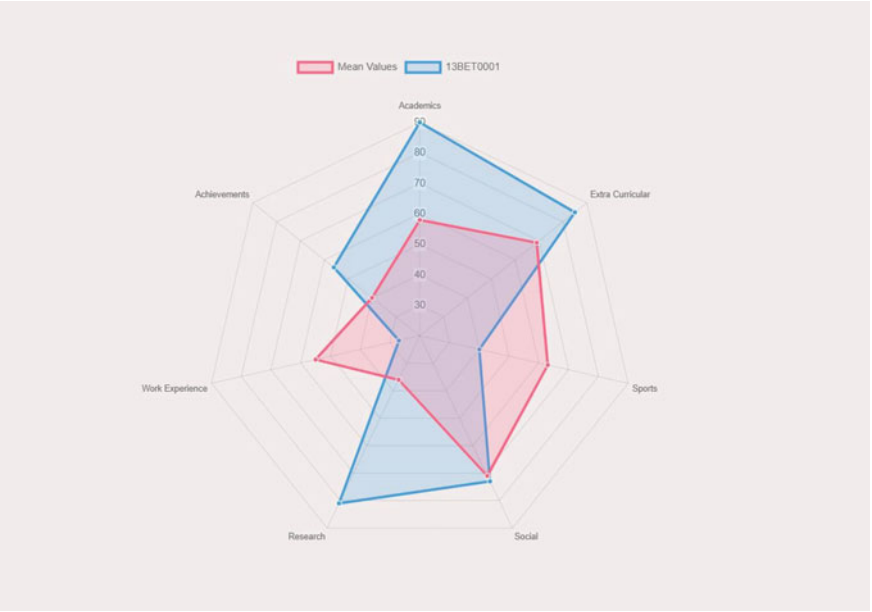


Fig. 5 Analytical overview of the student's performance

References

1. Garain U, Halder B (2008) On automatic authenticity verification of printed security documents. In: Sixth Indian conference on computer vision, graphics and image processing. IEEE, pp 706–713
2. Lakmal C, Dangalla S, Herath C, Wickramarathna C, Dias G, Fernando S (2017) IDStack—the common protocol for document verification built on digital signatures. In: National information technology conference (NITC). IEEE, pp 96–99
3. Putro PAW, Luthfi M (2019) An authentic and secure printed document from forgery attack by combining perceptual hash and optical character recognition. In: International conference on informatics, multimedia, cyber and information system (ICIMCIS). IEEE, pp 157–162
4. Brdesee HS (2019) An online verification system of students and graduates documents and certificates: a developed strategy that prevents fraud qualifications. Int J Smart Educ Urban Soc (IJSEUS) 10(2):1–18
5. MIT Media Lab Learning Initiative and Learning Machine Digital certificates project. [Online]. Available: http://certificates.media.mit.edu/
6. (2017) ShoCard|Identity for a Mobile World. In: Shocard.com [online]. Available: https://shocard.com/
7. Turkanović M, Hölbl M, Košič K, Heričko M, Kamišalić A (2018) EduCTX: a blockchain-based higher education credit platform. IEEE Access 6:5112–5127
8. Badr A, Rafferty L, Mahmoud QH, Elgazzar K, Hung PC (2019) A permissioned blockchain-based system for verification of academic records. In: 10th IFIP international conference on new technologies, mobility and security (NTMS). IEEE, pp 1–5
9. Nakamoto S (2019) Bitcoin: a peer-to-peer electronic cash system. Manubot
10. Chowdhury MJM, Colman A, Kabir MA, Han J, Sarda P (2018) Blockchain versus database: a critical analysis. In: 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 1348–1353
11. Buterin V (2017) Ethereum: a next generation smart contract and decentralized application platform (2013). http://ethereum.org/ethereum.html
12. Metcalfe W (2020) Ethereum, smart contracts, DApps. In: Blockchain and crypt currency. Springer, Singapore, pp 77–93
13. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang FY (2019) Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Trans Syst Man Cybern: Syst 49(11):2266–2277
14. ConsenSys (n.d.) Metamask wallet. MetaMask Docs. https://docs.metamask.io/guide/
15. Rinkeby Ethereum Test Network. AirSwap Support (n.d.) https://support.airswap.io/en/articles/2831385-what-is-rinkeby
16. Remix (n.d.) Remix—ethereum IDE. Remix. https://remix-ide.readthedocs.io/en/latest/
17. Wood G (n.d.) Solidity. https://docs.soliditylang.org/en/v0.8.3/
18. web3.js—ethereum JavaScript API—web3.js 1.0.0 documentation (n.d.) https://web3js.readthedocs.io/en/v1.3.4/
19. Dannen C (2017) Introducing ethereum and solidity, vol 318. Apress, Berkeley
20. Iyer K, Dannen C (2018) The ethereum development environment. In: Building games with ethereum smart contracts. Apress, Berkeley, pp 19–36
21. Lee WM (2019) Using the metamask chrome extension. In: Beginning ethereum smart contracts programming. Apress, Berkeley, pp 93–126
22. Lee WM (2019) Using the web3. js APIs. In: Beginning ethereum smart contracts programming. Apress, Berkeley, pp 169–198
23. Zheng G, Gao L, Huang L, Guan J (2021) Application binary interface (ABI). In: Ethereum smart contract development in solidity. Springer, Singapore, pp 139–158

Railroad Segmentation Using Modified UNet



R. S. Rampriya, Sabarinathan, R. Suganya, S. Babysruthi, Pitchuka Yamini, and Pernamitta Hanuma Reddy

Abstract One of the complex activities in our daily life humans do is driving. We humans handle unexpected situations like reacting to weather conditions, making decisions against the rules which make us save human life who are in danger. Thus, in this situation, we are approaching a technique artificial intelligence which is the actively researched area in autonomous driving. To succeed in this field, they will be using hardware and software components that support this function. The components like LIDAR, cameras, sensors and computer vision algorithms are need to be implemented more closely which help us to achieve autonomous driving. Generally, nowadays, in many places, trains lead to autonomous driving by detecting the rail path which means there is no any interaction of humans in driving. To detect the path, we had used encoder and decoder part, where in each part, there consists of few levels, where in each level, we had used CNN layers like convolution layer, pooling layer and convolution transpose layer. We have collected the dataset manually from Web, labeled using Photoshop and masked using MATLAB. We had compared this model with existing methods where the accuracy of our model is 92%, and segmentation is very good in our case. Thus, we have concluded that modified UNet provides better accuracy compared to all other methods.

Keywords Railroad detection · UNet · Segmentation · CNN · Convolution · Pooling · Transpose convolution

R. S. Rampriya (✉) · S. Babysruthi · P. Yamini · P. Hanuma Reddy
Department of CT, Anna University (MIT Campus), Chennai, TamilNadu, India

Sabarinathan
Couger Inc., Tokyo, Japan

R. Suganya
Department of IT, Thiagarajar College of Engineering, Madurai, TamilNadu, India
e-mail: rsuganya@tce.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_12

1 Introduction

Generally, there are lot of benefits for automated transport solutions for both developing and developed countries like lowering of operating costs, flexibility and higher safety standards. Many applications have been upgraded into autonomous, for instance: drones, self-driving cars, etc. At present condition, there can be reduction in human interaction in transport system. Nowadays, there are more suicides taking place, and thus, it is required to detect the obstacle where there will be setup of camera, takes snapshots and analyse it.

First there was an automation application known as elevators. When it was invented, every elevator had their own operator, how the trains have the operators. Even though it was invented, people refused to use elevators without an operator for around 50 years. As the same, after elevators, the next logical step is for autonomous trains. There are 64 metro trains operated autonomously all around the world in 2019.

Low-speed fully autonomous trains can become a really necessary means of transportation within the future. At present, the detection of obstacles of railroad primarily depends on trackside instrumentality and manual examination. The goal of the detection is to completely understand the atmosphere before the train. A crucial preprocessing step of atmosphere perception is to spot and phase the railroad region because it permits the train to spot the driving space so it will do effective obstacle detection. As such, activity correct railroad detection may be a key enabler of absolutely autonomous trains.

Artificial intelligence is an active research area in autonomous driving. To succeed in this field, they will be using hardware and software components that support this function. The components like LIDAR, cameras, sensors and the computer vision algorithms are need to be implemented closely which help us to achieve autonomous driving. The component "sensors" in autonomous driving are like our eyes which can identify the environment fast and accurate. Thus, based on this, we can understand that we can succeed in autonomous driving by well-functioning image recognition. This task can be done using DNN.

Thus, this system can be done based on camera-based components. Semantic segmentation is the core element in this field. Semantic segmentation is an image processing step which separates elements in an image according to their type and defines the boundaries.

2 Related Works

Nowadays, autonomous train driving is in the top place in the field of remote sensing image analysis. Thus, we were aiming on autonomous rails. We had referred many papers on that field and identified some of the pros and cons based on their models. There are lot of deep learning methods which helps us to achieve autonomous driving. Some of the references to our project were mentioned in Table 1:

Table 1 Comparison of different existing models

Related work	Year	Objective	Merits	Demerits
[1]	2009	To compute the optimal path using dynamic programming that gives the minimum cost to extract railroad track space	Dynamic programming gave the minimum cost to extract railroad track space	Negative effects of the camera movement when train is running at high speeds
[2]	2014	To develop very deep convolutional networks for large-scale image recognition	Proposed technique is used to train CNN efficiently for extracting multiple objects simultaneously	Computational complexity increases as training data increases
[3]	2015	Semantic segmentation of railway track images with deep convolutional neural networks (DCNN)	DCNN trained end-to-end for material classification, and it is more accurate	Not satisfied for long-range dependencies among image regions
[4]	2016	To detect obstacles by comparing input and reference train frontal view camera images	Obstacles are detected by applying image subtraction	Results vary during light conditions
[5]	2017	To develop deep convolutional encoder–decoder architecture for image segmentation	Proposed technique segments track effectively using encoder–decoder network	As number of layers increases, complexity increases
[6]	2018	Road extraction using deep residual UNet	Better performance by using combination of residual network and UNet	Segmentation results are poor
[7]	2018	Segmentation from filtering networks	Various kind of filtering approaches lead to extract more information	Approaches are leading to more or less satisfying results
[8]	2018	To understand the environment around the car through the use of the various sensors and the control modules	Approach can sense the car's environment and control modules	Poor detection in various conditions

(continued)

Table 1 (continued)

Related work	Year	Objective	Merits	Demerits
[9]	2018	To highlight class-dependent and simplifies the network	This is conceptually straightforward, lightweight and compatible with existing methods	To get much performance, large number of training dataset is needed
[1]	2019	To develop a segmentation network for railroad detection	Better results for segmentation	Requires more time for computation
[10]	2021	Focused on segmentation of liver and its tumor by minimizing the time and effort	Better segmentation results and showed improvement	Faced difficulty in segmenting small and irregular tumors
[11]	2021	Presented dense recurrent residual convolutional neural network which is a synthesis of recurrent CNN, residual network and dense convolutional network	Better performance in segmentation tasks with almost same number of network parameters	Not performing well for other benchmarking datasets to analyze its performance
Proposed system	2021	Focused on segmentation of railroad by reducing its training time and run time using UNet by modifying its number of layers	Shows better segmentation results and improved the performance	Detection of railroad in dark condition is poorer, need to increase the accuracy

We have proposed the method to detect the railroad by modifying the UNet model. The existing models represent the negative effects of camera movement and processing the images at high speed and not able to detect the path. Some methods resulted in more computation time. And some resulted in increase of model complexity as increase in number of layers and decrease in the performance of segmentation.

We have compared our model with residual UNet, where its run time is minimum of 30 min, i.e., it took more time to compute, process the image and detect the path. Even segmentation is poor. Thus, we have decreased the number of layers to 25 in modified UNet, and it took maximum of a minute for run time. Even training time of our model is less when compared to other models. However, using Gaussian blur feature of OpenCV 4.0.0 resulted in better visualization compared to other methods.

2.1 Railroad Detection

There are many methods of railroad or rail extraction in recent years. An algorithm is available that uses dynamic programming in railroad environments to extract the training course and railroad track in front of the train. They made this method using these steps: First the image needs to be detected. Thus, the image has been used twice, first the tracks will be marked, and the next, inside the tracks will be marked. These are used to find the path and then convert them into binary images using MATLAB.

2.2 Segmentation

Railroad detection requires the segmentation of the region that contains the railroad image, similar to the lane detection problem [8]. To perform image segmentation, fully convolutional network (FCN) is proposed in [12]. To improve the resolution of segmentation in [5], SegNet uses encoder–decoder architecture. To systematically aggregate multi-scale contextual information without losing resolution, in [13] dilated convolution is used. In [14] proposed an atrous spatial pyramid pooling (ASPP) method based on dilated convolution to robustly segment objects at multiple scales. A pyramid scene parsing network (PSPNet) uses global context information for fast and efficient semantic video segmentation, a dynamic video segmentation network (DVSNet) presented by Xu et al. [15].

2.3 UNet

Our railroad detection task is done using UNet architecture. The architecture contains two paths. First path is the contraction path (also called the encoder) which is used to capture the context in the image. The encoder is just a traditional stack of convolutional and max pooling layers. The second path is the symmetric expanding path (also called the decoder) which is used to enable precise localization using transposed convolutions. Thus, it is an end-to-end fully convolutional network (FCN), i.e., it only contains convolutional layers and does not contain any dense layer because of which it can accept image of any size. In the encoder, the size of the image gradually reduces, while the depth gradually increases.

This basically means the network learns the “WHAT” information in the image; however, it has lost the “WHERE” information. The right-hand side is the expansion path 3 (decoder) where we apply transposed convolutions along with regular convolutions. In the decoder, the size of the image gradually increases, and the depth gradually decreases. Decoder recovers the “WHERE” information (precise localization) by gradually applying upsampling. To get better precise locations, at every step of the decoder, we use skip connections by concatenating the output of the transposed

convolution layers with the feature maps from the encoder at the same level: u6 = u6 + c4; u7 = u7 + c3; u8 = u8 + c2; u9 = u9 + c1. After every concatenation, we again apply two consecutive regular convolutions so that the model can learn to assemble a more precise output. This gives the architecture a symmetric U-shape, hence the name UNet. On a high level, we have the following relationship:

$$\text{Input}(400 \times 600 \times 3) => \text{Encoder} => \text{Decoder} => \text{Output}(400 \times 600 \times 1)$$

3 Proposed System

At present, we have a number of deep neural network architectures which can be used for semantic segmentation. Execution time is one of the factors of efficiency. Semantic labeling is employed as a preprocessing step. UNet model is first designed model for biomedical image segmentation.

3.1 System Architecture

The model we are going to use is UNet model. The primary objective of this model is to convert the input feature map into a vector. We have to reconstruct an image from this vector where the UNet comes into the picture. We use this feature map which helps us to convert the feature map into a vector which is a reverse function. To do this, we have contracting and expanding path.

This system comprises two parts, i.e., encoding and decoding. The encoder part encodes the input image into compact representations. The decoder part recovers the representations to a pixel-wise categorization, i.e., semantic segmentation. The input and output of our semantic segmentation network have same size in width and height of 400*600. We have used this model, but we have reduced the number of convolutional layers and then trained the model with our collected dataset. The loss function used for image segmentation is a pixel-wise cross-entropy loss and used stochastic gradient decent (SGD) as optimizer with momentum 0.9, learning rate 0.01 and weight decay 10–4.

Figure 1 represents our system architecture which is an end-to-end fully convolutional network; that is, it contains only convolution layers not dense layers. There is total 25 layers in our model. Table 2 represents the process that takes place in every level. In encoder side, there are three levels where in each level, it performs convolution twice consecutively and pooling. In decoder side, there are three levels where in each level, it performs transpose convolution and convolution. And at each level of decoder, concatenation takes place by concatenating output of transposed convolution with the feature maps of encoder at respective level.

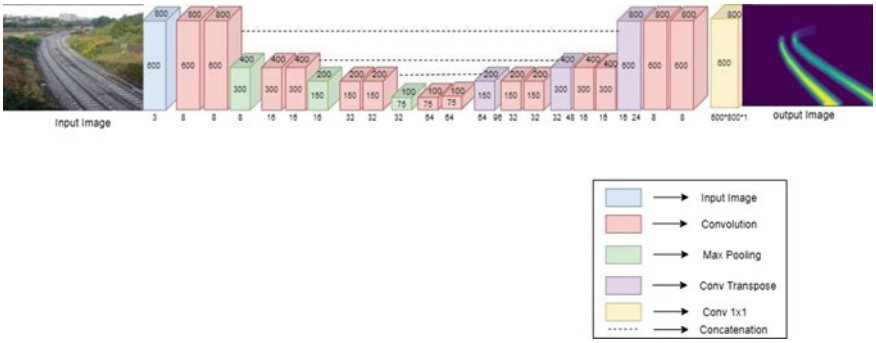


Fig. 1 Main design of the modified UNet

Table 2 Network structure of modified UNet

	Unit level	Layers	Filter	Stride	Output size
Input					600 × 800 × 3
Encoder	Level 1	Twice conv1, Max pool1	3 × 3/8, 2 × 2	1	300 × 400 × 8
	Level 2	Twice conv2, Max pool2	3 × 3/16, 2 × 2	1	150 × 200 × 16
	Level 3	Twice conv3, Max pool3	3 × 3/32, 2 × 2	1	75 × 100 × 32
	Level 4	Twice conv4	3 × 3/64	1	75 × 100 × 64
Decoder	Level 5	conv2DTrans, Twice conv5	2 × 2/64, 3 × 3/32	2	150 × 200 × 32
	Level 6	conv2DTrans, Twice conv6	2 × 2/32, 3 × 3/16	2	300 × 400 × 16
	Level 7	conv2DTrans, Twice conv7	2 × 2/16, 3 × 3/8	2	600 × 800 × 8
Output		conv8	1 × 1		600 × 800 × 1

Convolution layer is the building block of CNN. This involves in dot product o*f* two matrices, where one matrix is the set of learnable parameters known as kernel and the other matrix is restricted portion of receptive field. In pooling layer, it involves sliding a two-dimensional filter over each channel of feature map and summarizing the features lying within the region covered by the filter. Transposed convolution is the opposite process of convolution layer. The main process behind this is that input and output are handled in backward direction where this converts low resolution image to high resolution image.

4 Implementation and Results

To implement our model, first we need dataset. So, we have collected the dataset which consists of rail track images of around 2000. Some of the images were collected from Web, and some were from videos where we took snapshots.

It consists of various types of images with different influences of lighting conditions which made our model to train in all conditions and makes us to test our model.

4.1 Preprocessing

In preprocessing step, each and every image is used twice, first for marking rail tracks, and the next is to shade inside the track. Likewise, images are labeled using some painting applications like paint, Adobe Photoshop, etc. This process is represented in Fig. 2. We marked rail lines and rail inside part red in color. These photos need to be converted into binary images. Thus, they were masked using MATLAB. To do these, we have used Color Thresholder app under image processing and computer vision in MATLAB. Since we labeled the collected images manually, every images are masked properly that rose to give better segmentation results. We were able to mark the images properly, but cannot be able to represent dark images. Thus, this will not give proper result in case of dark images (varying weather conditions).

We worked on different models with same dataset but that took more hours for training and gives less accuracy compared to UNet model. Then, we concluded to

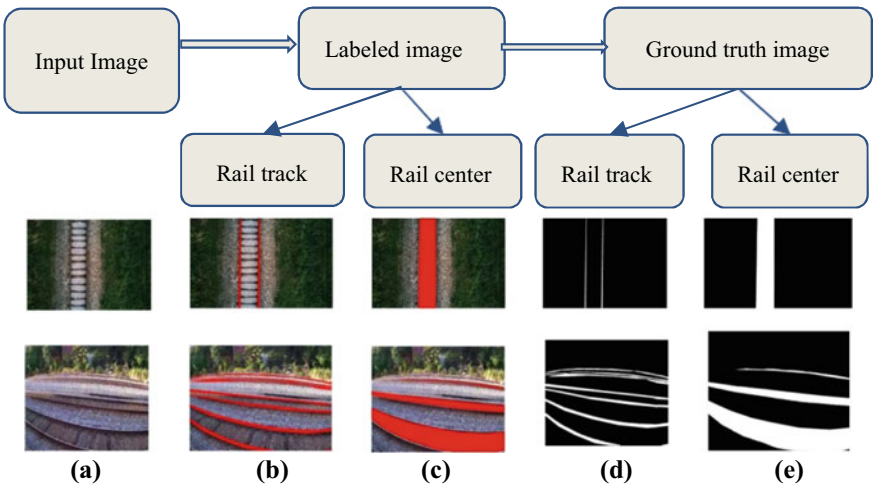


Fig. 2 **a** Input image, **b** labeled rail track, **c** labeled rail center, **d** ground truth rail track, **e** ground truth rail center

optimize our model by reducing time complexity and space complexity. As a result, we modified architecture by reducing the number of layers.

The proposed system is implemented using UNet model in Google Colab with 15 GB of RAM. We have used the dataset which was collected manually from the Web which consists of 2000. We got the accuracy of 92% which was better when compared to other methods.

We had used residual UNet model using residual units and detected the railroad using same dataset. But we got better segmentation in the case of above proposed model that is modified UNet. In residual UNet, there are 64 layers, whereas now we have 25 layers which decrease the complexity of model.

Modified UNet model provided better segmentation because of loss function binary cross-entropy and conv2Dtranspose function for upsampling. As well as using Gaussian blur feature of OpenCV 4.0.0 results in better visualization compared to the previous work. Even though it took 15 GB of memory, it highly reduces the train time and run time.

Figure 3 represents the predicted image of the residual UNet model and modified UNet model. The predicted image has three channels where each color represents track, rail inside and background or environment of the image.

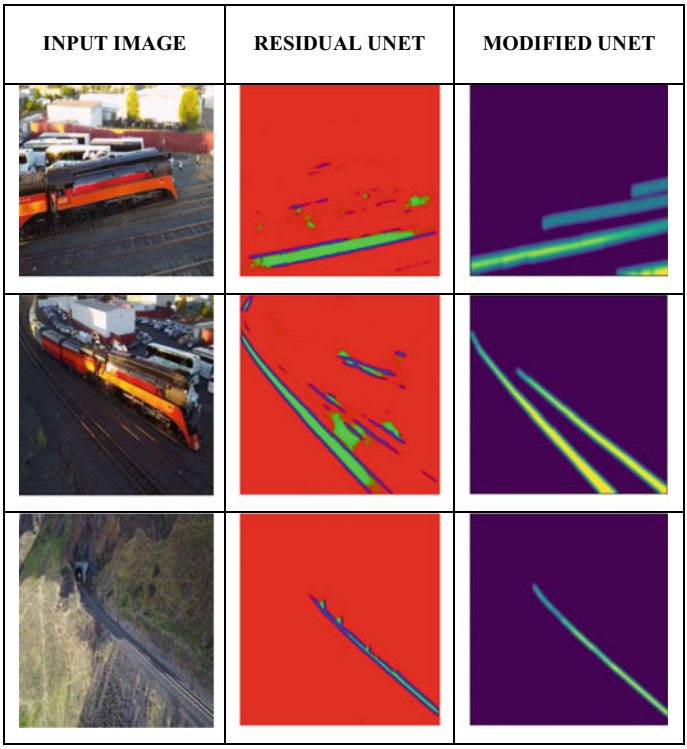


Fig. 3 Input image and its corresponding predicted images of residual UNet and modified UNet

The total parameters of modified UNet is 143,729, trainable parameters are 143,729, and total parameters of residual UNet are 729,504, trainable parameters are 726,048, and non-trainable parameters are 3456. The predicted images of residual UNet have three channels where blue represents track, green represents rail inside, and red represents background or environment of the image.

We have plotted the graph for training accuracy and testing accuracy against epochs which is shown in Fig. 4. This represents that our model is not overfitting because there is only slight difference between the lines, i.e., train score and cross-validation score.

The above graph Fig. 4 represents that loss minimization function (binary cross-entropy which is commonly used in machine learning as a loss function. It is a measure in the field of information theory which generally calculates the difference between two probability distributions) did a great job, reduced loss highly, and there is early stopping at 15th epoch.

The components and its respective values of modified UNet are mentioned in Table 3. We have reduced the layers in modified UNet where our previous work model, i.e., residual UNet had 64 layers. This results in reduction of model complexity. And major important component in modified UNet is 15 GB of RAM. This decreased the training time and run time of our model.

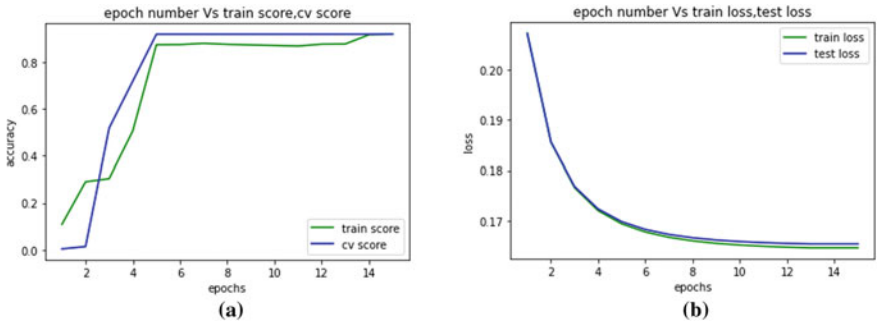


Fig. 4 **a** Epoch number versus train score, cv score, **b** epoch number versus train loss, test loss

Table 3 Components of modified UNet

Components	Value
Layers	25
Trainable parameters	143,729
Non-trainable parameters	0
RAM	15 GB
Loss function	Binary cross-entropy
Loss	−11e−7
Software	Google Colab
Optimizer	Adam

Table 4 Accuracy results

Deep learning models	Pixel accuracy
FCN	80.7
Residual UNet	91.7
Modified UNet	92

5 Performance Evaluation

For the evaluation of model, we need to see the performance of the segmentation by comparing its binary mask of segmentation result to its ground truth where similarity is estimated. We have used accuracy as performance metric and compared with other models. Generally, accuracy provides good segmentation results where it represents the ratio of correctly segmented samples to the total samples. The accuracy got by different models is mentioned in Table 4.

Even though accuracy of residual UNet and modified UNet is closer, the segmentation is very good in modified UNet. But modified UNet required 15 GB of RAM. Even though it used 15 GB of RAM, it highly reduced training time and run time when compared to other methods. This is one of the major advantages, because to detect the rail path, the time taken to detect should be less. Thus, this has been satisfied in this modified UNet model.

6 Conclusion

In this paper, we have implemented UNet by modifying its number of layer and detected the railroad. We can even detect the railroad using learning algorithms, but it is difficult to detect the railroad using bounding box. Thus, we have collected dataset and converted them to binary manually and utilized them to detect the railroad. These have helped our model and gave the highest detection preformation when compared to other methods. However, this model requires GPU to run that is around 15 GB of RAM. In future, we are going to focus in the increase of accuracy and to manage space complexity.

References

1. Wang Y, Wang L, Hu YH, Qui J (2019) RailNet: a segmentation network for railroad detection
2. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV]
3. Gibert X, Chellappa R, Patel VM (2015) Material classification and semantic segmentation of railway track images with deep convolutional neural networks
4. Mukojima H, Murase H, Deguchi D, Kawanishi Y, Ide I, Ukai M (2016) Moving camera background-subtraction for obstacle detection on railway tracks. IEEE

5. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39
6. Zhang Z, Liu Q, Wang Y (2018) Road extraction by deep residual U-Net
7. Le Saux B, Beaupère A, Boulch A, Brossard J, Manier A, Villemin G (2018) Railway detection: from filtering to segmentation networks. In: Proceedings of the IEEE international geoscience and remote sensing symposium, Valencia, Spain
8. Neven D, De Brabandere B, Georgoulis S, Proesmans M, Van Gool L (2018) Towards end-to-end lane detection: an instance segmentation approach. In: Proceedings of the IEEE intelligent vehicles symposium (IV)
9. Zhang H, Dana K, Shi J, Zhang Z, Wang X, Tyagi A, Agrawal A (2018) Context encoding for semantic segmentation. In: IEEE/CVF conference on computer vision and pattern recognition
10. Ayalew YA, Fante KA, Mohammed A (2021) Modified U-Net for liver cancer segmentation from computed tomography images with a new class balancing method. J Mater Sci
11. Dutta K (2021) Densely connected recurrent residual (dense R2UNet) convolutional neural network for segmentation of lung CT images
12. Erhan D, Bengio Y, Courville A, Vincent P (2009) Visualizing higher layer features of a deep network. University of Montreal, Montreal, QC, Canada, Technical Report
13. Yu F, Koltun V (2016) Multi-scale context aggregation by dilated convolutions. In: Proceedings of the ICLR
14. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the ECCV
15. Xu Y-S, Fu T-J, Yang H-K, Lee C-Y (2018) Dynamic video segmentation network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition
16. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation, CVPR
17. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: MICCAI
18. Pathak D, Krahenbuhl P, Darrell T (2015) Constrained convolutional neural networks for weakly supervised segmentation. In: ICCV
19. Girshick R, Donahue J, Darrell T, Malik J (2015) Region-based convolutional networks for accurate object detection and segmentation, PAMI
20. Nassu BT, Ukai M (2011) Rail extraction for driver support in railways. In: Proceedings of the IEEE intelligent vehicles symposium (IV), Baden-Baden, Germany
21. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: NIPS
22. Lyu P, Yao C, Wu W, Yan S, Bai X (2018) Multi-oriented scene text detection via corner localization and region segmentation. In: Proceedings of the IEEE/CVF conference computer vision pattern recognition
23. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR)

Weight Pruning in Convolutional Neural Network for Efficiency in Plant Disease Detection



Priyang Patel, Madhav Tripathi, Mohammed Husain Bohara, and Parth Goel

Abstract Plant infection is an industrious issue for smallholder ranchers, representing a danger to their livelihoods and meal preservation. Image classification in agriculture has become possible thanks to the latest spike in mobile use and computer vision models. Convolutional neural networks are the cutting edge in picture acknowledgment, and they can give a quick and precise finding. In spite of the way that these CNN models are profoundly helpful in an assortment of PC vision exercises, the high number of boundaries makes them computational and memory escalated. Pruning is a significant procedure for diminishing the quantities of boundaries in a CNN model by wiping out superfluous loads or channels without bargaining generally speaking accuracy. The effectiveness of a weight pruning in CNN model in detecting crop disease is studied in this paper. The created models, which are available as a Web API, can detect different plant diseases according to selected plant. For training and validating the model, the dataset used is new plant diseases dataset available on kaggle. The suggested approach will obtain far better accuracy than the normal approach, according to validation results. This illustrates CNNs' technological flexibility in classifying plant diseases and opens the way for AI solutions for smallholders.

Keywords Weight pruning · Classification · CNN · Plant disease detection · Precision agriculture

P. Patel (✉) · M. Tripathi · M. H. Bohara · P. Goel
Department of Computer Science and Engineering, Devang Patel Institute of Advance Technology and Research (DEPSTAR), CHARUSAT, Charotar University of Science and Technology (CHARUSAT), CHARUSAT Campus, Changa 388421, India
e-mail: 18dcs077@charusat.edu.in

M. Tripathi
e-mail: 18dcs129@charusat.edu.in

M. H. Bohara
e-mail: Mohammedbohara.ce@charusat.ac.in

P. Goel
e-mail: Parthgoel.ce@charusat.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_13

1 Introduction

To satisfy the anticipated need, worldwide yield creation would increment by in any event half [1]. Right now, the greater part of action happens in Africa and Asia, wherein excess of 80% of ranchers are family-run and have practically zero agrarian experience [2, 3]. Subsequently, yields of over half are typical because of vermin and illnesses [4]. The regular strategy for human translation by visual perception is not, at this point, suitable for ordering crop infections. PC vision models incorporate an advantageous, standard, and precise answer for an emergency. Everything necessary is a Web association and a cell phone with a camera. Mainstream business application 'Plantix Preview' [5] can exhibit how this can be executed. This application has been successful in giving clients ability as well as in making a drawing in an online local area. A convolutional neural organization (CNN) has made various examination achievements in numerous fields, including picture acknowledgment, object recognizable proof, and semantic division, because of late advances in the capacity of designs handling units. In any case, because of its enormous model scale, it is hard to use on PCs with negligible assets [6].

Pruning has been broadly explored as an approach to pack CNN models. It points by barring the model's less critical loads or channels. The profound pressure technique was presented by Han et al. [7]. The aim of this analysis is to see how well weight pruning can be effective to train a plant disease classifier. There are different datasets for each plant, containing different diseases. An API for each model is developed to classify the disease on plant leaves. The plants are (potato) *Solanum tuberosum*, (tomato) *Solanum lycopersicum*, (apple) *Malus domestica*, (cherry) *Prunus avium*, (corn) *Zea mays*, (grapes) *Vitis vinifera*, (bell pepper) *Capsicum annuum* Group, and (strawberry) *Fragaria* × *ananassa*. The model will be learned to recognize a limited range of diseases or states of health for each species. The particular purposes of this examination are to

- Use an approval and test dataset to decide the model's overall viability in ordering plant illnesses.
- Evaluate the model's performance when weight pruning.
- Use the took-in models to make API and a mobile application.

First, a review of related work is presented. Second, what resources and approaches were used to implement this strategy? Third, discuss the classification's fundamental idea and the method employed to achieve it. Fourth, explained how and why pruning can be an effective solution with pseudocode. Fifth, for all models, an assessment of three-layer CNN effectiveness after and before pruning. Sixth, touched about how to deploy and integrate the solution. Finally, the article's conclusion is presented in the final part.

2 Related Work

Training an ANN with a high degree of uncertainty requires a lot of computation. The existence of redundancy in ANNs, as well as the observation that the ranking of the relevance of the weights changes slightly throughout training, enlightened that [8]. Multi-scale convolution applies procedures such as grouping convolution, depth separated convolution, feedback loop, channel shuffle, and others to enhance feature map, extract more plentiful features, minimize model parameters, and achieve network lightweight. Pruning process is performed for the lighter model that has been originally trained to cut filter weight that really is not important to reduce model redundancies [9]. The strategy's sole goal is to generate meaningful connections and reduce the remaining network structure [10].

3 Data Acquisition and Preprocessing

Data acquisition of photographs of potato, tomato, apple, cherry, corn, grapes, bell pepper, and strawberry is from 'The Plant Village Dataset' [11], an open-access archive available on kaggle. A number of groups are selected for each species, with information available in Table 1. All the dataset had been split according to plant category. The splitting of dataset is done: 80% for training and 20% for validation for each plant. Then, all the images are resized as 250 × 250 and given us input in training as well as validating in batch size of 100. All of the work for this study was done on Google Collab having memory 13 GB of RAM, processor 1 × 2.3 GHz single core hyperthreaded Xeon Processors (2 threads with 1 core), and graphics 1 T K80 with 2496 CUDA cores and 12 GB of GDDR5 VRAM, compute 3.7.

Table 1 Dataset info

Plant	Number of classes	Total images
Apple	4	9710
Cherry	2	3684
Corn	4	9145
Grape	4	9027
Peach	2	4457
Pepper	2	4876
Potato	3	7128
Strawberry	2	4498
Tomato	10	22,930

4 Need of Pruning and How to Achieve It

It was observed that many of the weights in a fully trained neural network are extremely near to zero. This means that these weights have no bearing on the neural network's output. Using this concept, go over the CNN layers in the model, looking for filter channels with weights that are very near to zero, and removing them from the layer entirely. Following that, link the individual CNN layers as needed, i.e., using a layer to pad an input/output tensor with zeros to acquire the correct shape where necessary; to ensure that information may still flow properly through the model. As a result, the model is substantially smaller but still capable of performing the image categorization and object identification tasks for which it was trained.

To begin, in phase one, three-layered CNN model is defined and trained each model according to plant category and data available. The reason for building different models for different plants is to get better functionality without compromising the accuracy, building different end points for API such that model can process fast and get a quick response in mobile, and also for user-friendly UI of mobile application, it is very beneficial.

In phase two, applied the pruning method to decrease network intricacy and lighten the overfitting issue. In CNN, there are a ton of standards, yet some of them do not contribute a lot to the last exhibition, so they are excess. The calculation of convolution layers and totally associated layers includes countless gliding point framework augmentations, and most CNNs have huge model boundaries. Pruning is a promising strategy for dealing with the problem that CNN models can be difficult to use on Machines with limited resources [12]. The profound speed increase strategy was presented by Han et al. [12]. All connection loads under a specific level are killed, alongside extra pressure through quantization and encoding techniques, namely Huffman encoding, and afterward, precision is calibrated [13]. This will bring about a model that is exceptionally inadequate. Underneath Fig. 1 shows the weight pruning strategy.

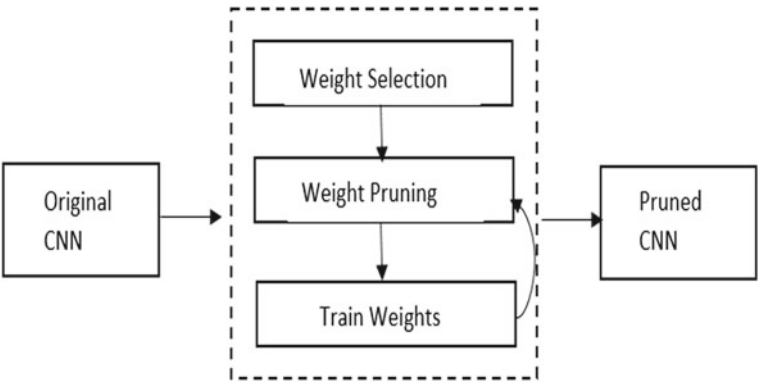


Fig. 1 Weight pruning strategy

The following three stages make up weight pruning.

1. Consider the significance of weights. Rather than having the final weight values, learn which relations are essential. Random weights were used to start the network. The Adam optimization technique was used to optimize network weights using a categorical cross-entropy loss metric.
2. Unimportant connections are pruned from the organization. Under a specific level, all association loads are set to nothing. This outcomes in an neural network that is meagerly wired. The meager grids are put away in a smaller scanty line and compacted inadequate segment designs. Subsequently, zero-weighted loads do not affect memory.
3. To gain proficiency with the last loads for the leftover scanty connections, retrain this inadequate organization. It ought to be recollected that during the retraining stage, loads with a worth of zero would stay unaltered.

Subsequent to resetting certain loads to 0, it will evaluate the segment of the neural network that has a minor impact on accuracy. From that point forward, sort the loads, and put down a boundary. Loads beneath the edge are set to nothing and stay consistent during the preparation. Keep on rehearsing until the model is prepared.

Pseudocode

Pseudo Code.

```
Input: w, Represent the weights in Layer it
Output: Re Is the subset of weights which are removed
Re ←  ϕ;
Li ←  Li1, Li2,..., Li11;    /* Li is the total number of
Layers in CNN */
For each item P in Li do
A = original_acc - Zero_reset acc
/* original_acc is the Original Accuracy of the CNN Zero_reset_acc is
the Accuracy after Resetting weights zero */
Rank w, according to A;
 set a threshold value as T;
    if w < 1 then
       reset w = 0;
       Continue training until accuracy recover;
    and if
Re ← w;
End For
```

5 Results and Comparison

After saving all the models, API end points are the same as the number of plants category created. In API, just need to pass the image along with the plant name. For

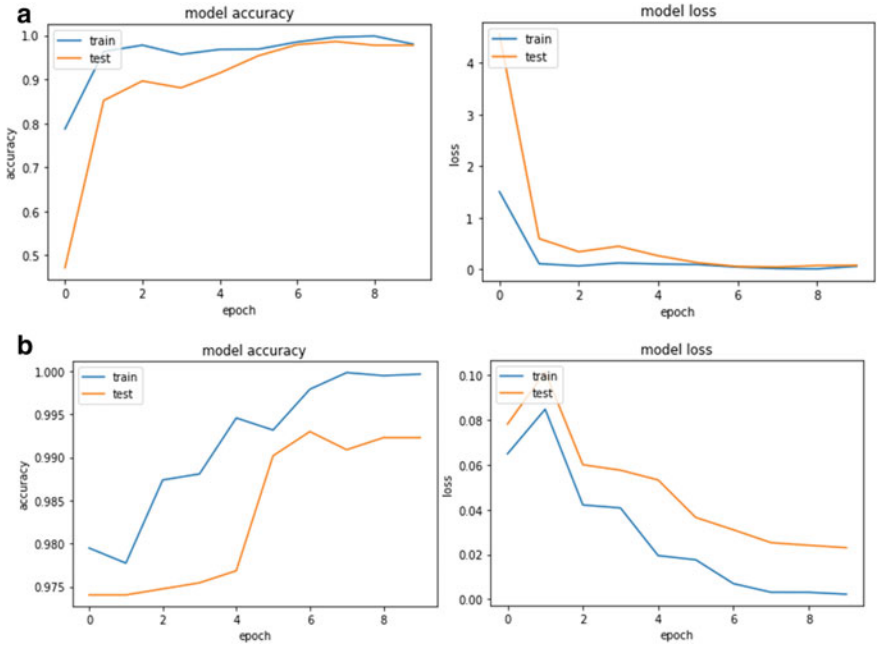


Fig. 2 **a** Graph for potato model before pruning, **b** Graph for potato model after pruning

Table 2 Comparison of validation time and accuracy

Plant	Before pruning (validation data)	After pruning (validation data)
Apple	Time (s): 57.747 Accuracy: 0.9835	Time (s): 55.237 Accuracy: 0.9821
Cherry	Time (s): 32.042 Accuracy: 0.9988	Time (s): 31.265 Accuracy: 0.997
Corn	Time (s): 66.357 Accuracy: 0.9595	Time (s): 65.121 Accuracy: 0.9617
Grape	Time (s): 63.684 Accuracy: 0.9745	Time (s): 62.695 Accuracy: 0.9612
Peach	Time (s): 28.110 Accuracy: 0.9943	Time(s): 26.833 Accuracy: 0.9943
Pepper	Time (s): 58.782 Accuracy: 0.9825	Time (s): 58.726 Accuracy: 0.9825
Potato	Time (s): 46.609 Accuracy: 0.9796	Time (s): 45.344 Accuracy: 0.9786
Strawberry	Time (s): 38.402 Accuracy: 0.9800	Time (s): 35.572 Accuracy: 0.9777
Tomato	Time (s): 80.385 Accuracy: 0.9645	Time (s): 77.932 Accuracy: 0.9644

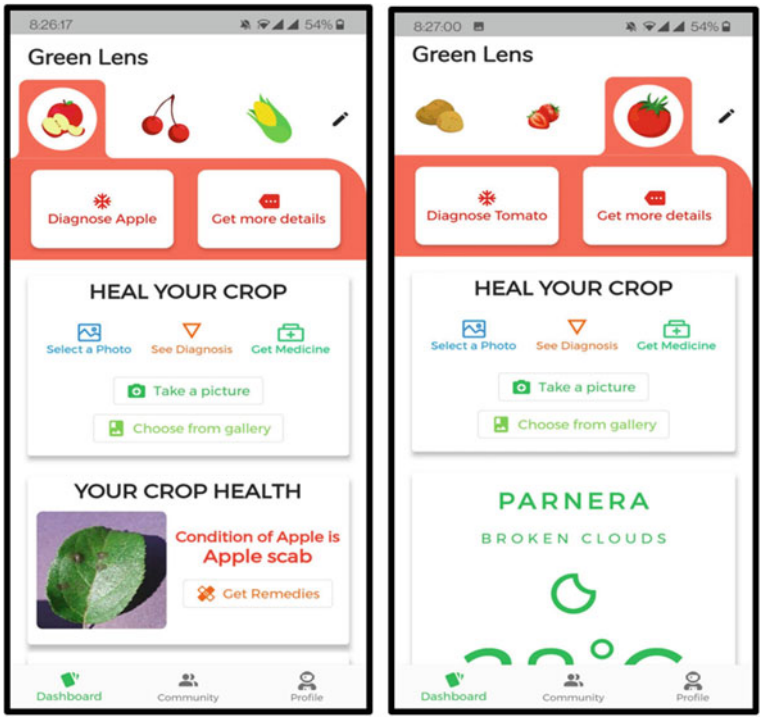


Fig. 3 Snapshot of mobile application

instance, the potato model is given by the graphs in Fig. 2a, b. Table 2 depicts the results.

6 Deployment and Integration

Finally, the models were deployed as API [14], and it is used as a service in mobile applications (Fig. 3). This provides the smallholders with a live disease classification with user-friendly UI and a platform to form a community.

7 Conclusion

Small-scale farmers rely on prompt and reliable crop disease detection to avoid losses. In this article, a convolutional neural network is being used. Using this approach of splitting models according to plant category and pruning technique, the results are more precise and better than that of previous approaches. The end product was an app

for detecting plant diseases. This service is free, simple to use, and what is needed is a smartphone and an Internet connection to use it. Adding to this, it can be concluded that weight pruning can increase a model's efficiency. The findings indicate that the proposed approach will efficiently eliminate redundancy in CNN models and outperforms current approaches and also overcome the problem of overfitting. The size of the models has no significant difference, and in accuracy, slight changes are observed, but average time taken for validating decreases by 2 s. There are at least two major future directions for this approach. First, better applications can be created with faster predictions. Second, compressing these models can lead to directly using it in applications without any API such that the process will be much faster.

References

1. Hunter MC, Smith RG, Schipanski ME, Atwood LW, David A (2017) Agriculture in 2050: recalibrating targets for sustainable intensification. Mortensen BioSci 67(4):386–391
2. PlantVillage. Plantvillage.psu.edu, 2020 [Online]. Available: https://plantvillage.psu.edu/. Accessed 28 Dec 2020
3. Klauser D (2018) Challenges in monitoring and managing plant diseases in developing countries. J Plant Dis Prot 125:235–237
4. Muimba-Kankolongo A (2018) Food crop production by smallholder farmers in Southern Africa 1st edition challenges and opportunities for improvement. Academic Press
5. Plantix Preview. Plantix [Online]. https://plantix.net/en/
6. Yang W, Jin L, Wang S, Cu Z, Chen X, Chen L (2019) Thinning of convolutional neural network with mixed pruning
7. Han S, Mao H, Dally WJ (2016) Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding
8. Zhang J, Chen X, Song M, Li T (2019) Eager pruning: algorithm and architecture support for fast training of deep neural networks
9. Wu Y, Xu L (2020) Lightweight compressed depth neural network for tomato disease diagnosis
10. Finlinson A, Moschoyiannis S (2020) Synthesis and pruning as a dynamic compression strategy for efficient deep neural networks
11. Samir B (2019) New plant diseases dataset: image dataset containing different healthy and unhealthy crop leaves
12. Han S, Mao H, Dally WJ (2016) Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding, vol 56, no 4, pp 3–7
13. Gong Y, Liu L, Yang M, Bourdev L (2014) Compressing deep convolutional networks using vector quantization
14. Bohara MH, Mishra M, Chaudhary S (2013) RESTful web service integration using Android platform. In: 2013 fourth international conference on computing, communications and networking technologies (ICCCNT). IEEE

An Edge Computing-Assisted Internet of Things-Based Secure Smart Home



S. Narasimha Swamy, P. Shivanarayanamurthy, Gaurav Purohit, and Kota Solomon Raju

Abstract Internet of Things (IoT) gains massive popularity due to the integration of different technologies like sensors, embedded, information, and communication technologies. The smart home is the key application domain of IoT, which brings comfort and security to the residents and increases cost saving. IoT plays a significant role in monitoring and analyzing the events in smart homes; these events are shared with the residents as alerts. Network congestion and response time increase in the smart home as the number of requests/responses increases to the cloud. However, local data processing and immediate responses are the primary requirements of a smart home. Edge computing satisfies all these requirements by executing the operations at the edge of the network. In this paper, edge computing-assisted IoT-based secure smart home architecture is proposed. The architecture facilitates features like data storage and processing, data monitoring, security, and visualization. Also, a lightweight authentication scheme is developed in order to identify the legitimate devices in the smart home network. In this architecture, Raspberry Pi3 boards are used as end devices. Raspberry Pi4 and Samsung Artik-710 are used as the edge nodes which provide the storage, processing, analysis, alert, and security characteristics at the edge of the network in smart homes. Finally, performance analysis of the edge nodes used in this architecture is given.

Keywords Internet of Things · Smart homes · Edge computing · Device authentication · Monitoring

S. Narasimha Swamy (✉) · G. Purohit · K. S. Raju
Academy of Scientific and Innovative Research, Ghaziabad, India

S. Narasimha Swamy · P. Shivanarayanamurthy · G. Purohit · K. S. Raju
CSIR-Central Electronics Engineering Institute, Pilani, India

S. Narasimha Swamy
Rashtreeya Vidyalaya College of Engineering, Bengaluru, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_14

1 Introduction

Internet of Things is a global network of "Things" or "Objects" or "End Devices." The idea of IoT was primarily initiated in the MIT Auto-ID laboratory [1]. IoT devices possess sensing and actuating, computing, and communicating characteristics. Sensing allows the IoT devices to collect the data from the environment using different sensors, and actuation allows the triggering of the events based on the decisions. Computing plays a significant role in making decisions by analyzing the data coming from smart sensors. The communication feature enables IoT devices to share the data at anytime and anywhere. These features made use of IoT in various applications like smart homes, smart parking, smart health care, intelligent transportation, smart farming, structural health monitoring, industrial IoT, etc. The use of IoT applications is growing gradually, and the ultimate goal of the IoT is to assist human beings in their daily needs and activities. Figure 1 shows the significant share of IoT applications in the current market [2].

The smart home concept is not new, which was existing from 1930 with the vision of "Homes for Tomorrow" [3]. A smart home is an important and influential application in the society which brings well-being in terms of health, power saving, monitoring, safety, and security, etc. Smart home is also referred to as "Intelligent Homes" most of the time [4]. A smart home is an interconnection of household devices (lights, fans, automotives, air conditioners, mobiles, laptops, doors, washing machines, etc.) that work in coordination to enhance the resident's quality of life. A smart home comprised of heterogeneous networks and devices to make resident's life better. A smart home uses various networking technologies like body area networks (BAN), personal area networks (PAN), local area networks (LAN), and wide area networks (WAN). Home appliances, sensors, computing devices,

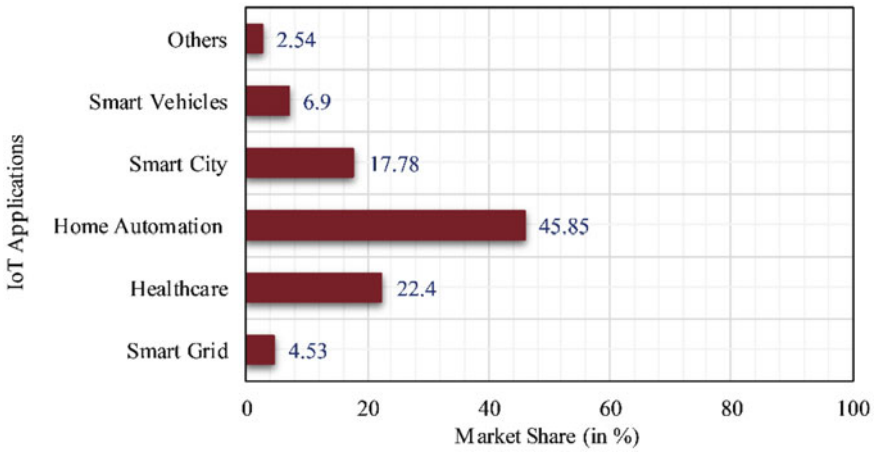


Fig. 1 Most common IoT applications in the market

networking devices, multimedia devices, etc., are the most commonly used in smart homes [5].

In recent days, the number of devices in smart homes increases gradually. Approximately, a person in a smart home may have three devices. The existing IoT smart home paradigms periodically collect the massive amount of data from the sensors from the environment and send it to the centralized remote server; these servers process the data. However, centralized remote servers have the limitations like latency, reliability, security, mobility, and others. The use of edge computing overcomes most of the preceded limitations. Edge computing is a new distributed computing paradigm, which facilitates data storage and performs all the computations, and brings the data storage at the edge of the network [6, 7].

In future, smart homes are common in the society; it comprises tiny devices ranging from toothbrushes to big home appliances like refrigerators, washing machines, etc. In recent days, the number of devices joining smart home networks is increasing; this creates the device and data monitoring, and management more challenging; to handle this challenge effectively, decentralized smart home architecture is presented. The proposed work realizes the multistoried building to illustrate the edge computing-assisted IoT-based smart home architecture. A graphical user interface is also provided to perform analysis and visualization. The proposed architecture uses the advantages of both IoT and edge computing technologies efficiently to make the smart home network intelligent, secure, reliable, and scalable.

2 Related Work

Mocrii et al. offer an in-depth analysis of major challenges, technologies, and architectures available in IoT-based smart homes [3]. S. Gunputh et al. proposed a low-cost centralized smart home system in which an Arduino mega-2560 microcontroller acts as a key component [8]. Mokhtari et al. proposed a REST-based smart home architecture. In this, the fog layer performs the limited processing and storage of the data. The authors also used the cloud layer in order to achieve high scalability and flexibility [9]. To allow effective and seamless communication between heterogeneous devices, a new multilayer cloud computing architecture for smart homes is proposed by Tao et al. [10]. Verma et al. developed a fog-based real-time health monitoring application in the smart home. In this, the gateway acts as a fog node, which augments the data processing activities [11]. Jabbar et al. developed an IoT home model that monitors home conditions and automates and controls home appliances using the Internet. The model uses NodeMCU as the gateway. This model also uses the Adafruit IO cloud server for data storage. The authors also developed rich user interfaces for both smartphones and laptops [12]. Mora et al. proposed the monitoring architectures for smart homes, and it is based on cloud computing [13]. Chen et al. presented a new smart home solution called Smart Home 2.0; this provides communication between the residents and the plants inside homes. The authors have developed a mobile-based cloud solution to store, manage and visualize the data [14]. Xu et al. provided the

smart lighting solution for intelligent buildings. In this solution, the gateway offers intelligent functions, and the cloud server provides long-term storage capability [15].

In summary, the conventional smart home solutions use the traditional IoT networks; in this IoT devices, gateway and cloud are the major components. The gateway and cloud perform most of all the activities, including storing, processing, and analysis. The proposed works bring the storing, processing, and computing functionalities one step below the fog computing/gateway, i.e., at the edge of the network. The use of edge computing in the smart home brings real-time data processing and analysis assets; this also empowers swift decision-making, less resource consumption, and privacy and security.

3 Proposed Work

The architecture of the proposed smart home system is shown in Fig. 2; this realizes the multistoried building scenario. End devices, edge–nodes, gateway, and server/cloud are the key components of this architecture. In this work, the devices deployed in the smart home are categorized based on the regions; this helps to identify the origin of the data. As shown in Fig. 3, the living room is considered a region-1, and devices that exist in the kitchen are in region-2. The devices in bedrooms 1 and 2 belong to regions-3 and 4, respectively. Devices present in the passage are considered to be in region-5. Finally, devices deployed in the bathroom are viewed as in region-6; the main reason to use the sensors is to monitor the elderly people and kid's activities.

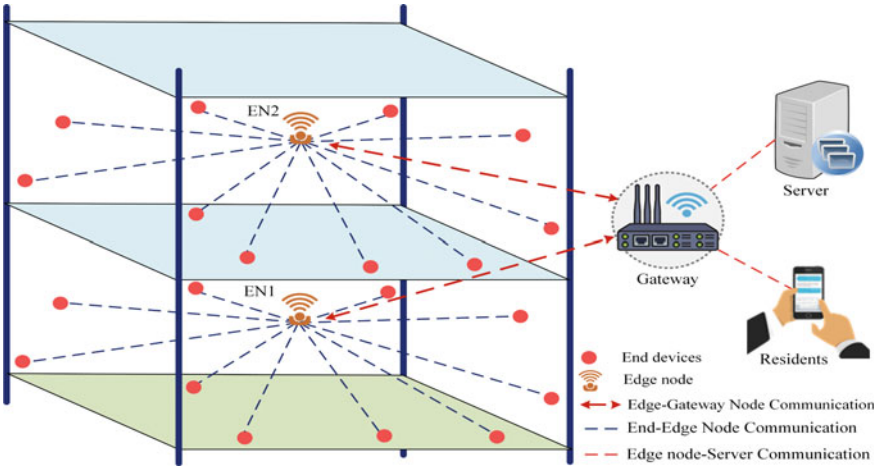


Fig. 2 Realization of edge-based smart home networks in a multistoried smart home

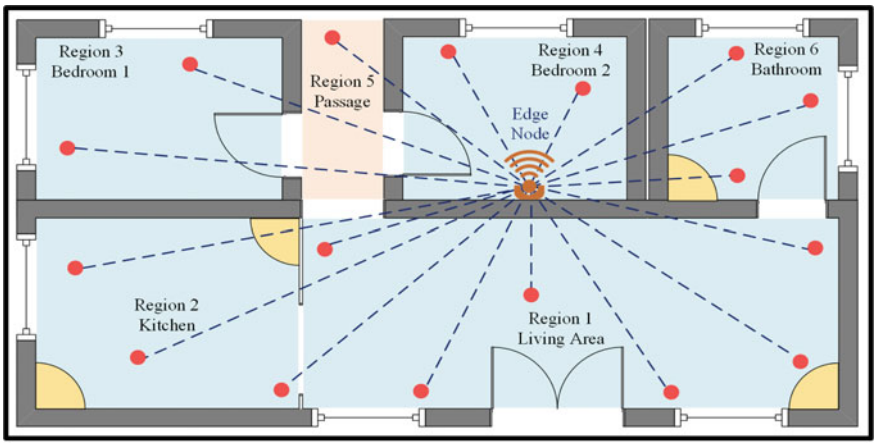


Fig. 3 Deployment of IoT devices in multistoried smart home

3.1 System Implementation

End Devices/Nodes: End devices are the piece of hardware deployed over the smart home area, whose primary task is to collect real-time data from the different regions of the smart home by using the sensors and sending it to the edge node using the wireless communication standards (Wi-Fi). End devices are connected by the various sensors to capture the real-time data from the environment and forward them to the associated edge nodes. In this work, Raspberry Pi3 is used as the end nodes; these nodes are connected with various sensors like DHT11, light-dependent resistor (LDR), ultrasonic proximity sensor, accelerometer sensor, and cameras to capture the real-time data from the smart home environment.

Edge Nodes/Edge Devices: Edge node is assumed to be rich in communication and computation resources. These nodes are used to monitor, manage, and analyze the data coming from the devices. Edge node receives the data from the end devices beneath it. The edge nodes provide services like (1) storage and processing, (2) monitoring, analysis, and notifications, and (3) security. In this paper, Samsung Artik-10 and Raspberry Pi4 are used as the edge nodes. Characteristics of the edge nodes in IoT-based smart home networks are as follows.

- ENs perform the real-time analysis of incoming data.
- ENs act as a nano data center, which stores the data temporarily.
- ENs also perform the authentication of the devices.
- ENs provide the requested services to the server.

Storage and Processing: Huge amount of data is being generated by the sensors in the smart home periodically. In this work, received data is stored in the MongoDB database. It is a document-oriented non-relational database, which provides swift read

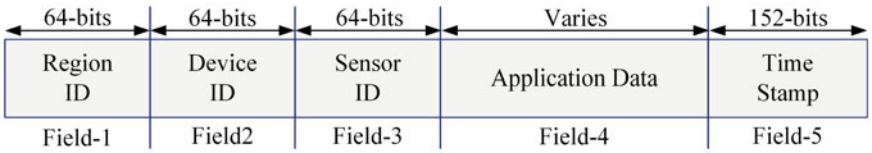


Fig. 4 Data block in MongoDB database

and write operations. MongoDB stores the data in binary object notation (BSON) for—mat with key-value pair. Real-time processing and horizontal scalability are the key features of MongoDB [16]. Edge node stores the data coming from the devices in the format given in Fig. 4. All the fields except "Field-4" possess a fixed size. However, the size of the data varies based on the sensors. For example, images require more memory compared to numerical and text data.

Monitoring, Analysis, and Notification: Edge nodes periodically interact with the underlying end devices to check the operational status by sending a control message. For this purpose, edge nodes maintain the profile library of the underlying end devices; the profile library comprises FloorId, RegionId, DeviceId, and SensorId. Also, incoming data is monitored continuously for anomalies; these anomalous values are communicated to the residents immediately as notification and stored separately for further uses. In this architecture, a lightweight statistical method is used to analyze the intended behavior of the system. When a new value is received, the edge node calculates the mean of the recent five historical values from the database; then, the calculated mean value is compared with the received value, and the maximum acceptable deviation should be less than ± 5 of the calculated mean value. If the received value does not fit into the model, the value is considered as the anomalous value.

Security: It is one of the significant features provided by the edge nodes to secure the underlying end devices from unauthorized access. For example, in recent days, voice-assisted and BLE-based infotainment devices are common in smart homes. Anyone can be connected to the BLE devices and can control the infotainment devices. Similarly, anyone can send the voice command and control the voice-assisted devices. Edge nodes bring the security immediate next to where end devices are deployed and provide the computational flexibility to implement the security algorithms. Another important advantage is that edge nodes decrease the chances of being hacked. In the proposed authentication algorithm, one-way hash, XOR, and concatenation functions are used. Security comprises two important phases: registration and authentication. Table 1 describes the notations used in the authentication protocol.

- *Registration*: If an end device wants to be a part smart home network, first, it should be registered with the edge node by adding the required details into the server. The communication entities use the secure communication channel during this process. The registration process of the end devices in the smart home networks is shown in Algorithm 1. The registration process starts with the end devices by

Table 1 Notations used in the authentication protocol

Symbol	Description	Size of the symbol
DeviceId	Identity of the device	64-bits
Alias_DeviceId	Alias identity of the device	64-bits
ServerId	Unique identity of the server	64-bits
EdgeID	Identity of the edge–node	64-bits
Secret_Key	Secret key for each round	64-bits
$OR = \{R_1, R_2 \ldots R_{6n}\}$	Operating regions of smart home	16-bits
R_n	Random number	64-bits
$T_1, T_2 \ldots\ldots T_n$	Time stamp	184-bits
TC	Current time stamp	184-bits
$M = \{M_1, M_2 \ldots M_n\}$	Message exchanged between the entities	Varying size
$\|\|$	Concatenation operation	–
$\oplus$	XOR function	–
h	One-way hash function	64-bits

sending the registration request to the server. The server receives the registration request and processes it. After processing, OneTime_Key (64-bits) is generated and shared between the end device (M2) and edge node (M3), respectively. End nodes use the OneTime_Key to establish the communication with the edge node for the first time; this ensures the successful registration of the end node. Later, end nodes use the secret key to exchange the application data.

ALGORITHM 1: ***Device Registration***

```
Participants: Server, Edge-node, and End-devices
// Performed by the End-devices
procedure Send_Registration_Request ()
    DeviceId ← Generate_DeviceId ()
    OR ← Region_List []
    T1 ← Get_Current_Time ()
    M1 ← (DeviceId||OR||T1)
    Send (M1)
    Wait ()
    Receive(M2)
    if (TC-T2) ≤ Threshold_Time
        DeviceId'←Alias_DeviceId⊕EdgeID
        B0←h(DeviceId'||OneTime_Key||T2)
        if A0 == B0
            Store (Alias_DeviceId, OneTime_Key)
        end if
    end if
end procedure

// Performed by the Server
procedure Process_Registration_Request ()
    Receive (M1)
    TC←Get_Current_Time ()
    OneTime_Key ←ServerId ⊕ Rn
    if (TC-T1) ≤ Threshold_Time
        Alias_DeviceId←DeviceId⊕EdgeID
        T2 ←Get_Current_Time ()
        A0 ←h(DeviceId||OneTime_Key||T2)
        M2 ← (Alias_DeviceId||OneTime_Key||A0|| T2)
        M3 ←(Alias_DeviceId||OneTime_Key||A0|| T2)
    end if
    Send(M2) // Message Sent to the Device
    Send (M3) // Message forwarded to the Edge-node
end procedure

// Performed by the Edge-node
procedure Add_Device_To_Region ()
    Receive (M3)
    TC←Get_Current_Time ()
    if (TC-T2) ≤ Threshold_Time
        DeviceId'←Alias_DeviceId⊕EdgeID
        B0←h(DeviceId'|| OneTime_Key||T2)
        if A0 == B0
            Store (DeviceId, Alias_DeviceId, OneTime_Key)
        end if
    end if
end procedure
```

- *Authentication*: Authentication protocol allows the end device and edge node to identify each other. During the authentication, process makes use of the public channels. The authentication process uses the public channel to share the information. On successful authentication, the end devices send the captured data to the edge nodes. Algorithm 2 depicts the detailed steps in the authentication process. The authentication process comprises edge nodes and end devices, which exchange messages to authenticate each other. In this process, the edge node

generates a random square matrix (in this case 4 * 4) and extracts the principal and secondary diagonals. The secret key is generated by combining the principal and secondary diagonals. Next, the secret key is shared among the end devices registered under the edge device, and this secret key is used for further communication.

ALGORITHM 2: ***Device Authentication***

```
Participants: Edge-nodes and End-nodes
procedure Authenticate_Device ()
    random_array[][]
    array1[]
    array2[]
    end_node_list[]
    for i←0 to n
        for j←0 to n
            if i == j
                array1[] ←random_array[i][j]
            end if
            if (i+j) == (n-1)
                array2[] ←random_array[i][j]
            end if
        end for
    end for
    Secret_Key ←Concat(array1[], array2[])
    for i ←0 to length(end_node_list)
        T3 ←Get_Current_Time ()
        C ←h (Alias_DeviceId[i] ||EdgeID||S_Key||T3)
        Send (Alias_DeviceId[i], EdgeID, Secret_Key, C, T3)
    end for
end procedure
```

Gateway: It is a networking device, which is an entry and exit point of the smart home network. It also supports long-range communication by providing interoperability between heterogeneous networks and devices. Security features of the network can be increased by analyzing and filtering the packets. It also provides a high level of availability and security.

Server: All the edge nodes in the smart home network are reported to the server and flush the data collected from the underlying end devices in a timely manner. The server can access the edge nodes for a specific purpose. The server provides the graphical user interface to perform device registration and analysis purpose. A high-end system (laptop) with 12 gigabytes random access memory (RAM) and 1 terabyte (TB) read-only memory (ROM) is used as a server. The detailed specification of the hardware used in developing the prototype is given in Table 2 [17, 18].

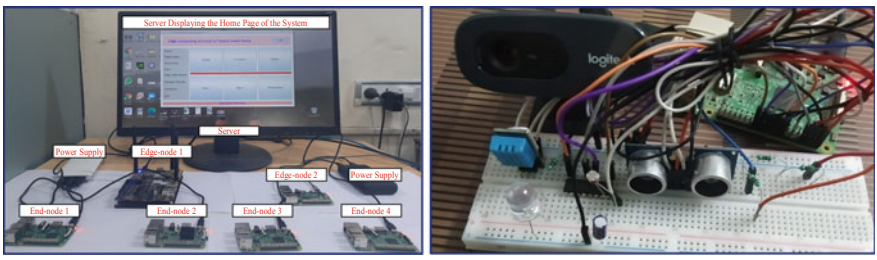
Table 2 Technical specification of the hardware used

Specification	End node	Edge nodes		Server
	Raspberry Pi3	Artik-10	Raspberry Pi4	Windows10
Architecture	ARM Cortex-A53	Quad Core ARM Cortex A7	Quad Core ARM Cortex-A72	I5-7200U
Clock speed	1.2 GHz	10/1.3 and 1 GHz	1.5 GHz	2.5 HHz
RAM	1 GB	2 GB	4 GB	12 GB
Flash memory	16 GB	32 GB	32 GB	1 TB
Communication Standard supported	802.11, BLE	802.11, BLE, Zigbee	802.11, BLE	802.11, BLE
Operating system	Raspbian	Ubuntu 16.04	Ubuntu 20.04	Windows-10

4 Results

In this section, the prototype of the proposed system and the output of the various features of the proposed system are exhibited. The GUI of the smart home system is implemented using JAVA, and the algorithms used in the proposed system design are implemented using Python3 scripting language. The hashlib Python library is used to carry out a one-way hash operation. Figure 5a depicts the prototype of the proposed system in the laboratory environment along with the login page of the system, and Table 1 highlights the detailed specification of the hardware used. Figure 5b shows the different sensors connected to the end devices. Figure 6 depicts the real-time visualization of the sensor data. The details of the edge node, such as memory utilization, number of devices beneath it, and operating system details, are shown in Fig. 7. Real-time streaming GUI of the smart home (Living Room) system is provided in Fig. 8.

Time taken by the different edge nodes used in the proposed work to generate the dynamic key is given in Fig. 9. In this, the *x*-axis signifies the number of iterations



(a) **(b)**

Fig. 5 **a** Prototype implementation of edge-based IoT device monitoring and management system and **b** various sensors used in the proposed prototype

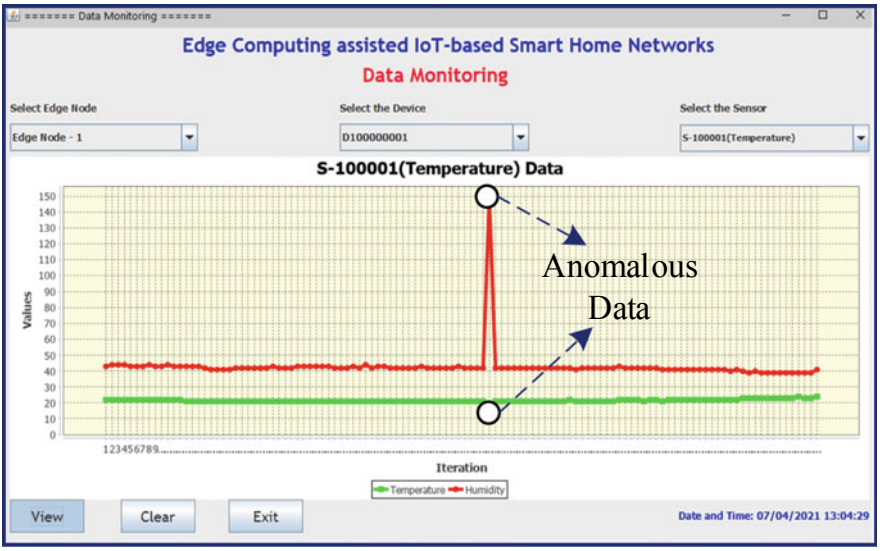


Fig. 6 Visualization of sensor values with anomalies detected

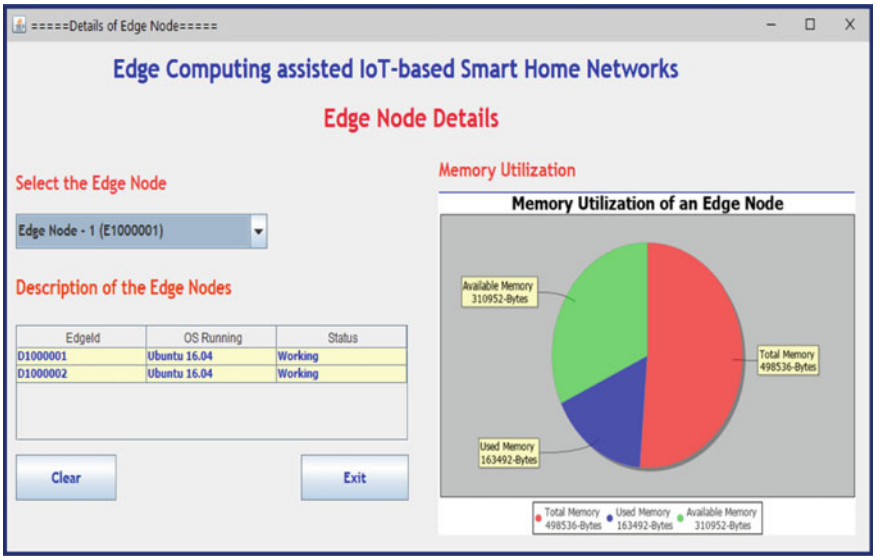


Fig. 7 Visualization of edge node details in smart home system

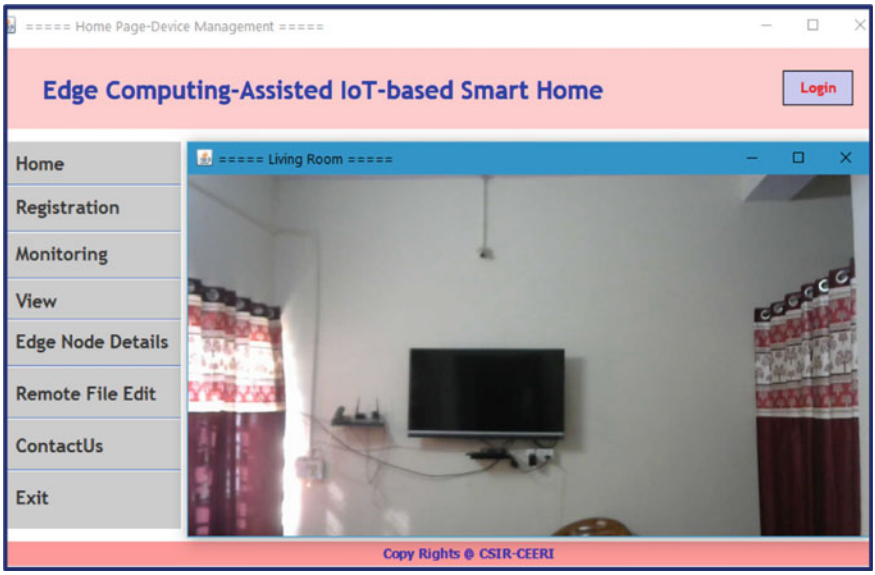


Fig. 8 Live streaming of the living room on smart home GUI

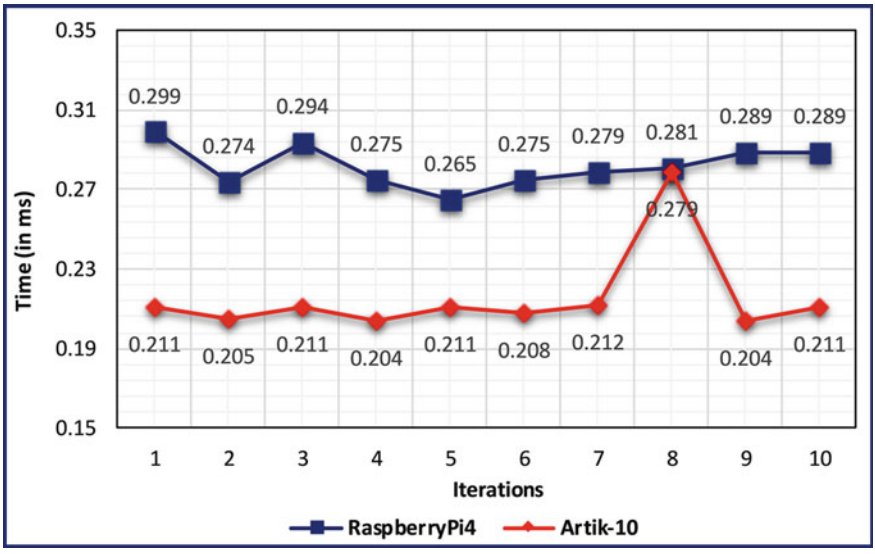


Fig. 9 Secret key generation time of Raspberry Pi4 versus Samsun Artik-10

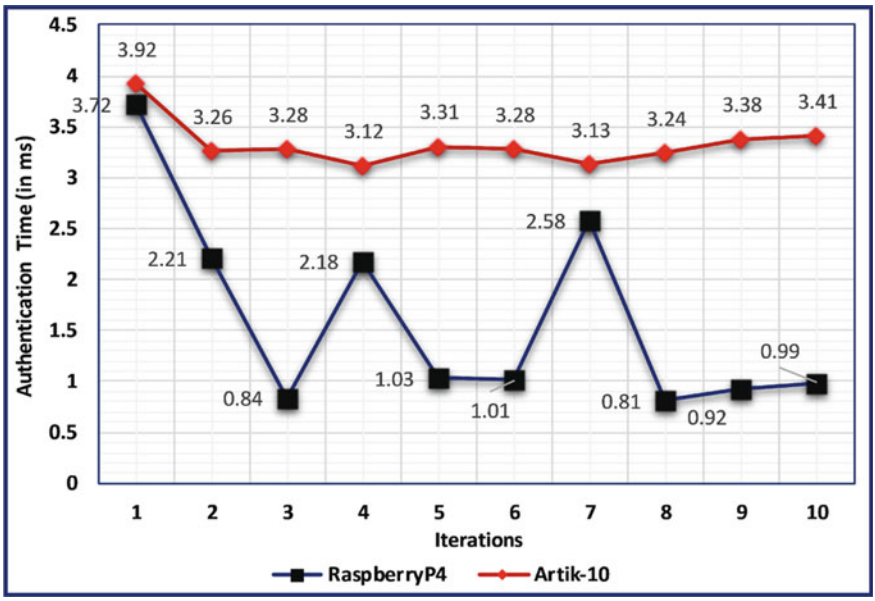


Fig. 10 Authentication time: Raspberry Pi4 versus Samsung Artik-10

performed during the conduction of the experiment, and *y*-axis labels the time taken by devices to generate the dynamic key. Artik-10 has shown better performance during the generation of the dynamic secret key generation. The communication cost of the algorithm at the time of dynamic secret key exchange of authentication process is about 63 bytes. One of the significant cryptographic operations used in the proposed system is hashing technique; Artik-10 consumes 0.118 ms to generate a 64-bit message digest. Similarly, Raspberry Pi4 requires 0.105 ms. During the registration process, four messages are exchanged between the server, end nodes, and edge devices. At the time of the authentication process, two messages are exchanged between the edge devices and end nodes. Figure 10 shows the authentication time of the algorithm. Here, the Raspberry Pi4 outperforms well. Figure 11 shows round trip time of the algorithms. Here, Raspberry Pi4 shows a better performance compared to Artik-10.

5 Conclusion

In this paper, we proposed and developed an edge computing-assisted IoT-based secure smart home system. The prototype of the system is demonstrated using heterogeneous hardware; the devices like Raspberry Pi4 and Samsung Artik-10 are used as the edge devices, and Raspberry Pi Zero and Raspberry Pi3 are used as the end nodes. The system provides the features like data storage, processing, monitoring,

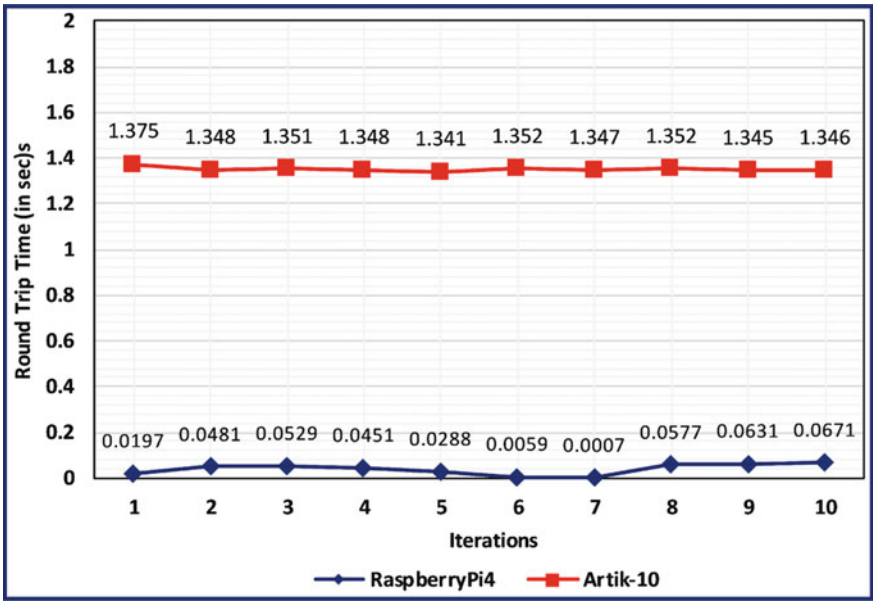


Fig. 11 Round trip time (RTT): Raspberry Pi4 versus Samsung Artik-10

and analyzing. The system sends the notification to the residents during unfamiliar situations. Also, it provides secure communication between the end devices and edge nodes using the simple matrix-based device authentication scheme. The use of edge computing in smart homes eliminates the complete relay on cloud computing. A detailed prototyping scenario is presented in this work; this prototype scenario may seek the attention of the researchers to develop multiple IoT services for multiple applications using edge computing.

The proposed method is limited to the creation of a prototype in the laboratory environment; this provides minimal characteristics like sensing and actuation along with the analysis of incoming data from sensors in order to identify the anomalous data. However, analysis of the live video captured from different regions of the smart home environment for multiple activities is not carried out. The major endowment to the smart home concept is video analysis. In future, the smart home solutions need to provide the feature like real-time object detection and tracking at the outdoors; this helps in identifying the fraudulent activities at the surroundings. Also, to enable the infotainment characteristic to make residents happier by analyzing the facial expressions is one of the requirements. Monitoring, tracking, and analyzing children's and elderly people activities make a caregiver's life easy. Developing these models at the edge of the network reduces the latency in responding to the particular action, and these models need to be tested in real time.

References

1. Hafidh B, Al Osman H, Arteaga-Falconi JS, Dong H, El Saddik A (2017) SITE: the simple internet of things enabler for smart homes. IEEE Access 5:2034–2049
2. Guillén E, Sánchez J, Ramos CO (2017) A model to evaluate the performance of IoT applications. In: Proceedings of the international multiconference of engineers and computer scientists 2017
3. Mocrii D, Chen Y, Musilek P (2018) IoT-based smart homes: a review of system architecture, software, communications, privacy, and security. Internet Things 1–2:81–98
4. Zaidan AA et al (2018) A survey on communication components for IoT-based technologies in smart homes. Telecommun Syst 69:1–25
5. Saito N (2013) Ecological home network: an overview. Proc IEEE 101:2428–2435
6. De Donno M, Tange K, Dragoni N (2019) Foundations and evolution of modern computing paradigms: cloud, IoT, edge, and fog. IEEE Access 7:150936–150948
7. Swamy SN, Kota SR (2020) An empirical study on system level aspects of Internet of Things (IoT). IEEE Access 8:188082–188134
8. Gunputh S, Murdan AP, Oree V (2017) Design and implementation of a low-cost Arduino-based smart home system, IEEE 9th International Conference on Communication Software and Networks (ICCSN), pp 1491-1495. https://doi.org/10.1109/ICCSN.2017.8230356
9. Mokhtari G, Anvari-Moghaddam A, Zhang Q (2019) A new layered architecture for future big data-driven smart homes. IEEE Access 7:19002–19012
10. Tao M, Zuo J, Liu Z, Castiglione A, Palmieri F (2018) Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes. Future Gener Comput Syst 78:1040–1051
11. Verma P, Sood SK (2018) Fog Assisted-IoT enabled patient health monitoring in smart homes. IEEE Internet Things J 5:1789–1796
12. Jabbar WA et al (2018) Design and fabrication of smart home with Internet of Things enabled automation system. IEEE Access XX
13. Mora N, Matrella G, Ciampolini P (2018) Cloud-based behavioral monitoring in smart homes. Sensors 18:1951
14. Chen M et al (2017) Smart home 2.0: innovative smart home system powered by botanical IoT and emotion detection. Mob Netw Appl 22:1159–1169
15. Xu W et al (2019) The design, implementation, and deployment of a smart lighting system for smart buildings. IEEE Internet Things J 6:7266–7281
16. Swamy SN, Sowmyarani CN (2018) Repeated data management framework for IoT: a case study on weather monitoring and forecasting. In: 2018 4th international conference on recent advances in information technology (RAIT). IEEE, pp 1–7. https://doi.org/10.1109/RAIT.2018.8389046
17. Wootton C (2016) Beginning Samsung ARTIK: a guide for developers (Apress, 2016)
18. Sneddon BJ (2020) Must read: the Raspberry Pi4 is here, and it's a BEAST. Available at: https://www.omgubuntu.co.uk/2019/06/raspberry-pi-4-specs-price-buy. Accessed 28 Jun 2021

Onion Price Prediction Using Machine Learning Approaches



Kunda Suresh Babu and K. Mallikharjuna Rao

Abstract In terms of monetary operations, rates of onion are the main influences. Unforeseen price fluctuations are symptoms of market uncertainty. Machine learning today offers massive tools for predicting commodity values to counter market uncertainty. In this article, we will focus on the machine learning (ML) method to onion demand prediction. The provision is done using the information received from the Agriculture Ministry of India. We utilized ML models for prediction, for example, support vector machine (SVM), random forest, Naïve Bayes, classification tree (CT), and network neural (NN). Then we tested and evaluated our strategies to figure out which strategy gives the greatest precision efficiency. We notice analog output in all our strategies. We try to determine how onion prices are preferable (low), reasonable (mid), and costly (high) through the above strategies.

Keywords Price fluctuations · Random decision forest · Onion price prediction · Machine learning

1 Introduction

Asia has several types of essential foodstuffs. Those were garlic, coriander, jalapeno, tomato, okra, onion, and carrot. Basic foods get to be an everyday necessity for any people living in the society, particularly in India. The increase and sudden drop in the cost of food staples can thus be a prevalent theme and impact other staple food. For example, Fig. 1 shows directly that from January 2015 until July 2018, the rates of certain essential food products have a similar trend. Prices for all commodities increased from September 2016 until April 2017 in Fig. 1. Price fluctuations can be affected by various parameters, including fuel oil prices and climate

K. S. Babu (✉) · K. Mallikharjuna Rao
School of Computer Science and Engineering, VIT-AP University, Amaravati, India
e-mail: suresh.20phd7019@vitap.ac.in

K. Mallikharjuna Rao
e-mail: mallikharjuna.rao@vitap.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_15

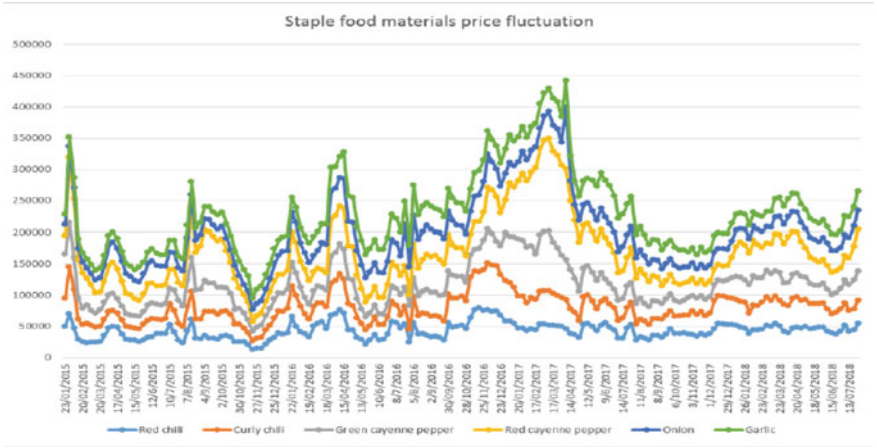


Fig. 1 Prices in India of certain basic food ingredients from January 2015 to July 2018 (red chili, curly chili, green cayenne pepper, red cayenne pepper, onion, and garlic)

conditions (snowfall, humidity, precipitation, political sustainability, exchange rates, global supply, etc.).

In our regular diet, onion is indeed a very important meal. Around 5 million tonnes of onions are required for one year in India. In India's current economy, sudden shifts in the cost of onions have become a key problem. India witnessed a dramatic shift in onion prices at the end of 2019. The cost of onions was 29 rupees per kg on February 1 in the year 2019, but it was 218 rupees per kg on December 17. Because of its uncommon behavior, this move gained considerable attention. Figure 2 shows the sudden price change in the onion market in the year 2019.

We figured out from the price that the variability rate was very high. This price is not acceptable for India's unprivileged people. Due to the uncertain nature and volatility of knowledge, financial prediction is complex. In addition, the predicted outcome is affected by variables such as climatic situations, efficiency, limited storage, shipping, and supply–demand ratio, making predictions more complex. Computer systems are operating smartly in this age of artificial intelligence. Many researchers employed devices that are used widely for the intent of forecasting. The most interesting approach for the implementation of ML is the financial sector. For this reason, we continue to forecast the price of onions by implementing the ML method to our captured onion price data. Many researchers demonstrated that numerous tools, such as Scikit-Learn, TensorFlow, Matplotlib, Pandas, and NumPy, are accessible for ML applications. Various feature selection techniques can be used to make the sample data easy to use. Predicting the cost of onions is not an easy job. The number of factors that affected the estimation was discussed earlier. By offering the best forecasting methods such as supervised learning algorithms, unsupervised learning algorithms, reinforcement learning algorithms, machine learning strengthens everyone. Since our study is a form of prediction, we chose a supervised

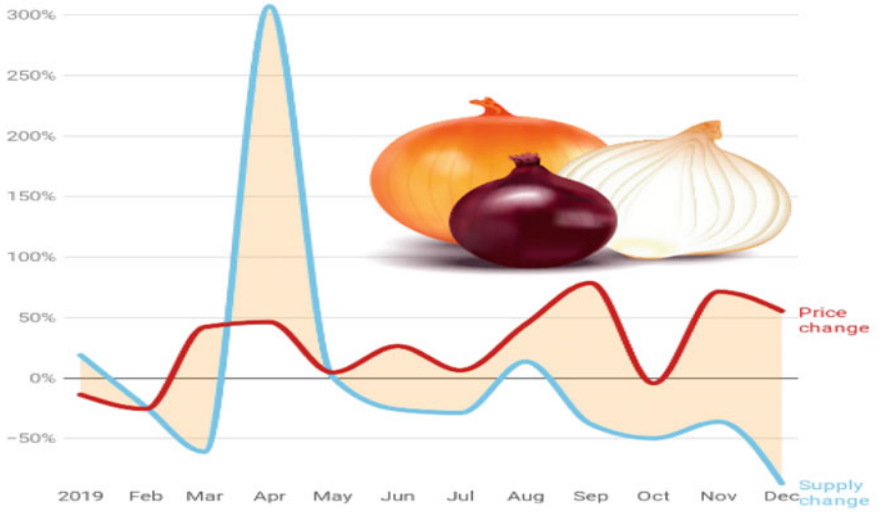


Fig. 2 Sudden price change of onion in the year 2019

learning algorithm to make forecasts. For predicting onion prices, the above ML methods can be applied. The main contributions of our work are as follows: (1) we tried to predict onion prices using ML methods like SVM, RF, NB, CT, and NN. (2) All the ML methods were evaluated using accuracy, precision, sensitivity, specificity, and misclassification rate. (3) Comparative analysis has shown that the NN model accomplished the highest accuracy of 98.17% by using a 30% data usage rate. (4) Experimental results have shown that the NN model has the least misclassification rate of 1.83% when the data usage rate is 30%. This article is structured as follows. A summary of related work is provided in Sect. 2. A suggested approach is given in Sect. 3. We notice desired results in Sect. 5 and evaluate different results that we accomplished to forecast onion prices through various ML algorithms. After this, with a summary concerning potential scope, we end our article.

2 Related Works

Asnhari et al. [1] employed autoregressive integrated moving average (ARIMA) with regression models like linear regression and Fourier model to predict the fluctuating prices of major food products. For major food staple onion, this is 96.57%; the maximum predictive performance that uses Fourier regression with ARIMA is achieved. The best precision for major food product red chili is produced to 99.84% using ARIMA with multiple linear regressions. They concluded that ARIMA with Fourier regression has been more efficient than multiple regressions with the ARIMA process, as the precision of Fourier with ARIMA is very robust, without disruption

of currency fluctuations. (Sugiarto et al. [2]) Clustering methods have been implemented to recognize a difference in food prices among Indonesia's natives. Findings have identified three cells by the Indian Provinces of Papua, West Papua, and merging the cost of meat, chicken, bread, beans, and chilies between other coconuts, which appear to be a bit bigger than other categories of the two. They investigated the optimum sum of product data scale population for the commodity in 34 provinces. They also published the results of numerous comparative experiments between clusters using the Tukey range test. However, the cluster value for goods demonstrated a far better average cost for eggs and red onion than other clusters. Furthermore, in grain, meat, poultry, pork, and Chile, the cluster had no major separate status that the entire category surpassed other types. No substantial mean differential was observed in particular for beef.

(Anggraeni et al. [3]) The purpose of this researcher's study is to estimate the cost of chili as a product to support the policy-maker sustains its stock. The approach included in this analysis was the artificial neural network by using input parameters, i.e., the cost, output, and usage of chili by the user. The mean average percentage error (MAPE) value created by the model was 16.19%, which was deemed sufficiently accurate. (Pulido et al. [4]) They investigated the implementation of a variety of frameworks of neural networks to model and forecast the sophisticated economic time series of dynamic behavior. To correlate the findings, they have used different neural network models and training algorithms and choose at the end which one is better for this reason. These often associate the simulation observations with blurred philosophical models and the standard method to using a mathematical framework. They used true time batches of consumer product prices to test design. In the USA, real prices of tomatoes and green onion exhibit dynamic variations in time, and conventional techniques will forecast them quite difficult. For this reason, to simulate and forecast the evolution of certain prices in the US industry, they preferred a neural network approach.

Melin et al. [5] simulated and predicted prices of consumer goods in the US market by employing modular neural networks. To evaluate the accuracy, a monolithic modular neural network with distinct evaluation metrics was implemented and calculated in which the right alternative is modular networking results in monolithic networks. Modular neural networks' predictor performance was obviously dominated by mundane neural nets, and modular neural models for this sort of issue were the favored factor. (Takanashi et al. [6]) This study focused on Welsh onions' predicted disease infection model for leaf parasites on Welsh onions. Onions' rust fungus disease is the most typical disease, and the role of the probability density of Weibull is ideal to close down the infection rate. The system included the wetness period and relative humidity to estimate rust bacterium disease of Welsh onions. Producers can then use the model to decide on what day the rate of infection may rapidly rise, and then the required precautions for the disease might also be allowed. Thus, the defined predictive approach should apply to multiple conditions acquisition technique. (Ali et al. [7]) A machine-based scheme of assistance to individuals in the foodservice industry has been implemented to facilitate strategic decisions,

choosing solutions, addressing challenges, and preparing for the future. The framework was allocated into five main elements; (a) data management, (b) model the control scheme, (c) the power structures, (d) the database system of computer technology, and (e) the user interface. This system performed the following functions: firstly, traceability of content including deliverables to suppliers, quantities, dates, name, price, types, and numbers packages, etc., secondly, reporting for machines and product lines, and the third part concerns labor reports including the following: jobs, productivity, and changes, or sources of failure. Fourth, the security status, the later phase was a quality study that contains quantities, kinds, dates, and so on. A case study applied to the computer-based system of technical management aid to this plant has been used in the field of onion dehydration.

(Rosyid et al. [8]) They used exponential smoothing models to estimate the complexities in the trend of food prices. Their study was aimed to predict food prices at the regional level in Indonesia. The phases of analysis consisted of (1) past performance, (2) preprocessing (3) fragmented databases, (4) prediction, and (5) confirmation and assessment. To estimate potential food price projections, the test regime is set up by scenario 1 and scenario 2. Scenario 1 generated a MAPE value of 3.08%, and scenario 2 achieved 8.24%. This was an assessment system. The degree of precision is very high as the product of an error rate under 10%. Since the prospecting cycle has been shorter, scenario 1 gains stronger performance, so the triggers of variables impacting markets are comparatively more permanent and produce a greater degree of data relevance between forecasted data and real knowledge. (Duanduan [9]) They tried to form a role model for a prediction of the He Nan Province variability pattern of the consumer price index. The index of consumer prices would increase or decrease from 2011 to 2015, and this would give an outline of steady and increment, in general, according to the model. In addition, the study stated so many variables likely to influence a price index for consumed products, and determined the growing prices of the two major parts in the consumer inflation figures, foodstuffs, and domicile goods, which is the chief cause impacting the consumers market price. Furthermore, the coverage impact and the microeconomic influence by the national government also add to the overall sustained growth rate for Province He Nan.

(Ye [10]) There was an investment prediction trend focused on wavelet analysis and ARIMA-SVR. By decomposition of wavelet and wavelet reconstruction, the equality rate is broken into reconstructed components and fault components. When forecasting the repaired portion and incomplete section, both the ARIMA probe and SVR simulator are used, and the final prediction effects will be calculated. The trading days cost for the software simulation was chosen as the experimental data between January 6, 2016, and January 30, 2017, totaling 270, of which the first 240 were the training set and the last 40 the test set. Exploratory analysis shows that, as contrasted with a standard forecasted model, this is the real framework for asset price predictions, increasing dramatically the precision of the forecast. (Jin et al. [11]) They proposed a vegetable market forecasting model using the deep learning algorithm that was used to estimate monthly vegetable price data by using price data, weather parameters, and other data for head managing divisions. In Korean farming

production, this model was extended to Chinese cabbages and radishes. Product assessment results show that Chinese cabbages and radishes are predicted to have a 92.06–88.74% accuracy for the recorded vegetable pay model, respectively, for the style. As well as they were awaiting that the model could endeavor independently to shift profitable business model and to create respective politics to save the economic consequences with regard to agriculture yield.

(Apichottanakul et al. [12]) They provided neuron networking to forecast Thai rice as a shareholder of a market for the global market. There are two theories formed as a result of two hypotheses. First, according to exporting prices, profitability for rice in multiple nations is anticipated. In addition, only the export prices of rice sourced from such countries are found. Export prices shall be used concerning entry parameters; the reference may be assigned to the worldwide industry for Thai rice as the only commodity parameter estimate. The empirical knowledge generated from the Agricultural Department, American, through the result has been more valuable, with MAPE 4.69% and MAPE 10.92% averages of the model for 2005. (Setiyowati et al. [13]) The proposed models involve a non-stationary forecasting model using ARIMA and generalized auto regressive conditional heteroskedasticity (GARCH) rice price models. There are two attitudes, i.e., good and spiky, with a daily expense in rice. The non-stationary form was used to satisfy all tasks. Forecasting traditional and gap in rice price trends are developed to produce the overall rice rate for up to one day. Consequently, the prediction of the prices of rice will provide rice operators with deliberate and useful knowledge. (Fang et al. [14]) The image clustering-based and deep learning systems are proposed to reliably predict the next pattern in the fruit market. Initially, fruit cost is divided into several solid periods, at stages, two-dimensional graph image fittings, and each object segments into an entry curvature and the exit curvature. On the exit curvatures, K-type tags are also extracted by employing K-means clustering to form a training set. Ultimately, the training set is used to train many computational models. A comparison reveals that the neural network has more sophisticated fruit market predictability, than does the other two, where the predictive precision might not be reasonably large for the qualified systems to predict.

(Chen et al. [15]) Fuzhou cabbage was investigated as an instance, and the wavelet analysis (WA) has been used to minimize cabbage noise data. Then, with an improved normalization technique, normalize the information. Lastly, the structured data are delivered on artificial recurring neural network (ARNN) basis for the forecast. Excellent performance with a new model called WA-ARNN than the traditional ARNN model is achieved. Tests also suggested that the approach achieved high efficiency and clarity throughout. Matocq et al. [16] provided knowledge framework for small-scale onion producers which includes concepts like weekly rates charged in major regions of the production of onions, market price, revenues of dry onions, etc.

Many experiments have been carried out for predictive purposes in the area of ML. To overcome the associated uncertainty, an overwhelming series of experiments have been carried out to forecast the future prices of products. Such experiments have addressed the issues and applied machine learning methods to solve the issues. Many researchers succeeded by using a model focused on ML to predict the price

of onions. This paper demonstrated the dependence of the expected outcome on the nature of input data. Three techniques were used to pick attributes. Until predicting the data set, four categories are usually divided into training and test sets results. Finally, for precise estimation, the researchers developed an ensemble model. The approach to ML works productively on quantitative series data results. It is therefore a good place for us to estimate the price of onions through ML. Most analysts used Indian SVM algorithms to find the market and availability of the building resources of paper wood products in Maharashtra [17]. Wood products market prices were determined by a fourteen-year statistical compilation and analysis of timber. The analysts found market supplies for the next year after evaluating the information gathering. Production was wood products shortfall. In addition to SVM, our study could provide more algorithms and compare the precision. Research on onion price prediction employing statistical approaches like ARIMA, generalized ARIMA, and seasonal ARIMA can tackle general linear problems but have the limitation that the performance is not stable and consistent for nonlinear price series. The typical drawback of statistical approaches is that they cannot evaluate the non-stationary or nonlinear correlation of time series data or manage multiple input parameters. So, statistical approaches are not considered in this work.

3 Proposed Method

Information gathering, statistical analysis, implementation, and assessment are qualitative research design of our research which is shown in Fig. 3. The proposed system design is shown in Fig. 4.

A. *Information Gathering*

We gathered the information we wanted from the Web site of the Ministry of Agriculture. We obtained 730 specimens of the cost of onions daily. Such data is computational-related, unorganized, and time series data. We established the following actions to focus on this info. The different characteristics of the data set considered for modeling are price attributes, production data, and meteorological data. Few data set features along with descriptions are given in Table 1.

B. *Statistical Analysis*

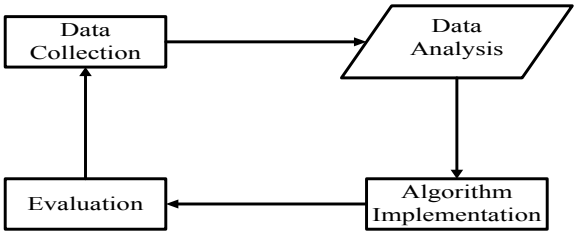


Fig. 3 Evaluation methodology

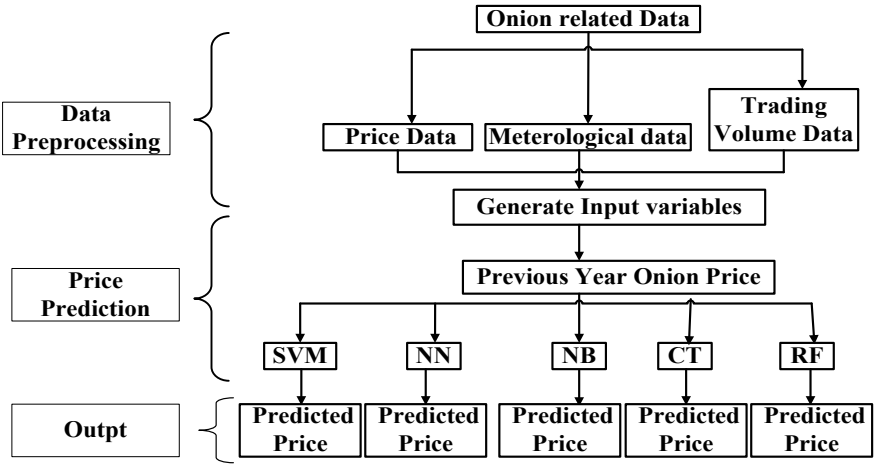


Fig. 4 Proposed system design

Table 1 Features and description of the data set

Feature type	Feature name	Description
Price	P_P	Present price
	Mean_Price	Mean total price difference monthly
	Price_Var	Price variation from last month
	Prev_Price	Previous month price
Production	Cum_Mon_Trade	The cumulative amount of monthly trade
	Diff_Trade	The difference in trading volume
Meteorological	Mean_Temp	Mean total temperature of the month
	Cyc_Adv	No. of cyclone advised days
	Cyc_War	No. of cyclone warned days

We compiled our information through organized formats before implementing methods. We regarded month, year, place, period, date, cost, and type as metrics that are a differentiated feature inferred from cost. Information was not in a healthy state in our gathered database because the database was blended with unnecessary information. Before implementation deployment, we had to delete unnecessary data from our data set. To this end, we utilized the famous Panda ML platform to ignore undesired information to construct the preferred data frame.

After categorizing the cost into these three groups, we illustrated it as minimum, intermediate, and maximum in the chart. The orange bars reflect the minimum price, the intermediate price is green, and the maximum price is the red bar. From Fig. 5, we can quickly calculate how many times a year the cost was minimum, intermediate, and maximum. Figure 6 depicts the cost of onion for two years 2018 and 2019, where we displayed the year on the *x*-axis and the cost of onion on the *y*-axis. Figure 7 displays the line graph reflecting the quarter on the *x*-axis and the associated cost on the *y*-axis. We obtained actionable insights from this chart through data interpretation. We found that the cost of onions was maximum at the starting of the year; at the ending of the year, the cost would be minimal. The cost of onions was poor after the starting of the succeeding year, and we also found that the cost would be maximum after this year. We note that this system was still preserved in the preceding year. Because of this framework, we considered the year code as a fixed attribute that enabled us more reliably estimate the cost of onions.

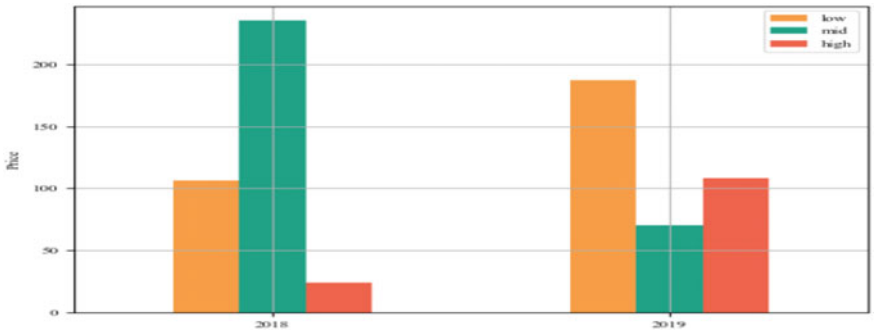


Fig. 5 Annual cost variation



Fig. 6 Monthly cost of onion

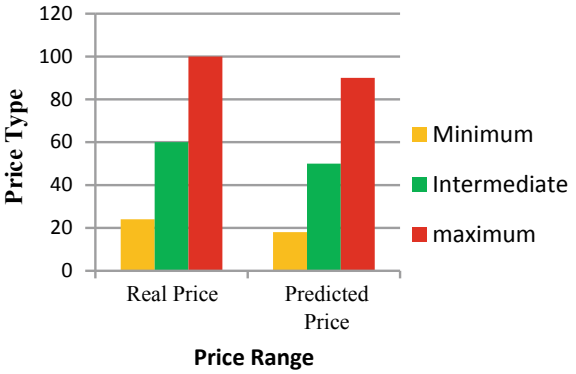


Fig. 7 Compare the true price and a predicted price

4 Implementation

Initially, we grouped the onion pricing into three groups and then implemented a supervised learning model more directly to classification techniques. Now we have the preprocessed data set for algorithm implementation to complete the data analysis portion. In this part, we used each method point by point and examined the effects of these implementations using the model shown in Fig. 8. Then we measured the precision of these implementations. We utilized various ML models to estimate the cost of onion. These algorithms are SVM, random decision forest, Naïve Bayes, CT, and NN. We constructed a data set at the outset of our study, and then we applied five separate ML methods. Each methodology produced useful outcomes for the prediction that were almost intended. We will focus on the outcome for further evaluation by solving this mechanism in the trial outcome portion. The SVM algorithm used for training the data set is given below.

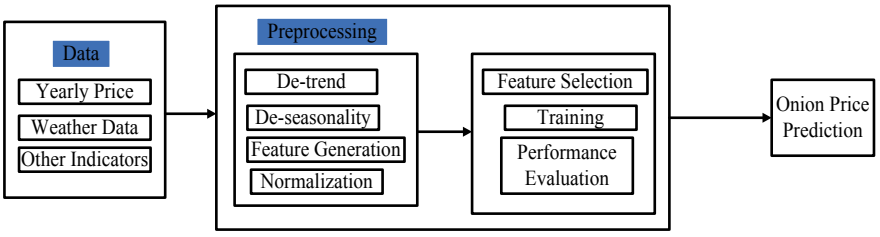


Fig. 8 Propounded prediction model

```
Algorithm SVM_Training_Process(P,Q,N)
{
Input: D← [P, Q]; //P defines input array of 'z' parameters and Q defines class labels array.
Q← array(C) // Class label
N← number_of_runs
l← learning_rate
e← error
Output: Evaluation of the model
Initialize:l←Math.random( );
for l in N
        e←0;
        for j in P
        if (Q[j] *(P[j]*g))<1 then
        overwrite : g←g + l * ((P[j]*Q[j])*(-2*(1/N)*g)
        otherwise
        overwrite: g←g+l *(-2*(1/N)*g)
        end if
        end
end
}
```

5 Results and Discussion

Using the chosen technique, the consistency of the implementations was evaluated and placed in a more efficient equation in the precision table on a compiled data set. Table 2 displays the predicted estimates containing the information consumption rate of columns, which is between 30 and 70%. Five simulations with related findings are seen in the following column. The yellow box arrives at the greatest precision of an identical model. The benefits of KNN were clarified by Keller et al. [19]. This system is most commonly used for ML since it is very simple to build, and the model is sub-optimal. They found forecasting accuracy 88.58% when the sum of data use had been 30% for the KNN model. Patil et al. [19] defined how using a definition combined to each class of a given data set Naïve Bayes forecasts. The Bayes principle uses this mechanism to grant a description. We attained 68.84% accuracy at the average data use rate when adding Naïve Bayes to our data set. The decision-making tree displays outstanding features while the collection is complex and was considered by Safavian et al. [20]. When the data usage rate is 30%, we got 97.72% accuracy using the classification tree. The SVM is the ideal faculties for financial forecasts. Basic knowledge about SVM was identified by Pradhan [21]. At a 40% data use rate, SVM has 80.14% accuracy of data quality. An excellent option for classification NN is to resolve the dilemma. Kangarani et al. [22] said NN is being used mostly to establish dynamically complex foretelling features, and the NN is far more consistent with results. In our work, the NN model accomplished the highest projection rate 98.17% by using 30% of knowledge provided in the black border box. An illustration of the real and estimated price is seen in Fig. 9. We calculated about 60 pairs in the current

Table 2 Performance analysis of various algorithms

Metric	Data usage rate	Algorithms				
		Random Forest	Naive Bayes	Decision tree	SVM	NN
Accuracy	30%	88.58%	68.49%	97.72%	77.17%	98.17%
	40%	87.67%	68.84%	95.89%	80.14%	92.47%
	50%	84.84%	67.67%	96.16%	74.25%	97.53%
	60%	83.47%	67.81%	97.03%	76.94%	97.95%
	70%	80.04%	67.71%	95.30%	70.65%	87.28%
Misclassification	30%	11.42%	31.51%	2.28%	22.83%	1.83%
	40%	12.33%	31.16%	4.11%	19.86%	7.53%
	50%	15.16%	32.33%	3.84%	25.75%	2.47%
	60%	16.53%	32.19%	2.97%	23.06%	2.05%
	70%	19.96%	32.29%	4.7%	29.35%	12.72%
Sensitivity	30%	90.15%	72.16%	97.99%	81.23%	98.09%
	40%	90.12%	76.43%	95.43%	78.12%	95.62%
	50%	83.12%	67.12%	95.12%	73.23%	96.15%
	60%	83.17%	66.61%	94.63%	75.99%	96.95%
	70%	79.04%	67.11%	95.22%	70.34%	87.12%
Specificity	30%	88.18%	65.16%	95.12%	76.13%	96.76%
	40%	86.43%	66.43%	94.54%	79.34%	91.23%
	50%	83.13%	64.23%	95.45%	73.12%	96.56%
	60%	82.22%	65.11%	96.03%	73.94%	97.32%
	70%	89.04%	64.31%	94.13%	72.45%	85.21%
Precision	30%	87.34%	67.54%	96.34%	76.43%	97.12%
	40%	86.75%	66.34%	95.45%	81.23%	96.34%
	50%	82.14%	66.13%	95.14%	73.45%	96.99%
	60%	83.33%	66.12%	92.03%	74.94%	96.12%
	70%	81.14%	62.61%	94.45%	74.15%	86.31%

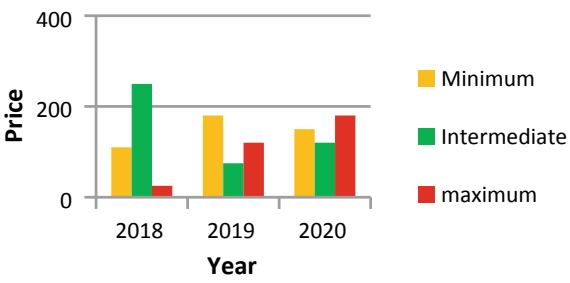


Fig. 9 Annual price variation

data set (May, June-2020) for analysis and expressed the rating scale in a bar chart. In its actual price component represents by the red rectangle, the chart generates exorbitant prices because every day of May, June, we are priced by elevated. So we intend to see quite a hefty premium in our outlook. Almost all the higher prices shown were generated by the forecasting element as depicted by the red rectangle with only a few average prices shown by the green rectangle. On the forecast variable, in which the elevated cost price is expressed by a red rectangle is 81.4% while the average value price, represented by a green one, is 18.6%.

Table 2 shows the prediction outcomes. In ML, the data usage rate is defined as the ratio of utilized data for ML to total data, which is changed from 30 to 70%. The yellow color cells indicate the outcome of each approach with the greatest accuracy rate. The cell in the black border box filled with yellow color displays the greatest accuracy rate of the five techniques. Naive Bayes' prediction rate is usually poor. The higher the rates of data use in random forest, the lower the rate of prediction is. There is no obvious connection between data utilization rate and prediction rate in the SVM and decision tree. The red color cells indicate the outcome of each approach with the least misclassification rate. The cell in the black border box filled with red color displays the least misclassification rate of the five techniques. The orange color cells indicate the outcome of each approach with the greatest sensitivity rate. The cell in the black border box filled with orange color displays the greatest sensitivity rate of the five techniques. The green color cells indicate the outcome of each approach with the greatest specificity rate. The cell in the black border box filled with green color displays the greatest specificity rate of the five techniques. The blue color cells indicate the outcome of each approach with the greatest precision rate. The cell in the black border box filled with blue color displays the greatest precision rate of the five techniques.

6 Conclusion and Future Scope

In future work, we will perform investigations to reduce high volatility by incorporating some of the new data set features into the predicted model that impact the fast rise and decrease in onion prices. Furthermore, we will also carry out the study on predicting onion production through the use of import/export volumes data to stabilize onion prices. Every methodology fits strongly in our research. Five methods have been studied, and the best method can be used for a healthy forecast for the onion cost. Ultimately, by way of a formula, we can predict the prospective cost of onion. Based on this estimated price, onion market prices will be estimated, since we know demand–supply plays a major role in the balance condition of markets. But now if you can forecast the current onion availability based on this estimate, we can have an equilibrium condition in the onion industry that will help us eliminate onion market volatility. The key drawback of our work is uncommon data behavior and a small number of documents. In the coming work in the data set, we need a more mature technology. With the inclusion of other Andhra Pradesh cities and the prediction of

the cost of onion for all of Andhra Pradesh, we will maximize the amount of our information. Our analysis, based on its data set, can be utilized to predict the optimally effective pricing scope of a monetary good. We can also use the mathematical optimization term to align the onion supply chain with an ML solution to decrease the onion market's fragility.

References

1. Asnhari SF, Gunawan PH, Rusmawati Y (2019, July) Predicting staple food materials price using multi variables factors (regression and Fourier models with ARIMA). In: 2019 7th international conference on information and communication technology (ICoICT). IEEE, pp 1–5
2. Sugiarto D, Ariwibowo AA, Mardianto I, Najih M, Hakim L (2018, October) Cluster analysis of Indonesian province based on prices of several basic food commodities. In: 2018 third international conference on informatics and computing (ICIC). IEEE, pp 1–4
3. Anggraeni W, Mahananto F, Rofiq MA, Andri KB, Zaini Z, Subriadi AP (2018, November) Agricultural strategic commodity price forecasting using artificial neural network. In: 2018 international seminar on research of information technology and intelligent systems (ISRITI). IEEE, pp 347–352
4. Pulido M, Melin P, Castillo O (2011, July) Genetic optimization of ensemble neural networks for complex time series prediction. In: The 2011 international joint conference on neural networks. IEEE, pp 202–206
5. Melin P, Urias J, Quintero J, Ramirez M, Blanchet O (2006, July) Forecasting economic time series using modular neural networks and the fuzzy Sugeno integral as a response integration method. In: The 2006 IEEE international joint conference on neural network proceedings. IEEE, pp 4363–4368
6. Takanashi H, Furuya H, Chonan S (2007, October) Prediction of disease infection of Welsh onions by rust fungus based on temperature and wetness duration. In: 2007 IEEE international conference on control applications. IEEE, pp 325–330
7. Ali SA, Bahnasawy AH (2011, September) Decision support system for technical management of food processing industries. In: 2011 international conference on internet computing and information services. IEEE, pp 20–24
8. Rosyid HA, Widiyaningtyas T, Hadinata NF (2019, October) Implementation of the exponential smoothing method for forecasting food prices at provincial levels on Java Island. In: 2019 fourth international conference on informatics and computing (ICIC). IEEE, pp 1–5
9. Duanduan Z (2011, August) Predicting price trends in Henan Province during the period of the twelfth five-year plan based on grey fuzzy theory. In: 2011 2nd international conference on artificial intelligence, management science and electronic commerce (AIMSEC). IEEE, pp 5699–5703
10. Ye T (2017, April) Stock forecasting method based on wavelet analysis and ARIMA-SVR model. In: 2017 3rd international conference on information management (ICIM). IEEE, pp 102–106
11. Jin D, Yin H, Gu Y, Yoo SJ (2019, November) Forecasting of vegetable prices using STL-LSTM method. In: 2019 6th international conference on systems and informatics (ICSAI). IEEE, pp 866–871
12. Apichottanakul A, Piewthongngam K, Pathumnakul S (2009, December) Using an artificial neural network to forecast the market share of Thai rice. In: 2009 IEEE international conference on industrial engineering and engineering management. IEEE, pp 665–668
13. Setiyowati S, Pasaribu US, Mukhaiyar U (2013, August) Non-stationary model for rice prices in Bandung, Indonesia. In: 2013 3rd international conference on instrumentation control and automation (ICA). IEEE, pp 286–291

14. Fang Y, Min X, Zheng L, Zhang D (2019, October) Fruit market trend forecast using Kmeans-based deep learning models. In: 2019 IEEE 10th international conference on software engineering and service science (ICSESS). IEEE, pp 628–633
15. Chen Q, Lin X, Zhong Y, Xie Z (2019, December) Price prediction of agricultural products based on wavelet Analysis-LSTM. In: 2019 IEEE international conference on parallel & distributed processing with applications, big data & cloud computing, sustainable computing & communications, social computing & networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, pp 984–990
16. Matocq G, Tapella E, Cattaneo C (2001, January) Information systems for smallholder agricultural producers: the MERCOSUR Onion's information net. In: Proceedings of the 34th annual Hawaii international conference on system sciences. IEEE, p 10
17. Anandhi V, Chezian RM (2013) Support vector regression to forecast the demand and supply of pulpwood. Int J Future Comput Commun 2(3):266
18. Keller JM, Gray MR, Givens JA (1985) A fuzzy k-nearest neighbor algorithm. IEEE Trans Syst Man Cyber 4:580–585
19. Goyal A, Mehta R (2012) Performance comparison of Naïve Bayes and J48 classification algorithms. Int J Appl Eng Res 7(11):2012
20. Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. IEEE Trans Syst Man Cyber 21(3):660–674
21. Pradhan A (2012) Support vector machine—a survey. Int J Emer Technol Adv Eng 2(8):82–85
22. Farahani MK, Mehralian S (2013) Comparison between artificial neural network and neuro-fuzzy for gold price prediction. In: 2013 13th Iranian conference on fuzzy systems (IFSC). IEEE, pp 1–5

Development of an IoT-Based Bird Control System Using a Hybrid Deep Learning CNN-SVM Classifier Model



R. PushpaLakshmi

Abstract According to Ministry of Agriculture and Farmers Welfare report, 63 bird species out of a total of 1364 were found to be involved in causing damage to various agricultural and horticultural crops at various stages of growth. Firstly, this work aims to classify the main bird species that causes crop damage based on deep learning by capturing facial landmarks using a convolution neural network (CNN) and optimization based on R-CNN with support vector machine (SVM) classification. Here we extract facial landmarks from the bird images captured in farm. Finally, we train the SVM classifier with the output of the CNN model along with optimized features. The proposed architecture is trained and evaluated on bird species data set available in Kaggle data port. The second part of the work employs IoT devices to produce predatory sound that will entice birds away from the area, using a megaphone and MP3 module.

Keywords Convolution neural network · Optimization · Support vector machine · Deep learning · Internet of things

1 Introduction

Nearly half of the population in India mainly depends on agriculture for their income [1]. 70% of rural households still depend on agriculture as their primary source of income, with 82% of farmers being small and marginal [2]. According to a study released by the Ministry of Agriculture, avian species cause losses in agriculture by destroying crops during the sowing, seedling and ripening stages, resulting in

R. PushpaLakshmi (✉)
Department of IT, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu 624622, India
e-mail: push@psnacet.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_16

economic losses for farmers. It described 1364 bird species from 19 families that caused damage to a variety of crops. Cereals were attacked by 52 bird species, pulses were attacked by 14 bird species, oilseeds were attacked by 15 species, and fruits were attacked by 23 species [3]. However, 46 bird species, including bluebirds, chickadees, swallows, owls, hawks, and woodpeckers, have been reported as beneficial to agriculture. More recently, existing business technologies such as remote sensing, the Internet of things (IoT), deep learning, data mining and robotics platforms have been applied to agriculture, resulting in the idea of smart farming. Smart farming is critical for addressing agricultural production's efficiency, environmental effects and food security [4]. To solve these issues, images make up a significant portion of the data gathered by remote sensing. Images can provide a complete image of agricultural conditions in many ways, and they can help with a number of problems [5]. As a consequence, analysing images is a main part of agriculture research. So diversified data analyzation approaches are adapted in various agriculture applications to identify images, classify them and detect anomalies [6].

Deep learning (DL) is one of the most recently used methods for image analysis. DL is a machine learning technique that is similar to artificial neural network (ANN) but has a higher learning capability. As a result, it has higher classification accuracy. In DL, properties are automatically extracted from raw data [7]. Depending on the network architecture used, DL encompasses numerous different components including different layers and functions. While processing images, DL eliminates the obligation of feature engineering, which is a time-consuming and complex operation [8]. In this paper, we propose an IoT-based bird control and analytics framework that uses a hybrid deep learning CNN-SVM classifier model to confirm the existence of birds in the field by image processing using deep learning. The proposed bird management system uses a convolutional neural network (CNN) skilled using DL algorithm to classify different birds that cause harm to agriculture. It uses Internet of things (IoT) devices to produce predatory sound in order to improve crop monitoring and avoid crop losses due to bird damage. Related research works are expressed in Sect. 2. The proposed work and the results are conveyed in Sects. 3 and 4. The article comes to an end with conclusion in Sect. 5.

2 Related Work

Yoshihashi et al. [8] used time-lapse images to track birds in the vicinity of windmill. In time span-based picture making method, the frames/second taken differs within displaying and capturing images. The system uses a fixed camera and Haar-based property separation, adaptive boosting (AdaBoost) and histogram of oriented gradients (HOG) classifier. The method is only moderately effective when evaluated on hawks genre and crows breed. Rosa et al. [9] investigated ML process in oceanic trackers to detect and recognize stuff automatically. The analysis of the algorithm is quantified plying six algorithms. The algorithms performed poorly while grading the bird genre. In [10, 11], the bird family was perceived using video material. But

still, it has been noted to trigger fallacious signal in large bird species in high wind conditions, and it lacks an automatic species recognition algorithm.

The automatic bird identification system proposed in [12] applies CNN-based DL system for bird identification. It uses radar to detect flying birds, and it mainly focused on two key bird classes [13]. In [14], to prevent birds from damaging crops, the system used a solar-powered bird repeller device with an efficient bird scarer sound, which uses solar energy to produce loud noises that shock birds and other animals, causing them to leave the field. Miranda et al. [15] sorted pests by using DL process. VGG19 methodology was followed in discriminating 24 different pests from crop images. The authors of [16] used multiple CNN approaches to detect and classify 12 different pests. Rustia et al. [16] developed a remote greenhouse pest monitoring system using an IoT network and a wireless imaging system. The imaging method counts the insects on the trap sheet using a *k*-means chroma-based clustering and totalling blobs subroutine. The automated remote imaging system [17] proposed an integrated remote imaging device for crop safety, in which various trap segments were adapted to remotely control creature folks in farm fields.

The BirdCLEF 2016 challenge [18] introduced a new classifier trained by spectrogram that accrue preferable performance. The BirdNET algorithm tags differing birds predicated on their sound [19]. Reference [20] counts and maps birds by employing Polaroid. As a result, the aim of this paper is to automate the IoT-based bird repellent system and determine birds that cause losses to agriculture land with fewer errors. Our main contributions, which are threefold, help us achieve this goal. First, we use a deep neural network-based object detection model to help automate the process of identifying harmful birds. Finally, we use IoT devices to create predatory sound in order to monitor dangerous land bird movements.

3 The Proposed System

3.1 Hybrid Deep Learning CNN-SVM Classifier Model

Figure 1 exhibits the structure of the presented process. The bird species is detected using facial landmarks. The images are captured using camera fixed in agriculture land, and the faces are detected using Haar features. Digital image features called Haar-like features are employed in object recognition. Haar is a mathematical term that refers to a class of square-shaped functions known as wavelets. The image property Haar like makes use of image contrasts. According to Yoshihashi et al., Haar and CNN are the principal ways for discovery and cataloguing, respectively [21].

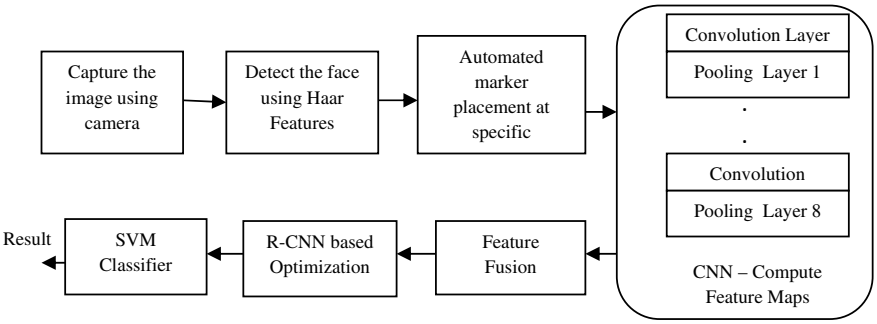


Fig. 1 Proposed CNN-SVM model

The CNN extracts various properties of the given image. In comparison with other classification algorithms, CNN requires substantially less pre-processing. The outlook of birds is learnt, and different sites of face are extracted. The extracted site information is normalized, and their properties are identified. Finally, the classifier utilizes the obtained properties to figure out the species of bird. The part-based R-CNN [22] is applied for optimization of feature selection. The optimization algorithm is used to improve the classifier result.

Facial Landmark Extraction and Normalization

Field marks on the head (eyebrow stripe, eyeline, whisker mark, colour of beak, throat, breast), field marks on the wing (wing bars, primaries, secondaries, speculum), tail and legs can all be used to identify birds, as described in [23]. Figure 2 depicts

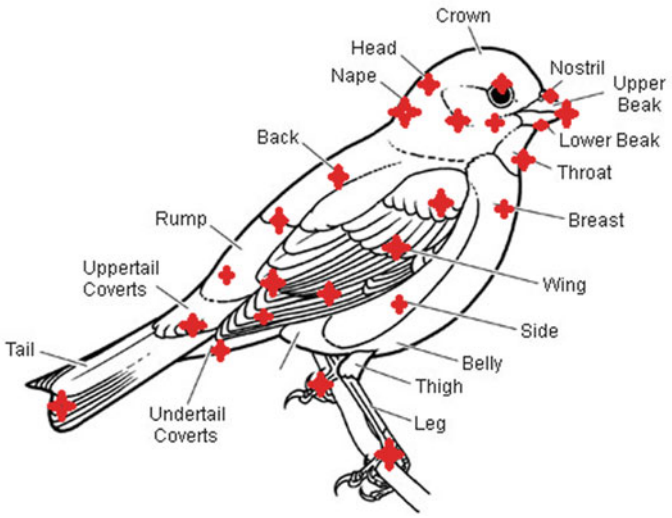


Fig. 2 Landmarks selected on birds

the exact locations of facial landmarks. It seems that the more the landmark locations in bird photos that can be found, the better the forecast is. Furthermore, the utmost significant criterion in developing a recent detection device is the ability to extract facial landmarks. Birds have distinct appearance having differences between classes. In [24], probabilistic model was used to trace different regions of bird. For part identification, Chai et al. [24] use symbiotic segmentation. Similar to [25], the exemplar-based model of [26] isolates birds region by enforcing consistence in posture and other subcategories. Recently, the methodology suggested in [22, 27] results in further improvement. We first normalize the selected properties to improve the performance of the proposed system. First and foremost, rotation is applied on each images. After rotation, the image is rescaled into 100 × 100 pixels.

Feature Selection

The data is transferred to an ANN for classification, as shown in Fig. 1. Our application's neural network is fed with 100 × 100 pixel colour images as input. The activation function mentioned in [21, 28] is opted. The function is depicted in Eq. 1.

$$y = \max(0, x) \rightarrow \partial f \partial x = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ Otherwise} \end{cases} \tag{1}$$

In the final layer, the softmax function shown in Eq. 2 is used.

$$\text{softmax } (z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \tag{2}$$

The eight stages of CNN with convolution and pooling layers are applied for facial landmark classification. The following are the settings for convolutional layers: $size(layer1) = 3 \times 3 \times 1$, $stride(layer1) = 1$, $filters(layer1) = 8$; $size(layer2) = 3 \times 3 \times 8$, $stride(layer2) = 1$ and $filters(layer2) = 16$; a max pooling and ReLU layer follow each convolutional layer. In the same kernel map, pooling summarizes the outputs of adjacent groups of neurons. With a stride of 2, we use 2 × 2 max pooling, which means the pooling regions do not overlap. The size of the final connected layer equals to the number of target categories, classified as harmful and harmless, which is then followed with a softmax function (Fig. 3).

The R-CNN approach of Zhang et al. [main] is applied for feature selection. This paper focuses on three main objects (head, tail, wing) of bird. The R-CNN assigns weights $\{w_0, w_1, \ldots, w_n\}$ for each part and has corresponding detectors $\{d_0, d_1, \ldots, d_n\}$. The detector score is calculated as given in Eq. 3.

$$\text{Detector Score on part } x, d_i(x) = \sigma\left(w_i^t \phi(x)\right) \tag{3}$$

where $\sigma(.)$—sigmoid function and $\phi(x)$—feature picked at region x.

The joint configuration of all the locations and parts is obtained by using Eq. 4.

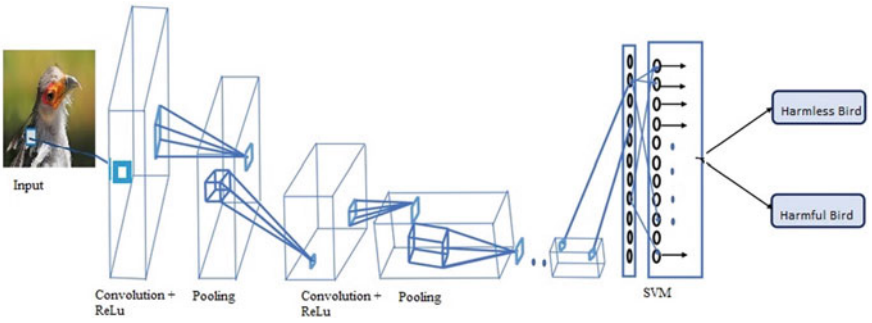


Fig. 3 CNN-SVM architecture

$$X^* = \arg\max_X \Delta(X) \prod_{i=0}^{n} d_i(x_i) \tag{4}$$

where $\Delta(X)$—scoring function of the object and bounding box $\Delta_{\text{box}}(X) = \prod_{i=1}^{n} c_{x_0}(x_i)$.

It takes into account any possible object window and selects a window having leading component rate. The proposed model is then skilled with SVM with the final property representation. The SVM classifier employs a positive and negative binary learner. The quantity of learners equals classes.

3.2 IoT-Based Bird Control System

The main aim of this model is to keep the birds from destroying the farmers' crop yield. On a raised platform, the model will be erected on the corners and in the middle of the field track. As a result, if we connect many cameras in sequence, we can cover a large hectare of land. The model's key feature is sound emission, which is used to scare the birds. The Arduino board, microSD card, MP3 module, audio amplifier and sound emitting system are the model's key components.

The SD card is a removable flash memory card that can be used to store data. It saves recordings of predators such as hawks, owls and other birds. An MP3 module is used to play the sounds stored on the SD card. To make the signals obtained from the MP3 module louder, an audio amplifier is used. Sound-emitting devices, such as megaphones, are used in various areas of agricultural land to keep birds from destroying farmers' crop yields. The main focus of this paper is on identifying birds that cause crop damage. So, only the fundamental concept of an IoT-based control system has been discussed. The CNN-SVM model's classified results will be stored in the Oracle IoT asset monitoring cloud service database, as shown in Fig. 4. Oracle cloud service will activate all of the registered devices in the IoT model. Oracle

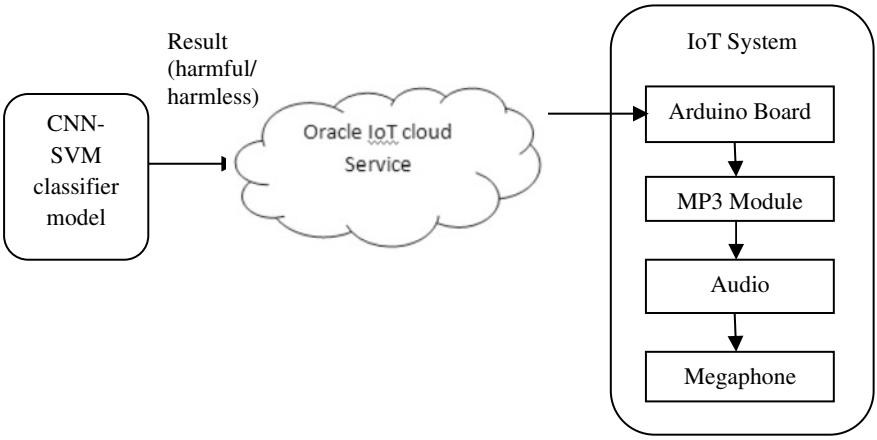


Fig. 4 IoT bird repeller model

Internet of things cloud service is connected to devices either indirectly or directly. Indirectly linked devices communicate with a gateway, which sends data to the Oracle Internet of things cloud service.

4 Experimental Results

This section discusses the performance of the proposed model. Reviews are carried out using the Kaggle website's commonly used benchmark data set of 265 bird species. The classification challenge requires computer vision systems to distinguish between 265 different bird species. 36,609 training images, 1325 evaluation images (five per species) and 1325 validation images (five per species) are included in the data set. All images are .jpg files with dimensions of 224 × 224 × 3 colours. A train set, a test set and a validation set are all included in the data set. We keep two classes for the sake of the experiment: dangerous and harmless. Each set has two subdirectories (harmful and harmless), each with 265 subdirectories, one for each bird species. The sample species of birds from the harmless class are depicted in Fig. 5.

First, we took the number of correct positives and negatives and then number of incorrect positives and negatives from the confusion matrix to calculate accuracy, precision and recall. The confusion matrix was then used to measure accuracy, precision, and recall. We provide a confusion matrix for the data set (Fig. 6).

The number of true positives for harmful birds observed is 460, with 220 true positives for non-harmful birds. The proposed detection and classification methodology performs admirably, with an accuracy of approximately 96.01%.



Fig. 5 Sample training set images of class: harmless

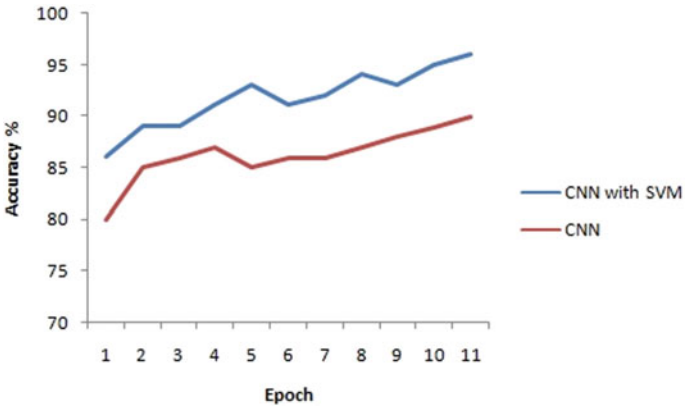


Fig. 6 Accuracy obtained

5 Conclusion and Future Directions

One of the most important tasks in a bird control system is detecting and classifying birds that cause harm to agricultural land. The proposed approach employs an IoT and deep learning-based bird detection and control framework that uses the CNN-SVM model to identify and classify bird species into two categories: dangerous and harmless. Oracle IoT cloud services are used to keep the harmful birds away from the land by means of raising predators sounds. The future extensions of this work involve investigation about the frequency of occurrence of various bird species over time in order to predict possible bird attacks on crops.

References

1. Indian Agriculture and Allied Industries Industry Report (2021)
2. India at a Glance Report, FAO in India (2021) Birds impacting agricultural crops a major concern, DownToEarth, DTE Staff (2019). https://www.downtoearth.org.in/news/agriculture/birdsimpacting-agricultural-crops-a-major-concern-64588
3. Tyagi AC (2016) Towards a second green revolution. Irrig Drain 65(4):388–389
4. Liaghat S, Balasundram SK (2010) A review: the role of remote sensing in precision agriculture. Am J Agric Biol Sci 5(1):50–55
5. Singh A, Ganapathysubramanian B, Singh AK, Sarkar S (2016) Machine learning for high-throughput stress phenotyping in plants. Trends Plant Sci 21(2):110–124
6. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
7. Amara J, Bouaziz B, Algergawy A (2017) A deep learning-based approach for banana leaf diseases classification. BTW Workshop, Stuttgart, pp 79–88
8. Yoshihashi R, Kawakami R, Iida M, Naemura T (2017) Evaluation of bird detection using time-lapse images around a wind farm. Wind Energy 20(12):1983–1995
9. Rosa IMD, Marques AT, Palminha G et al (2016) Classification success of six machine learning algorithms in radar ornithology. Ibis 158:28–42
10. Wiggelinkhuizen EJ, Barhorst SAM, Rademakers LWMM, den Boon HJ (2018) Bird collision monitoring system for multi-megawatt wind turbines, WT-Bird: prototype development and testing. https://publicaties.ecn.nl/PdfFetch.aspx?nr=ECN-M--07-048
11. Niemi J, Tanttu JT (2020) Deep learning–based automatic bird identification system for offshore wind farms. Wind Energy 1394–1407
12. Niemi J, Tanttu JT (2018) Deep learning case study for automatic bird ıdentification. Appl Sci 8(2089):1–15
13. Muminov A, Na D, Jeon YC, Lee C, Jeon HS (2017) Development of a solar powered bird repeller system with effective bird scarer sounds. In: International conference on ınformation science and communications technologies (ICISCT)
14. Miranda JL, Gerardo BD, Tanguilig BT (2014) Pest detection and extraction using image processing techniques. Int J Comput Commun Eng 3(3):189–192
15. Gondal MD, Khan YN (2015) Early pest detection from crop using image processing and computational intelligence. FAST-NU Res J 1(1):59–68
16. Rustia DJ, Chao JJ, Chung JY, Lin TT (2019) An online unsupervised deep learning approach for an automated pest insect monitoring system. In: Proceedings of the 2019 ASABE annual ınternational meeting, Boston, MA, USA, pp 1–5
17. Sprengel E, Jaggi M, Kilcher Y, Hofman T (2016) Audio based bird species identification using deep learning techniques. CLEF (Working Notes), 1609
18. Kahl S, Wood CM, Eibl M, Klinck H (2021) BirdNET: a deep learning solution for avian diversity monitoring. Ecol Inform 61(7):101236
19. Akçay HG et al (2020) Automated bird counting with deep learning for regional bird distribution mapping. Animals 10(7):2–24
20. Zhang N, Donahue J, Girshick R, Darrell T (2014) Part-based r-cnns for fine-grained category detection. In: ECCV, pp 834–849
21. Yoshihashi R, Kawakami R, Lida M, Naemura T (2017) Evaluation of bird detection using time-lapse images around a wind farm. Wind Energy 20(12):1983–1995
22. Bird ID Skills: Field Marks, All About Birds (2009). https://www.allaboutbirds.org/news/bird-id-skills-field-marks/
23. Wah C, Branson S, Perona P, Belongie S (2011) Multiclass recognition and part localization with humans in the loop. In: ICCV. https://doi.org/10.1109/ICCV.2011.6126539
24. Chai Y, Lempitsky V, Zisserman A (2013) Symbiotic segmentation and part localization for fine-grained categorization. In: ICCV. https://doi.org/10.1109/ICCV.2013.47
25. Liu J, Belhumeur P (2013) Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In: ICCV. https://doi.org/10.1109/ICCV.2013.313

26. Lin D, Shen X, Lu C, Jia J (2015) Deep lac: deep localization, alignment and classification for fine-grained recognition. In: CVPR. https://doi.org/10.1109/CVPR.2015.7298775
27. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors, arXiv:1207.0580 [cs.NE]
28. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: The 27th international conference on machine learning, pp 807–814
29. Verhoef JP, Westra CA, Korterink H, Curvers A (2018) WT-Bird a novel bird ımpact detection system. https://publicaties.ecn.nl/PdfFetch.aspx?nr=ECN-RX--02-055
30. Severtson D, Congdon B, Valentine C (2018) Apps, traps and LAMP's: 'Smart' improvements to pest and disease management. In: Proceedings of the 2018 grains research update, Perth, Australia, pp 26–27
31. Belhumeur P, Jacobs D, Kriegman D, Kumar N (2011) Localizing parts of faces using a consensus of exemplars. In: CVPR. https://doi.org/10.1109/TPAMI.2013.23

Social Media Analysis for Flood Nuggets Extraction Using Relevant Post Filtration



K. P. Aarthy, Kiruthya Ramesh, Vishnu Priya Prasanna Venkatesh, S. Chitrakala, and Shri. C. M. Bhatt

Abstract Floods are one of the most common and frequent natural disasters that affect the property and lives of millions of people worldwide. For better implementation of rescue operations, it is necessary to understand the depth of floods in localised areas. Stream gauging, satellite imagery and aerial photography are some common methodologies that are used in flood mapping. These methods are laborious and can only provide information about the coverage of flood. Social media being a huge repository of data can be used for extracting useful information regarding floods. However, not all posts in social media tagged with flood-related hashtags and keywords contain information relevant to flood. Filtering only the relevant posts helps in achieving better performance in depth estimation and location identification. Hence, we propose a novel approach to estimate the relevancy of posts with respect to flood before processing them further. Textual captions from social media posts can contain verbal descriptions of floods along with mentions of the location. Location is identified from text captions using named entity recognition. Depth information can be either directly stated with measurements or indirectly indicated using references which are identified for inferring depth in a location. The location metadata from social media posts is used for determining the location of the flooding event. The proposed flood nuggets extraction using relevant posts filtration (FERPF) algorithm devises a system to estimate flood water depth and location coordinates from the text captions and metadata from relevant social media posts filtered.

K. P. Aarthy (✉) · K. Ramesh · V. P. Prasanna Venkatesh · S. Chitrakala
Department of Computer Science and Engineering, College of Engineering Anna University, Guindy, India
e-mail: chitras@annauniv.edu

Shri. C. M. Bhatt
Scientist, Disaster Management Studies, Indian Institute of Remote Sensing, Dehradun, India
e-mail: cmbhatt@iirs.gov.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_17

Keywords Relevant posts filtration · Named entity recognition · Depth estimation · Text classification

1 Introduction

Floods are natural calamities that occur due to excessive overflow of water into dry lands. Flood mapping is a crucial element of flood risk management for it helps in estimating the level of water in a given location. There are many ways of collecting the necessary data required to estimate the floodwater level in different places such as stream gauge data, remote sensing data and field data collection. But these methodologies are labour intensive and highly risky. Moreover, they do not provide any accurate mentions of floodwater level in different locations.

Social media is a huge repository of data where millions of posts on various contents are tagged with topic-related hashtags. They contain textual captions accompanying image posts along with geospatial information linked with the location from where the post was made. Along with images, this metadata can also be used for deriving inferences regarding water levels and estimating location coordinates.

The FERPF algorithm proposed in this work is twofold. An integral part of the algorithm is relevant posts filtration which minimises unwanted computations and maximises accuracy by filtering the social media posts collected. Hence, only the relevant posts related to flooding events are considered further in the processing. From text captions, location information is extracted using named entity recognition (NER), and text captions are processed for depth mentions. Using metadata, location coordinates are extracted. By comparing the location from both text captions and metadata, precise location coordinates flooding event is inferred.

2 Related Work

Dirk Eilander et al. [1] proposed an approach to include wisdom of the crowd by using the observations posted on social media. In this approach, they look for direct depth mentions in the tweets by looking for mentions of numbers immediately followed by ‘m’ or ‘cm’.They looked for georeferences of locations in the tweet as well. This approach was tested as a pilot case for Jakarta floods by creating a database for the depth and locations identified from the tweets to create flood maps. Jing et al. [2] used text analysis in flood event estimation by using various resources and articles based on the Internet and combining the analysis with images for the estimation process. The text analysis was performed by collecting keywords and common flood-related terms for the flood estimation task. These keywords and terms are from the associated images which were related to the flooding event. The image features were combined with the text feature for the identification of a flooding event.

Smith et al. [3] explored the use of tweets during flooding events to identify the location of floods and the approximated level of floods in the area. They proposed the construction of a database with locations localised to a particular state of study and the usage of keyword and pattern matching to identify the depth mentions made indirectly in terms of verbal context. They proposed an integrated modelling framework based on social media. They used Twitter and identified tweets that may contain data about flooding. These messages are then processed to identify criteria to assess and thereby create a hydrodynamic model of the flood event. Their final objective is to create a simulation that represents the reported inundation within the city. These results can then be provided to authorities like emergency responders or local authorities. They proposed the usage of crowdsourced information like photographs and textual descriptions.

Middleton et al. [4] evaluated the best-of-class algorithms for location extraction by geoparsing tweets and geotagging Flickr posts in the benchmark data sets. Geoparsing from text is broadly classified as NER or named entity matching. To address location disambiguation, a set of possible location matches for a text token is taken, and the most likely ones based on available contextual evidence such as co-occurring mentions of other locations or post geotags are selected. Geotagging involves a combination of location identification and disambiguation, presented as a geotagging problem. Geotagging is applied to estimate the location of an image or video. To allow benchmarking against a commercial geocoding service, the authors developed an algorithm using simple NER and the commercial Google geocoder API. A JSON-formatted list of geocoded addresses representing possible matches that includes a longitude and latitude coordinate and well-formatted address with all the super-regions included is obtained as result. Elizabeth et al. [7] evaluated the use of social media as a platform to organise and manage volunteers during floods. They concluded that social media saved time and reached larger audience in comparison with traditional media. However, they highlighted the presence of irrelevant and inaccurate information that delays or hurdles emergency response.
de Bruijn et al. [8] created an early event detection and response system using social media to create a database for detecting floods in real times. They used geoparsing by applying a toponym-based algorithm to recognise locations and extract locations from tweets.

3 System Architecture

The proposed FERPF algorithm analyses the textual component in social media posts to identify depth-related information and extract location information using NER. It finds a precise location coordinate after consolidating the location extracted from the text and the metadata as well. To prevent irrelevant posts from being analysed, they are removed using the relevant posts filtration algorithm which uses a binary classifier to classify the text, before any further processing. The overall architecture

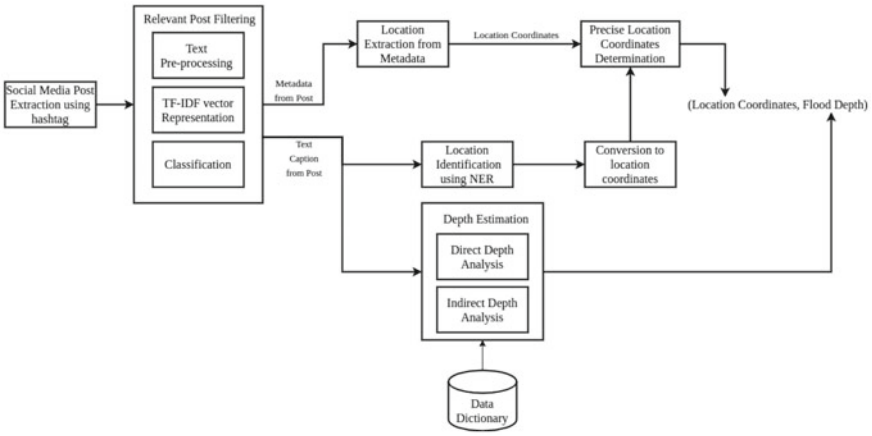


Fig. 1 Architecture of the proposed system

of the proposed system is shown in Fig. 1, and the flow is explained in detail as follows.

- The posts are first filtered by the relevant posts filtration algorithm which classifies them as flood related or not.
- The location mentions in the text captions are identified using NER. If the metadata also has location mentions, they are geocoded and the more precise one of both the location coordinates are identified.
- The direct and indirect depth mentions are identified from the text captions and converted into metric scale.
- The identified location coordinates of the area of flooding and the estimated depth are given as outputs.

3.1 Relevant Post Filtration

Existing works that extract depth or location-related information from social media do not include a filtration mechanism to discard non-flood-related tweets before using them for information extraction. In the proposed method, the posts are first checked if they are relevant to flood. It was seen that along with actual posts related to flood, few other categories like advertisements, memes, news articles, celebrities/individuals posting about floods but unrelated to the level of flooding and political opinions were also tagged along in flood-related hashtags. The training data was made to be representative and inclusive of the aforementioned non-flood categories by including sufficient examples from each of the categories. Social media posts might contain emoticons, punctuations and special characters. The text is first preprocessed before

being classified. The text has hashtags, other special characters and emoticons owing to the casual nature of the posts in social media.

The emoticons identified using their unicode ranges, special characters, newline, carriage returns and retweet mentions are removed. The stopwords from Natural Language Toolkit (NLTK) were used to remove the frequently occurring stopwords from the text. The sentences in the text are then lemmatised to bring them to their root forms. Feature extraction from the sentences is done using TF-IDF vectorization. TF-IDF is an occurrence-based representation. Flood-related posts are of different lengths, but they have commonly used words. When only the term frequency is considered, repetitive and common words can overpower the other relevant words. Hence, inverse document frequency is also considered in order to minimise this effect. The text is first converted to a TF-IDF vectorised representation and then classified using the random forest classifier.

The text converted to a TF-IDF vectorised representation is classified to obtain a binary decision on the relevance of the post. A random forest classifier with 400 estimators was trained to make a binary decision of whether the post is flood related or not. The posts classified as flood related are analysed further, and those classified as non-flood related are discarded from being proceeded further. Algorithm 1 shows the relevant posts filtration algorithm.

Input: Text Captions from Social Media
Output: Posts relevant to flood
begin
preprocessed text ⟵ removal of emoticons and special characters
preprocessed text ⟵ coversion to lowercase
lemmatized text ⟵ lemmatization of preprocessed text
text ⟵ TF-IDF representation of lemmatized text
class ⟵ Classify text using the trained random forest classifier
if *class is flood* **then**
| return text
end
else if *class is non-flood* **then**
| discard text
end
end

Algorithm 1: Relevant Post filtration

Table 1 shows the output of the text classifier to different categories of posts. The advertisement is rightly categorised as non-flood, and the flood-related post is classified as flood related.

Table 1 Outputs from the relevant post filtration

Input text caption	Output from relevant post filtration
Heavy rains in Kodambakkam, we are soaked up to knee-level. #chennairains #helpneeded	flood
Chiffon saree sale at just Rs.999. #chennaifloods #chennairains #sareesales	non-flood

3.2 Location Extraction and Identification

Some of the social media posts have the location of the post tagged in metadata. This metadata may contain the exact location of the flooding event tagged or in a higher level, and the city or state of the flooding event may be tagged. The caption in the social media post may contain location information. In order to identify this information from text, named entity recognition (NER) is used. NER aims at locating the named entities in a text and classifying them into their respective categories.

The available spaCy is a state-of-the-art open-source library for NLP in Python which is capable of recognising popular geopolitical entities (GPE). For identifying the local GPE in India, the existing spaCy NER model has been custom trained with Indian cities and places.

For custom training NER, sentences containing the local location names and other entities were used. The entities like ORG, PERSON and LOC were marked along with their position in the sentence. Thus, the spaCy NER model trained on this data set was able to predict the location names that are local to any specific region. The location name obtained is mapped onto its corresponding location coordinates using geocoding.

Geocoding is the computational process of transforming a physical address to a location on Earth's surface as numerical coordinates. For geocoding the identified location, the geopy Python library is used. The nomination geocoding service is built on top of the OpenStreet Map data. Since the location information is obtained through NER applied over the text caption and directly from metadata, the more precise location among these two has to be determined. This is done using precise location determination algorithm.

Upon analysing the location coordinates, it is found that, if an area, say 'area1' lies inside a particular area, say 'area2', then the latitude and longitude coordinates of 'area1' are smaller than the 'area2'. This is because 'area2' encloses 'area1' within it. By exploiting this fact, the precise location is determined by comparing the location coordinates obtained from metadata and text caption. If the latitude and longitude of location obtained from the metadata are smaller than the location obtained from the text caption, then the location coordinates from metadata are labelled as precise location. If the location coordinates from metadata are larger than that of from the text caption, the latter is labelled as precise location.

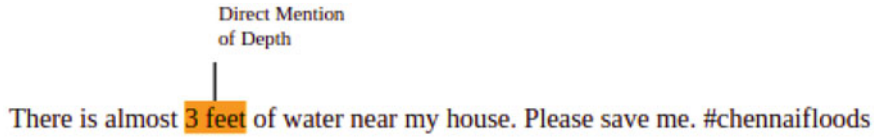


Fig. 2 Example for caption with direct depth mention

3.3 Depth Estimation

It is possible for text caption to contain mentions of depth which can be extracted for estimating flood water levels in different areas. There are mainly two ways in which depth information can be present in text captions, namely direct depth mentions and indirect depth mentions.

In direct depth mentions, the depth measure is directly mentioned within the captions. For example. in Fig. 2, there is a direct mention of depth, i.e '3 feet' which can be extracted for inference. Since there are several metrics like cms, feet, inches and metres along with their short forms that can be used for addressing depth, all the different metrics extracted are converted to a centimetre scale that is given as the final output.

Input: Text Captions from Social Media
Output: Estimated Depth
begin
if *caption has directDepthMentions* **then**
  extract depthMeasure along with measurement
  Convert depthMeasure to centimeters
  return depthMeasure along with measurement
end
else if *caption has indirectDepthMentions* **then**
  level ⟵ based on pointer found
end
if *levelModifier present* **then**
  increment or decrement level based on the levelModifier
end
depth ⟵ convert level to metric scale
end

Algorithm 2: Depth Estimation Algorithm

Depth of water is not always directly stated as a measurement. Oftentimes social media text captions use human body parts as pointers of references for indicating the depth of water. Knee, ankle, thigh, hip, waist and chest are a few. In Fig. 3, the presence of the word 'knees' is an indirect mention of depth information contained within the text. According to the scale mentioned in Chaudhary et al. [5], the different body parts mentioned are mapped onto an appropriate level. For example, as in the Chaudhary et al. [5], the word 'knees' is mapped onto level 4. Along with the mention

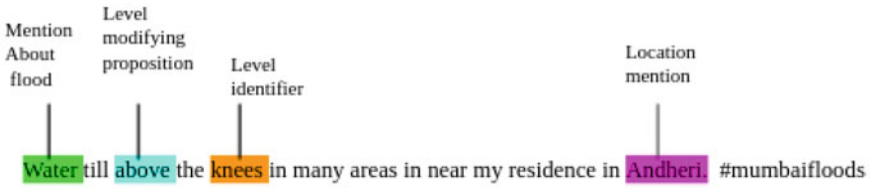


Fig. 3 Example for caption with indirect depth mention

of a level pointer, propositions within text captions like 'above','below', 'beneath' and 'higher' act as level modifiers where either the level identified is incremented or decremented accordingly. Table 2 contains a list of such indirect depth mentions and level modifiers. The final level is converted to centimetre scale based on the level scale given in Chaudhary et al. [5].

Input: Social media post
Output: Location coordinates of flooding event and flood depth
begin
relevantPost ⟵ relevantPostFiltration(hashtag) using Algorithm 1
locationFromMetadata ⟵ location coordinates from geotagged metadata of the social media post
caption ⟵ text caption from post
locationMention ⟵ identify location mention in caption through NER
locationFromText ⟵ geocode locationMention
if *locationFromMetadata not NULL and locationFromText not NULL* **then**
| preciseLocation ⟵ compare(locationFromMetadata, locationFromText)
end
else if *locationFromMetadata is Null* **then**
| preciseLocation ⟵ locationFromText
end
else if *locationFromText is Null* **then**
| preciseLocation ⟵ locationFromMetadata
end
level ⟵ depthEstimation(caption) using Algorithm 2
end

Algorithm 3: Flood nuggets extraction using relevant post filteration (FERPF) algorithm

The FERPF algorithm of the overall system as indicated in algorithm 3 works by first filtering and discarding the non-flood-related content. The relevant posts filtration algorithm (Algorithm 1) classifies the posts, and only the flood-related posts are considered further. Posts which do not have any depth or flood-related information are ignored from being processed further. The location coordinates obtained from metadata and from the posts using NER are compared to obtain a precise location. The text is then looked for depth mentions as in algorithm 2. The flood-level information obtained from direct or indirect mentions in the text is then converted to metric scale.

Table 2 Indirect depth mentions and level modifiers

Types of indirect depth mentions	Examples of indirect depth mentions
Level identifiers	'ankle', 'knee', 'hip', 'waist', 'neck', 'chest', 'head'
Increasing level modifiers	'above', 'over', 'more than', 'exceeding', 'higher than', 'greater than'
Decreasing level modifiers	'below', 'lower than', 'under', 'underneath', 'down than', 'lesser than', 'beneath'

4 Experimental Results

4.1 Data Set

The data set to train the relevant posts filtration model that does the initial filtering to remove unrelated posts was constructed by collecting posts related to flooding events from social media and microblogging websites like Instagram and Twitter by extracting data tagged with flood-related hashtags and labelling them for the classification task. While extracting the data, it was noticed that a significant amount of non-flood-related content was tagged along with hashtags related to flood. Figure 4 shows the distribution of non-flood-related content in a representative sample of data collected from social media. To train the classifier, around 7000 tweets belonging

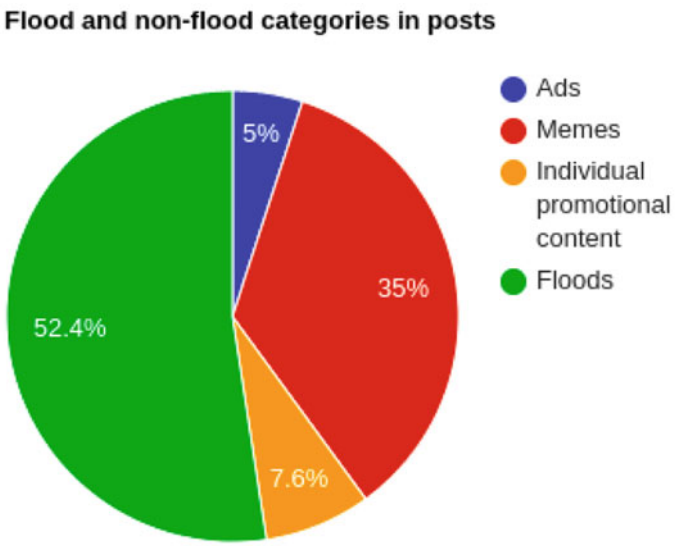


Fig. 4 Categories of contents in a representative sample of data

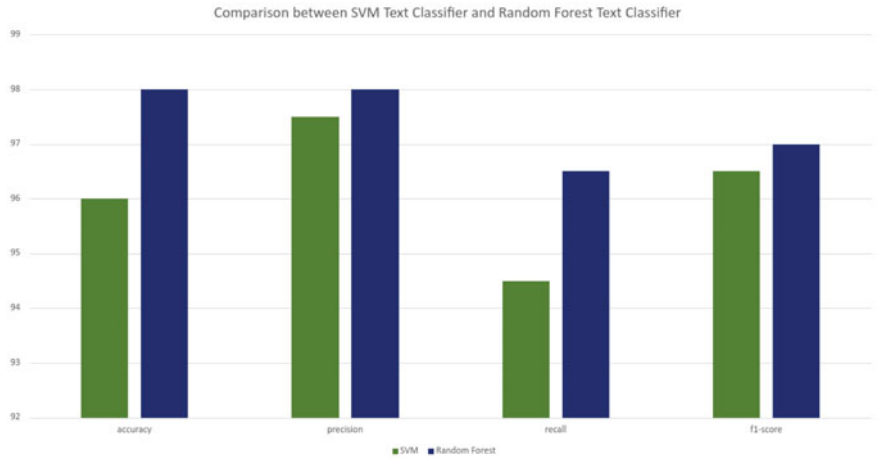


Fig. 5 Comparison of classifiers for relevant posts filtration

to flood-related and non-flood-related category were labelled. The NER model was trained with 1500 sentences to identify Indian locations by training it based on locations specific to Indian subcontinent.

4.2 Results

As an initial experimentation for the relevant posts filtration, a support vector machine classifier was trained and tested to classify the tweets to make a binary decision of relevance to flood. As seen from the comparison of the metrics for both the classifiers in Fig. 5, the random forest classifier performed better. The random forest classifier showed better performance as the number of gini estimators was increased. The improvement can be attributed to bagging and bootstrapping. Bootstrapping is a technique where the sampling is done randomly with replacements from the data set.

The text classifier is tested for precision, recall and accuracy. True positive (TP), false positive (FP), false negative (FN) and true negative (TN) cases are used to calculate these metrics.

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

$$F1 - score = 2\frac{Precision \cdot Recall}{(Precision + Recall)} \tag{3}$$

Table 3 Performance of relevant posts filtration model

Metrics	Values
Accuracy	0.983
Precision	0.98
Recall	0.965
F1-score	0.97

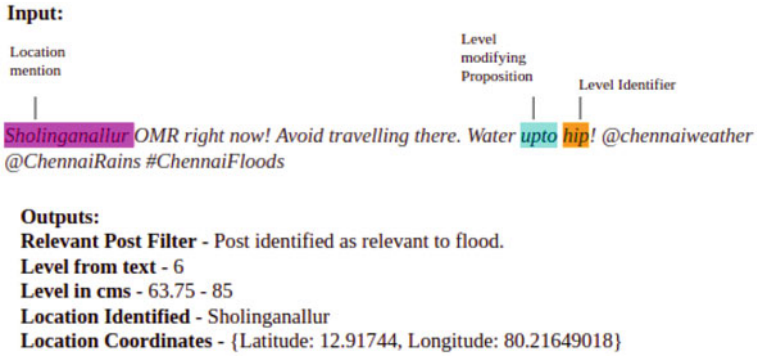


Fig. 6 Location and flood level estimated by the system

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{4}$$

Table 3 tabulates the performance metrics of the relevant posts filtration model. Figure 6 shows the output from the proposed FERPF system. The depth and location identified are given as outputs.

5 Conclusion

In this work, a novel algorithm was proposed to get the flood level and location of flooding events by processing social media posts. The relevant post filtration mechanism proposed in this work helped in improving accuracy and minimising unnecessary computation. A NER model is custom trained to find the names of local areas present in text. A dictionary-based mapping has been proposed to get the depth mentioned in text. The approach for depth estimation identifies both direct mentions where depth is directly mentioned and indirect mentions made contextually in the text. The high accuracy of the relevant posts filtration algorithm indicates that system behaves well in terms of classifying the relevant posts from other unrelated

posts found with flood-related hashtags. The location coordinates and flood nuggets identified by the system can be used in flood inundation maps and by emergency responders for rescue operations.

References

1. Dirk Eilander, Patricia Trambauer, Jurjen Wagemaker, Arnejan van Loenen (2016) Harvesting social media for generation of near real-time flood maps. Proc Eng 154:176-183, ISSN 1877-7058. https://doi.org/10.1016/j.proeng.2016.07.441
2. Jing M et al (2016) Integration of text and image analysis for flood event image recognition. 2016 27th Irish Signals and Systems Conference (ISSC), 1-6. https://doi.org/10.1109/ISSC.2016.7528454
3. Smith L, Liang Q, James P, Lin W (2017) Assessing the utility of social media for flood risk management. J Flood Risk Manage 10:370–380. https://doi.org/10.1111/jfr3.12154
4. Stuart E Middleton, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Yiannis Kompatsiaris (2018) Location extraction from social media: geoparsing, location disambiguation, and geotagging. ACM Trans Inf Syst 36, 4(40):27. https://doi.org/10.1145/3202662
5. Chaudhary P, D'Aronco S, Moy de Vitry M, Leitão JP, Wegner JD (2019) Flood-water level estimation from social media images. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci IV-2/W5, 5-12. https://doi.org/10.5194/isprs-annals-IV-2-W5-5-2019
6. Chaudhary P, D'Aronco S, Leitão JP, Schindler K, Wegner JD (2020) Water level prediction from social media images with a multi-task ranking approach. ISPRS J Photogramm Remote Sens 167:252–262. ISSN 0924-2716. https://doi.org/10.1016/j.isprsjprs.2020.07.003
7. Sharp EN, Carter H (2020) Examination of how social media can inform the management of volunteers during a flood disaster. J Flood Risk Management 13. https://doi.org/10.1111/jfr3.12665
8. de Bruijn JA, de Moel H, Jongman B et al (2019) A global database of historic and real-time flood events based on social media. Sci Data 6:311. https://doi.org/10.1038/s41597-019-0326-9

Fog Clustering-based Architecture for Load Balancing in Scientific Workflows



Mandeep Kaur and Rajni Aron

Abstract The rapid growth of intelligent devices accessing cloud data centers leads to network congestion and increased latency. Fog computing provides a ubiquitous and distributed environment in which different fog nodes are deployed near end-users to resolve cloud data centers' high latency problems, leading to reduced network traffic and latency. Nevertheless, it is a challenging task for fog layer resources to meet complex service quality constraints. Moreover, some objectives are considered to be achieved during the processing of complex and large workflow tasks, i.e., increased energy consumption, less utilization of resources. We have investigated the equal distribution of scientific workflow tasks among available resources to utilize resources and provide energy-aware load balancing properly. In this article, fog computing-based load balancing architecture has been proposed to enhance resource utilization in scientific workflow applications. We proposed a hybrid load balancing algorithm for optimum resource utilization in a fog environment. Our proposed algorithm improves resource utilization and reduces energy consumption as compared to the existing approach. For evaluation of the proposed approach, iFogSim has been used. The article concludes by providing directions for the future researchers.

Keywords Fog computing · Resource utilization · Load balancing · Scientific workflows

1 Introduction

Fog computing is providing cloud services to the end-users by deploying fog nodes near to them. Although it resolves the challenges of cloud computing, it has to face some issues while executing large complex tasks [6]. The problems faced by fog com-

M. Kaur (✉)
Lovely Professional University, Jalandhar, Punjab, India

Chandigarh University, Gharuan, Mohali, Punjab, India
e-mail: mandeep.e11416@cumail.in

R. Aron
NMIMS University, Mumbai, India
e-mail: rajni@nmims.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_18

puting involves load balancing, task scheduling, optimum resource utilization. This article mainly focused on resolving the load balancing issue faced by fog computing while executing scientific workflow applications. Scientific workflow applications include fine- and coarse-grained complex computational tasks with complex execution requirements [5]. These scientific workflow applications contain data transfer between large-scale organizations. A scientific workflow contains data-driven series of different business and scientific activities that includes task dependencies [1]. Scientific workflows involve complex and large computations, which means they require many resources to handle these tasks. These workflows have immense computational complexities, hence require an efficient framework and algorithm to reduce execution time and energy consumption. In a fog computing environment, fog nodes can be clustered to handle these complex workflow tasks.

In the past few years, workflows are becoming popular among several science disciplines, i.e., astronomy, meteorology, biomedical. Here are few examples of scientific workflows: CyberShake, LIGO, Montage, SIPHT [2]. Laser Interferometer Gravitational Wave Observatory (LIGO) workflows are used to observe and detect natural gravitational waves. The SCEC environment firstly used CyberShake to detect and measure earthquake hazards in different areas. This article proposed a fog clustering-based framework for the execution of scientific workflows. Along with this, a load balancing algorithm has been proposed for efficient distributing workflow tasks among all available fog resources to enhance resource utilization.

1.1 Our Contribution

This article has few major contributions that are described as follows:

- Fog clustering-based framework has been proposed for scientific workflow applications.
- A hybrid load balancing algorithm has been proposed to enhance resource utilization in a fog environment.

The rest of the paper has been organized as follows: Section 2 contains related work in this area. Section 3 proposes a fog clustering-based framework for scientific workflow applications. Section 4 describes the hybrid load balancing algorithm for fog computing. Section 5 contains the analyzed results and discussion. Finally, Sect. 6 concludes the article and provides future directions.

2 Related Work

Many researchers have explored the area of load balancing in fog computing and proposed several load balancing approaches. This section contains a review of different approaches provided by many researchers. De Maio et al. [2] proposed a Pareto-based task offloading approach for a fog computing environment named MOWO.

The authors aim to enhance reliability and reduce cost and latency. The proposed approach has been evaluated using real-world scientific workflows. Ijaz et al. [4] proposed a multi-objective scheduling algorithm to reduce energy consumption and makespan. They proposed workflow scheduling model for fog-cloud computing. The proposed approach has been analyzed on synthetic and real-world scientific workflows. The proposed approach reduces energy consumption to 50% compared to existing approaches.

Kaur [5] proposed an energy-efficient load balancing algorithm (EA-LB) for fog computing to evaluate scientific workflow applications. The proposed algorithm aims to reduce time, cost, and energy consumption while evaluating scientific workflows, i.e., GENOME and CyberShake. Simulation results are obtained by implementing the proposed approach in iFogSim and comparing their results with the existing approach. Tellez et al. [10] proposed a tabu search algorithm for load balancing in fog computing. The proposed approach aims to process received computational tasks immediately. The authors considered the bi-objective function considering the computational cost for both the fog and cloud environment. Synthetic tasks have been processed to obtain results. They mainly try to reduce computational cost and memory usage.

Particle swarm optimization (PSO) scheduling algorithm has been proposed for a cloud environment by Xie et al. [11] to reduce cost and makespan. Authors considered Montage, SIPHT, CyberShake, and Epigenomics workflows to evaluate their proposed approach and used WorkflowSim to obtain simulation results. Zhou et al. [12] proposed a workflow scheduling algorithm to reduce makespan and cost optimization in a cloud environment. The proposed approach is based on the fuzzy system. jMetal simulator has been used for evaluating real-world workflows. Serhani et al. [9] proposed cloud architecture for evaluating IoT workflow tasks. The proposed architecture keeps track of VM's current load, capacity, and state to overcome the overloading problem. The authors considered cost and time parameters to enhance resource utilization. To obtain simulation results, they considered Docker swarn cluster and PostgreSQL. Table 1 provides the summary of related work.

3 Fog Clustering-based Load Balancing Framework for Scientific Workflows

This section proposes fog clustering-based load balancing framework for scientific workflow applications. Execution of large computational tasks in fog computing requires a huge number of resources and energy consumption also. The proposed fog clustering-based load balancing framework for scientific workflows has been shown in Fig. 1. This architecture contains three layers preprocessing module, the fog layer (optimization module), and the cloud layer. These layers are explained as follows:

Table 1 Related work summary

Approach	Objective	Limitation
De Maio et al. [2]	To enhance reliability and to reduce cost and latency	Less attention provided to the utilization of resources
Ijaz et al. [4]	To propose workflow model to reduces energy consumption	The proposed approach considers bi-objective cost function only and proper resource management is not considered
Tellez et al. [10]	Proposed tabu search-based load balancing approach to reduce cost and memory usage	Task migration cost not considered
Xie et al. [11]	Proposed PSO-based scheduling algorithm to reduce cost and makespan	Energy consumption in resources has not given much attention
Serhani et al. [9]	To propose cloud architecture for reducing cost and time to enhance resource utilization	Energy consumption in computing resources has not considered

– **Preprocessing Layer:** The bottom layer is preprocessing layer that takes scientific workflows (LIGO, CyberShake) applications as input to the process in the fog layer. These workflows are large and complex in size that cannot be executed directly. So, the preprocessing module containing its local compiler parsed these large computational processes into smaller tasks, and these tasks are mapped onto fog nodes in the fog layer. Scientific workflows require immediate processing, so instead of passing them to the cloud layer, they need to be processed on the fog layer.
– **Fog optimization layer:** The next layer is the fog layer that is an optimization module that contains different fog clusters containing fog nodes, i.e., access points, routers, base stations, gateways, and switches that have limited storage, processing, and networking services [7]. All fog nodes are combined in clusters containing the specified number of fog nodes so that large and complex workflow computational tasks can be processed immediately. Workflows are large computational datasets that cannot directly run-on fog nodes that are having smaller capacities. So, these workflow tasks are parsed by processing module having its local compiler that smaller tasks that can be executed in available virtual nodes. Due to large and complex tasks, sometimes, a few fog nodes with limited capacities become overloaded, and some remain underloaded. So, load balancing is applied in the fog layer. Load balancer keeps track of all the available and occupied virtual resources in fog nodes. Here, the proposed hybrid load balancing algorithm is applied to distribute workflow tasks among all available resources in equal proportion so that overloading resources can be avoided.

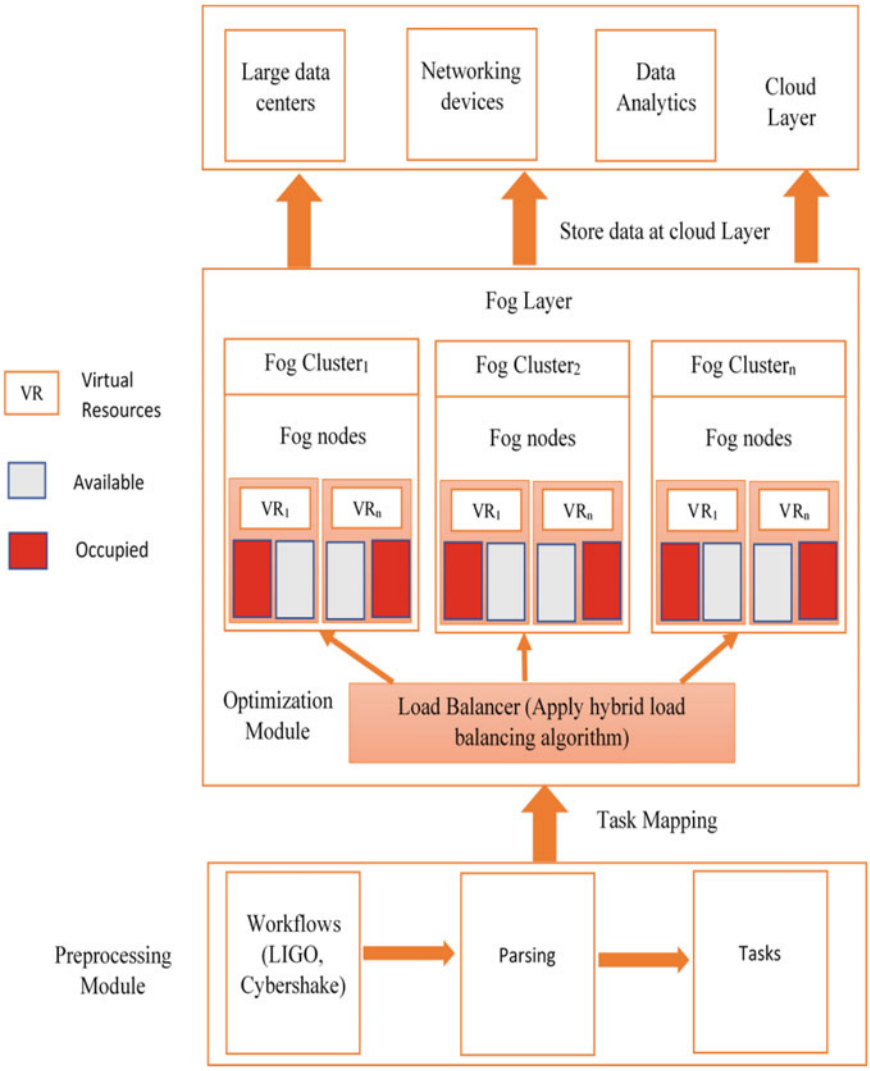


Fig. 1 Fog clustering-based load balancing framework

– **Cloud Layer:** The uppermost layer is the cloud layer that contains large data centers with bulk storage, high-capacity networking devices. The processed data in the fog layer is passed to cloud data centers for further storage. The cloud layer has high-performance servers that have powerful computational and storage capacity. The fog layer cannot store data permanently for future use, transferring it to the cloud layer.

The proposed architecture is mainly designed for load balancing in fog computing environment. Available fog nodes are divided into different clusters that have a local controller that maintains all the nodes in cluster. The cluster controller keeps track of all available nodes and manages task transfer if any nodes becomes overloaded.

4 Proposed Hybrid Load Balancing Algorithm

This section describes the proposed hybrid load balancing algorithm that we have proposed to avoid overloading resources in the fog layer and aims at maximum utilization of resources. Our proposed algorithm is the amalgamation of three different algorithms in which tabu search algorithm is used to find available and occupied resources in fog layer [10]. The other two algorithms, gray wolf optimization GWO) and ant colony optimization, have been used to optimize resources. GWO is a metaheuristic approach that mainly works on the behavior of gray wolves for hunting. The gray wolves are divided into four subgroups $\alpha, \beta, \gamma, \omega$ out of which α is the group leader that gives instructions to others for hunting [8]. ACO is based on the probability method to find different available routes while migrating tasks from overutilized fog resources to available resources. This algorithm works on ant's behavior of searching food [3].

Our proposed Algorithm 1 tends to reduce the overloading of resources in the fog layer so that all the available resources in the fog layer can be efficiently utilized. Different workflows have been selected, and these workflows are parsed into tasks to be executed on fog nodes; after initializing workflow tasks on the fog node, the load balancer analyzes the available and occupied resources. The tasks can be migrated from overloaded nodes to the available nodes. GWO will try to optimize all available resources. If it is optimized, then time delay and cost are analyzed; otherwise, ACO is applied to migrate tasks.

5 Result and Discussion

This section shows the experimental results obtained by evaluating considered scientific workflows. We have considered few scientific workflow applications, and these datasets are executed on iFogSim using eclipse IDE. Table 2 describes the experimental requirements during the evaluation of the proposed framework and algorithm. We have considered 2 to 20 fog nodes for executing workflows. These fog nodes are clustered into four different clusters containing five fog nodes in each. Fog clusters have combined fog nodes that are able to handle large and complex workflow tasks.

After evaluating the workflows in iFogSim, we have analyzed the computational cost of tasks and the time delay during the execution of tasks. The results obtained from iFogSim have been represented in the form of graphs in Figs. 2 and 3. Figure 2 represents cost analysis of computing workflows. The figure has subfigures in which

Algorithm 1: Hybrid load balancing algorithm

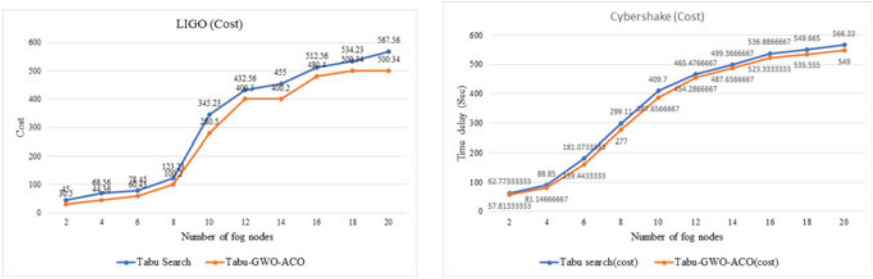
Require: Workflows and fog nodes (n)
Ensure: Optimum time and cost
1. Initialize the number of fog nodes and workflows
2. Parse workflows and extract tasks
3. Map task on fog nodes
4. Apply tabu search to search underutilized fog nodes
5. After finding n selected nodes
6. Apply gray wolf optimization (GWO)
7. Initialization population and define all combinations of X_α, X_β, X_γ to optimize
$X_\alpha \leftarrow G_\alpha$
$X_\beta \leftarrow G_\beta$
$X_\gamma \leftarrow G_\gamma$ // Here, G_α is leader, G_β becomes leader after α and γ are subordinates of β
8. **if** optimized, then calculate time delay and computational cost in fog nodes.
9. **else** apply ant colony optimization(ACO)
10. Update pheromones and obtain optimized fitness value
11. Migrate tasks using a threshold.
12. **if** optimized
13. go to Step 8
14. else go to Step 4
15. **endif**
16. **endif**

Table 2 Experimental requirement

Requirement	Description
Operating system	Windows7 64 bit
Simulator	iFogSim
MIPS	2000
Number of fog nodes	2 to 20
Number of fog clusters	4

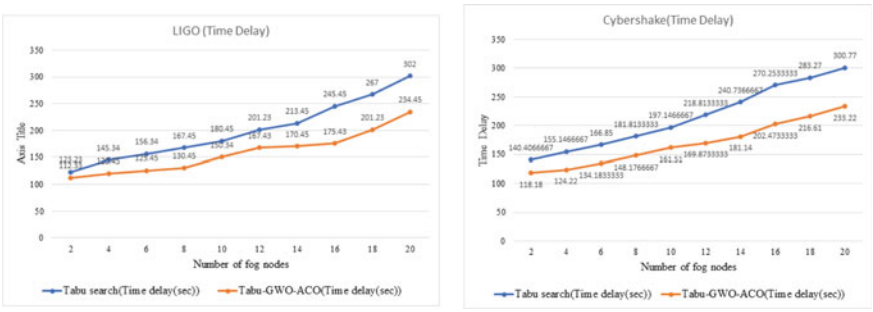
2(a) represents the cost of LIGO workflow and 2(b) represents the cost of CyberShake workflow. It can be seen from the graphs that with the increase in the number of fog nodes, cost also increases. Our proposed hybrid algorithm's results are compared with tabu search, and it can be seen from graphs that our proposed hybrid approach outperforms the existing approach. In the same way, Figs. 3a and b represents time delay calculated during execution of LIGO and CyberShake, respectively.

Proposed hybrid load balancing approach is applied by cluster controller that has its load balancer. The workflow tasks are large computational datasets that are decomposed into smaller tasks and assigned to the fog nodes. The number of fog nodes that are considered for evaluation is from 2–20 nodes. As the nodes increase, time delay and cost also increase. This work tries to reduce cost and time delay as compared to existing approach. The obtained results are compared with tabu search,



(a) LIGO (b) Cybershake

Fig. 2 Cost analysis of different workflows



(a) LIGO (b) Cybershake

Fig. 3 Time delay analysis of different workflows

and it has been obtained that proposed approach has reduced 20 % to 30% time delay and 20 to 25% cost.

6 Conclusion and Future Scope

The main aim of fog computing is to bring computing services near to the end-users by providing mobility and geographical distribution of fog nodes. This paper proposes a fog clustering-based load balancing framework for scientific workflows that aim to maximize resources. Along with this, hybrid load balancing algorithm has been proposed to implement in the fog layer. Our proposed approach has improved time delay to 18 and 22% in LIGO and CyberShake, respectively. However, it has improved the computing cost to 25 and 30% for LIGO and CyberShake, respectively. In the future, we will extend our work to reduce energy consumption in fog computing. However, fog security and load scheduling are open areas for the future researchers.

References

1. Asghari A, Sohrabi MK, Yaghmaee F (2020) A cloud resource management framework for multiple online scientific workflows using cooperative reinforcement learning agents. Comput Netw 179:107340
2. De Maio V, Kimovski D (2020) Multi-objective scheduling of extreme data scientific workflows in fog. Fut Gener Comput Syst 106:171–184
3. Hussein MK, Mousa MH (2020) Efficient task offloading for iot-based applications in fog computing using ant colony optimization. IEEE Access 8:37191–37201
4. Ijaz S, Munir EU, Ahmad SG, Rafique MM, Rana OF (2021) Energy-makespan optimization of workflow scheduling in fog–cloud computing. Computing, pp 1–27
5. Kaur M, Aron R (2020) Energy-aware load balancing in fog cloud computing. Materials Today: Proceedings
6. Kaur M, Aron R (2021) A systematic study of load balancing approaches in the fog computing environment. J Supercomp, 1–46
7. Mahmud R, Ramamohanarao K, Buyya R (2018) Latency-aware application module management for fog computing environments. ACM Transac Inter Technol (TOIT) 19(1):1–21
8. Patel D, Patra MK, Sahoo B (2020) Gwo based task allocation for load balancing in containerized cloud. In: 2020 International Conference on Inventive Computation Technologies (ICICT), pp 655–659. IEEE
9. Serhani MA, El-Kassabi HT, Shuaib K, Navaz AN, Benatallah B, Beheshti A (2020) Self-adapting cloud services orchestration for fulfilling intensive sensory data-driven iot workflows. Future Generation Computer Systems
10. Tellez N, Jimeno M, Salazar A, Nino-Ruiz E (2018) A tabu search method for load balancing in fog computing. Int J Artif Intell 16(2)
11. Xie Y, Zhu Y, Wang Y, Cheng Y, Xu R, Sani AS, Yuan D, Yang Y (2019) A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud-edge environment. Fut Gener Comput Syst 97:361–378
12. Zhou X, Zhang G, Sun J, Zhou J, Wei T, Hu S (2019) Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based heft. Fut Gener Comput Syst 93:278–289

Multi-font Curved Text Scene for Enhanced OCR Readability



P. M. Jena, B. K. Tripathy, and S. K. Parida

Abstract Newspapers contain artistic components like headings, advertisements and different stylish posters, which are printed in such a way that text lines are oriented in differently and contain multi-font curved images. WordArt in Microsoft office is one such widely used tools for text multi-orientation, which can also stretch, skew, bend or modify the shape of the text. This type of text is not horizontal and stylized through multi-font properties which makes it difficult to recognize through conventional optical character recognition (OCR) systems. In this paper, a novel technique is proposed which transforms multi-line and multi-font curved text images into single font horizontal format. This method uses vertical and horizontal projection bar threshold techniques to divide text through line segmentation and word segmentation. Each character is then individually reshaped through its centroid in order to align it with the vertical axis. This approach yields high recognition accuracy for text strings which OCR fails to recognize before alignment.

Keywords Vertical projection · Horizontal projection · Segmentation · Centroid · Translation · Rotation

1 Introduction

Images of typed, printed or handwritten text are converted into their machine-encoded electronic or mechanical form using OCR. The sources may be photocopies or scanned copies of documents and images of scenes superimposed texts on an image.

P. M. Jena
Department of Computer Application, Maharaja Sriram Chandra Bhanja Deo University, Baripada, Odisha 757003, India

B. K. Tripathy (✉)
School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu 632014, India
e-mail: tripathybk@vit.ac.in

S. K. Parida
Department of Maths, SCS Autonomous College, Puri, Odisha 752001, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_19

OCR system has been dedicated to a large volume of research efforts in the field of artificial intelligence and information technology. Several algorithms have been proposed in this direction [1]. There are many OCR systems, which are available in the market. These systems are capable of being applicable on straight line texts and are also specific to fixed script or specific languages. Various character formatting methods are available in the source materials which provide limitations for OCR system feature extraction and recognition. To make the text more attractive and stylish, different orientation of texts has been introduced. This affects in making the OCR system more complex for recognizing these characters. Different algorithms such as vehicle license plate recognition [2], text recognition in video and images [3–5], container identification mark recognition system [6], robotics and image retrieval from the database [7–9] are available. However, stylistics text, such as an advertisement in the front page of newspapers, graphic illustration or signboards, hording boards, is not single oriented and the texts contained being written in curved paths leading to the recognition being erratic. The methods adopted in the process of recognition are bidirectional in nature; first one is being to develop methods which are to recognize the text in their original format in curves, and the second is to transform these curved texts into their straight line form before recognizing. In addition, the texts in the straight line form are needed to be decomposed into either words or characters to be put into the OCR system. In this proposed approach, our main focus is to normalize text and convert curved or oriented text into straight line text. This has the advantage that no specific feature extraction and classification techniques or data set is required which makes our proposed method faster than other existing approaches.

2 Related Works

Xie et al. performed pattern recognition in 1991 and introduced translation, rotation and scale change of pattern and got 97% recognition accuracy from Arabic numerals. Tang et al. in 1991 use ring projection algorithm for multi-oriented English alphabets in the same year. Adam et al. in 2000 define an approach of multi-oriented and multi-scaled character for English stylish text used in engineering drawing in the year 2000. This approach uses Fourier–Mellin transform which is limited to recognize few characters and is time consuming. Hase et al. in 2003 recommend an approach in 2003 for multi-oriented characters. This approach uses parametric Eigen space for English characters which is limited to variation of size, font style and multiple scripts. Pal et al. [10] proposed an approach which is similar to that in [11], but for Indian texts. It is based on character segmentation without any corrections in skewness. In this method, recognition of individual characters are done after segmentation. This method is only for characters in Devanagari and Bangla. An approach for printed alphabet and numerical character reorganization based on different angles of views and extraction of feature which is done in space of two dimensions was proposed in 2007. Roy et al. proposed a method based on the graphical documents which contain differently oriented and differently scaled English text in 2008. This method uses

support vector machine (SVM) for character classification. Chiang et al. in 2011 proposed a method to handle non-homogeneous text. It is a general text recognition technique which applies character sizes and maximum desired string curvature for feature extraction in 2014. Otsu [12] proposes a method for more than one font size and type as well as more than one script text lines within the same document. Here, B-spline curve is used to approximate the pixel data instead of polynomial curve, and this can be used for handwritten text lines in three different languages. Kasar et al. introduced a new technique in 2013. This method consists of text string identification from the page and the steps of alignment that processes each and every text string individually and transformed it in the form of OCR readability. Su et al. [13] proposed technique for an image which contains words at the first step and created a sequential column vectors which convert based on histogram of oriented gradient (HOG). This method uses recurrent neural network (RNN) which further classifies the sequential feature vectors into the corresponding word. Zhang et al. [14] adopted a technique called fully convolutional network (FCN), and this is the model used to train and predict the text regions. Then text line hypotheses are estimated by the combination of feature of character compounds and salient map. Shi et al. introduce text detection method where oriented text is detected through a method called segment linking (SegLink). The main idea of this proposed method is for text decomposing into two local detectable elements, which are called as segments and links features. These features were applied in fully convolutional neural network (FCNN) where elements were detected densely at multiple scales by an end-to-end trained model. Liao et al. in 2020 proposed a module named differentiable binarization (DB), where segmentation network has been used to perform binarization process. A segmentation network adaptively set a limit for thresholds binarization process, which further enhances the text detection and simplifies the post-processing. In Fig. 1, we have captured some English text lines with different curve and font styles.

3 Proposed Approach

In this method, images are generated using Microsoft Office Word software after generating word files, and screen capture software snipping tool has been used for conversion into images format. For quality input to system, original grey-level document images are converted into binary image by applying Otsu [15]. Figure 1 shows six number of multi-font curved text line sample images. Multi-font curved or stylistic text present in document images poses difficulties in segmentation and reorganization.

Here, we have element of $m \times n$, and we use structuring this element in mathematical morphological operation. A page which contains multiple style of text line is our concerned image, where multiple lines are present and each line contains multiple words. In this paper, our innovative attempt is to represent a method, which comprises the following steps (Fig. 2).

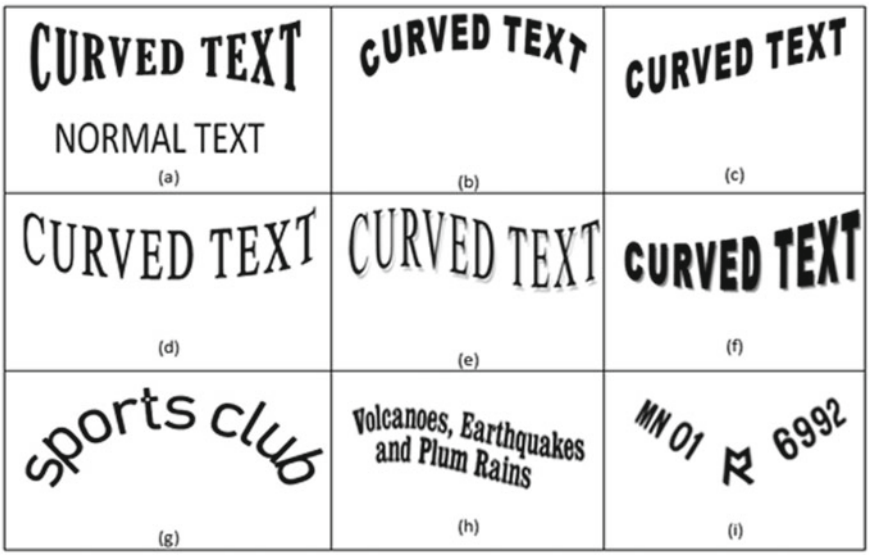


Fig. 1 English text lines with different curve and font styles

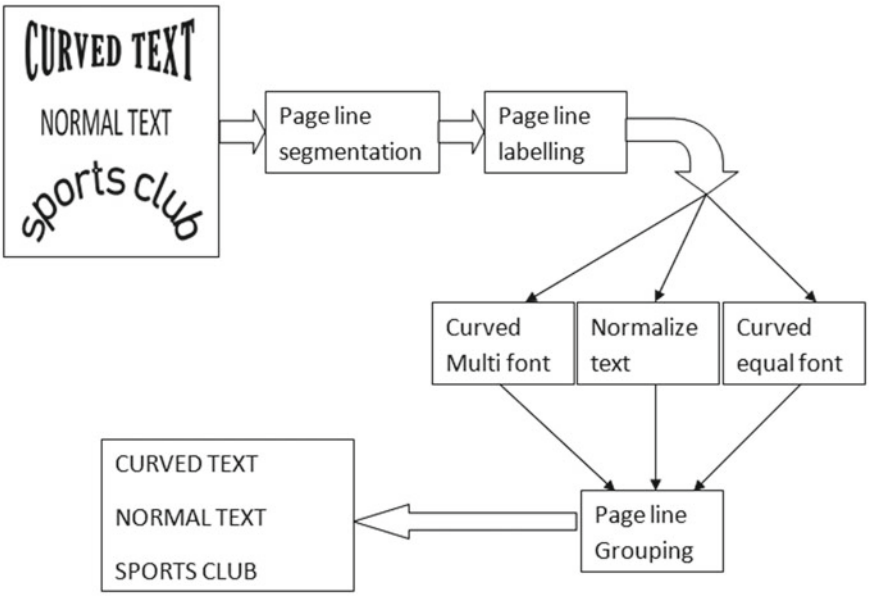


Fig. 2 Block diagram of the proposed text alignment method

3.1 Page Line Processes

Page Line Segmentation. In the context of discussion, Fig. 3 contains multiple text statements, and at the first step, segmentation should be obtained on multiple line statements. Multi-line segmentation has been done through horizontal projection bar as shown in Fig. 3. In this proposed method, horizontal projection bar represents position of lines and gaps between the lines. By applying some thresholding techniques, algorithm's multi-line segmented image found is shown in Fig. 3. Following MATLAB code is used for thresholding horizontal projection bar.

Page Line Labelling. It is the process of categorizing the segmented text line into three categories as curved multi-font, normalize text and curved equal font, where each text line contains multiple characters, and at the initial state, page line is labelling the segmented image into a distance detection technique in which there are processes in the next step. We have used bounding box of each character as shown in blue box in Fig. 4.

As shown in Fig. 4, yellow dots represent the centre of the each character, we have $c1$, $c2$, $c3$, … cn number of centre, brown line represents the distance between centre of the one character to centre of the other character that is $d1$, $d2$, $d3$ … dn, and the orange line $l1$, $l2$, $l3$ … ln represents the vertical length of the bounding box which can be represented as length of the character. Blue line represents actual distance which is the distance between the centre of last character and first character.

$$\text{Average length} = \frac{l1 + l2 + \cdots + \text{ln}}{\text{no of character}}$$

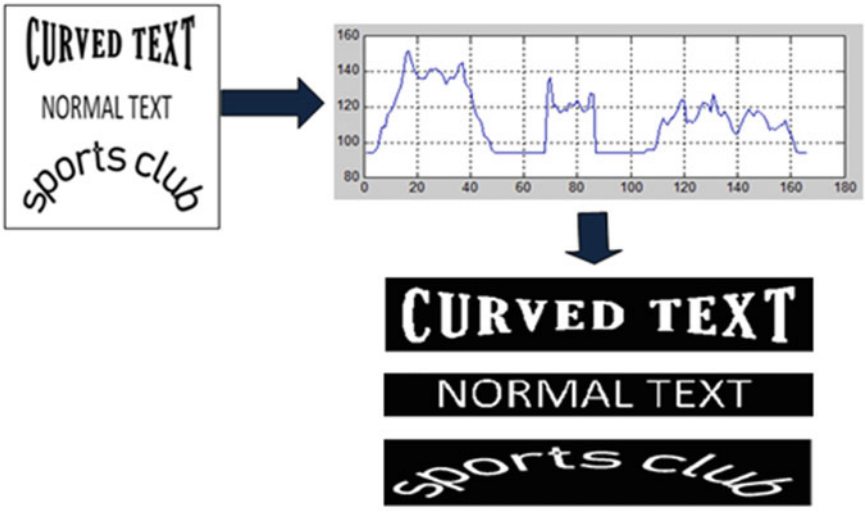


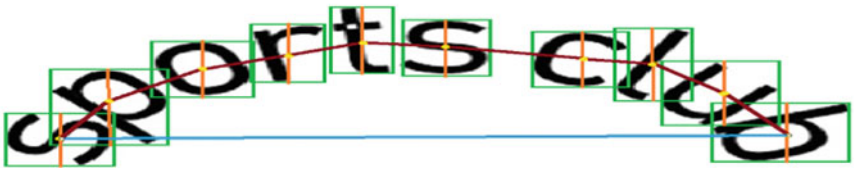
Fig. 3 Page line segmentation using horizontal projection bar



a. Character text line with different fonts



b. Character text line with equal fonts



c. Curved character line with equal fonts

Fig. 4 Page line labelling process

$$\text{Average centre distance} = \frac{d1 + d2 + \cdots + dn}{\text{no of character}}$$

If average length is varying from each character bounding box vertical length and average centre distance is same as actual distance, then we categorized it in multi-font and curved text line as shown in Fig. 4a. If average length is same as the each character bounding box vertical length and average centre distance is same as actual distance, then we categorized it into normalized text line as shown in Fig. 4b.

If average length is varying from each character bounding box vertical length and average centre distance is varying from actual distance, then we categorized it into equal font and curved text line as shown in Fig. 4c.

3.2 Curved Multi-font

After multi-line segmentation, each line may contain multiple words input present in Fig. 4 and is divided into three line segments statements as shown in Fig. 4a–c. Text contained in Fig. 4b can be detected by any OCR system, because this image contains normal text font and has no curved format. Text contained in Fig. 4a is a difficult task to recognize through OCR system. This image contains multi-orient, multi-font, curved text image which is very difficult to recognize. In our proposed method, multi-orient and multi-font curved text are converted into normal text format,

and after conversion, it is going to input it to OCR system where these texts can be easily recognized. These processes are described in below methods.

Word Segmentation. Each word is separated by some distance. Therefore, word segmentation is used to separate each word differently. From Fig. 5, input is given to word segmentation process through the following MATLAB code. Finally, it has been converted into different segmented word as shown in Fig. 5.

Character Segmentation. A single word may contain multiple characters. This proposed method implemented bounding box technique to segment each characters. From Fig. 5, we have segmented it into two words where each word contains multiple characters. Observation from Fig. 5 each character contains curved and multi-font images. For conversion into plan and unique font, the following techniques are applied.

Regionprops. This is a method in MATLAB which returns three elements from the object present in the image: area, bounding box and centroid. The count of pixels in the region is represented by the area value. Bounding box is the smallest rectangular box to which the different objects belong. The rectangular bounding box consists of four corner points, and it can be represented in the form of two-dimensional coordinates in a clockwise direction and centroid vector that specifies the centre of mass of the region. By applying regionprops, each character is divided into segments as shown in Fig. 6.

Resize. It is a method which resizes the font properties. By applying regionpros method from Fig. 5a, words are segmented into number of characters. By finding from Fig. 5a, each character contains multi-font text images which are difficult to detect through OCR system. To make all the font having equal size property, we have applied resize method as shown in Fig. 5a which contains character centroid

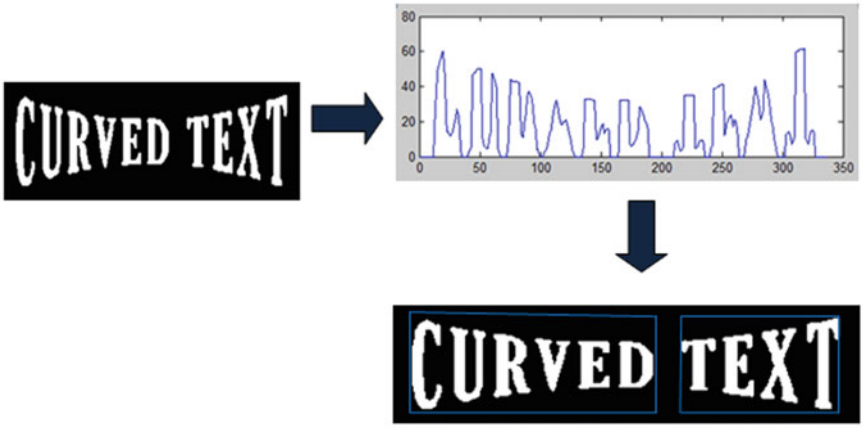


Fig. 5 Multi-word segmentation using vertical projection



Fig. 6 Multi-font curved text converted into plan text

(CC), and each CC value is the position of centre from x and y plane. In the proposed paper, 20 × 40 pixel of each character has been applied which is shown in Fig. 5b. Final Fig. 5b has been given to the OCR system through which it can easily detect the corresponding text (Fig. 7).

Algorithm 1: Multi-font curved Character OCR readability

Input: Multi-font curved word
Output: Normalize font readability to OCR

```
Begin
  Apply region properties into word
  Find area, cetroid and Bounding box of each charecter
  For i=1 to number of Bounding box do
    If Area_of_Character >=15 and Large Bounding box
      Resize the character by applying centroid alignment method
      Call centrod alignment method
    End If
  End For
End
```

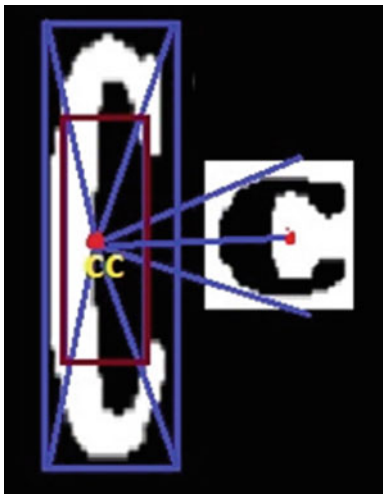


Fig. 7 Region segmentation through centre of image CC and reshape it

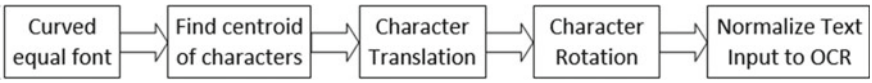


Fig. 8 Block diagram of steps from curved equal font to normalize text



(a) Curved shape equal font size

(b) Find centroid of each character and interconnect every points

Fig. 9 Curved equal font size text line and interconnection of centre points between characters

3.3 Curved Equal Font

Curved characters cannot be detected by our OCR system; in order to convert it into OCR readability format, we have to convert it into normalized text. As we have labelled the type of text line into three categories, curved equal font text line process through the steps is shown in Fig. 8.

The process of conversion from curved equal font into normalized text passes through number of steps. In the first step, centroid of each character is found by using regionprops properties and interconnects the entire centres through a curve line.

In the next step of processing, the fonts are made of equal size (Fig. 9(a)) and the points through which the operations of translation and rotation to be performed are determined (Fig. 9(b)). These operations are carried out until all the characters are in a single straight line (Fig. 10).

In our proposed method, every character having a centroid position is called centre of the character (CC) between two characters, where a centroid connection line is present (CL). This CL line creates an angle between perpendicular lines (PL) which gives the rotation angle $\theta 2$. $\Theta 1$ is the angle between perpendicular line and centroid of character which represent inclined curve between characters. In the next step, we perform translation of curved character (Fig. 11).

In the next step of our method, $\theta 1$ angle is updated to 90 degree because curved character translates its position to PL line, and $\theta 2$ is the clockwise angle of rotation from CC in order to make normalize text. Translation and rotation are executed between pair of two characters until all the characters are not in the normalize text form. Figure 12b shows the normalized text input to the OCR system through which it can be easily detected by OCR system.

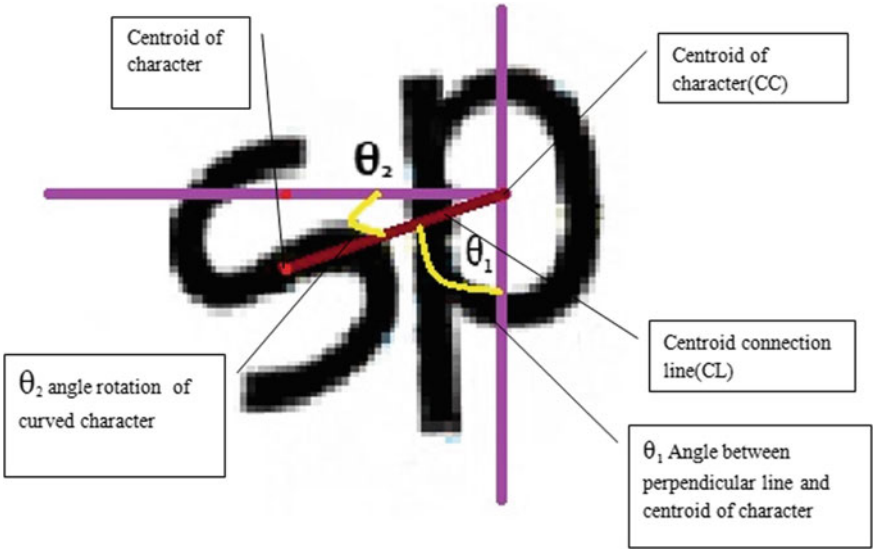
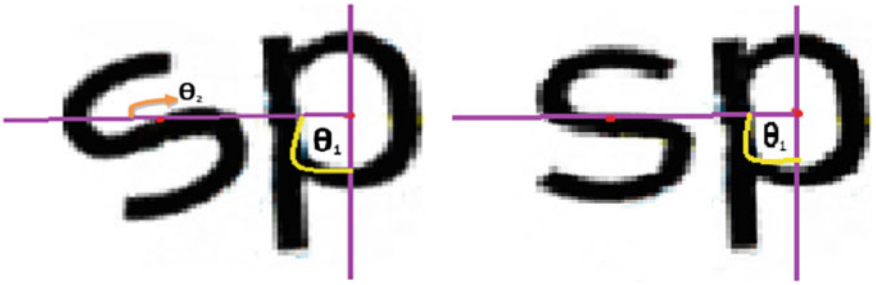


Fig. 10 Curved character pair having CC and CL



(a) Translation of character (b) Rotation of character

Fig. 11 Translation and rotation between pair of characters



(a) Curved shape equal font size (b) Normalized text input to OCR

Fig. 12 Final output from equal font size into normalize text input to OCR

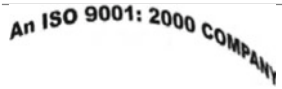
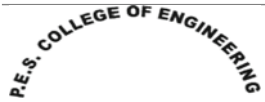
4 Results of Experiment and Analysis

One cannot find any benchmarked database containing images of documents containing text of different font and style. As a consequence for the testing purpose of our proposed method, a collection of 100 text images is arranged, which are curved and different in style from each other. In spite of the smallness of our database, it is taken care that it has texts of different styles, different orientations of characters, different layout styles, like text banded with axis, arranged in circles, spirals, wave type or triangles and non-normal. Also, it contains many combinations of such type of texts. Some of these sample images are shown in Fig. 1. In order to measure the performance of our proposed method, trial version of Nuance OmniPage professional 16 OCR software is used. We use Diego OrlandoBarragan Guerrero, Copyright (c) OCR software to evaluate the performance of our method. The noise-free binary images, which are being used, do not make the segmentation process any easier. Segmentation of these images into OCR is difficult because of its complex text layouts. Clearly, input text images to OCR system must be straight text lines to work reliably and accurately. It is worth noting that segmentation into text lines or its components is not simplified by the use of noise-free binary images because of complex text layouts. It is obvious that documents which have text lines in the straight line form are effective in reliable functionality of the current OCR systems. This proposed method not necessarily only takes care of characters of any specific script. However, this approach is applicable for texts written by using the English language only.

5 Conclusions and Future Work

Alignment of curvilinear text is a necessary step for analysis of multi-font curved text string. We have proposed a novel methodology for the multi-font curved character strings; therefore, offline OCRs can be easily recognized. Our propose method can handle multi-orient text lines having various layouts and various styles as shown in Table 1. While most of the popular online OCR system fails on the images, our propose method of text string alignment ensures as high accuracy rate of printed text alignment and further ensures high reorganization accuracy comparable of unskewed text images. Our propose method does not contain script-specific characteristics text strings needs which are curved multi-font characters, and for other scripts such as Hindi or Chinese language, it can be easily processed. Our future work is to compile the method with different trained classifier for high accuracy and robust detection of curved multi-font text string from real-world scene document images.

Table 1 Comparison study between proposed method and different free online OCR converter

Input text image	https://www.onlineocr.net	https://www.newocr.com	https://smallseotools.com	Proposed method
	CURVED TEXT NORMAL TEXT offtS %	(Upveo TE NORMAL TEXT riS¢PG	CURVED TEXT NORMAL TEXT ortsc(	CURVED TEXT NORMAL TEXT SPORTS CLUB
	r°ast oust eit Mao o°eke 01 leery 0 /op vee	fl (ag, Her go “ey, raoe	oust	Man who waits for roast duck to
	A0 iso 9001: 2000 __ “4 ‘;ceIh ‘xst	001; 2000 & 1so 9 Mn an uy	ISO 9001: 2000	An ISO 9001: 2000 COMPANY
	r°ast oust eit Mao o °eke 01 leery 0 /op vee	GE OF oo” ENG, ©Neee *4%o	EGE OF ENGI	P.E.S. COLLEGE OF ENGINEERING

References

1. Arica N, Yarman-Vural FT (2001) An overview of character recognition focused on off-line handwriting. IEEE Trans Syst Man Cybern Part C Appl Rev 31(2):216–233
2. Cui Y, Huang Q (1997) Character extraction of license plates from video. In: Proceedings of international conference on CVPR, 1997, IEEE, pp 502–507
3. Hua S, Liu W, Zhang HJ (2001) Automatic performance evaluation for video text detection. In: Proceedings of international conference on document analysis and recognition (ICDAR-2001), Seattle, WA, USA, pp 545–550
4. Tripathy BK, Chandramouli PVSSR (2013) Graph based segmentation of digital images, Edited volume. In: Handbook of research on computational intelligence for engineering, science and business. IGI Global Publications, pp 182–199
5. Singhania U, Tripathy BK (2021) Text-based image retrieval using deep learning. In: encyclopedia of information science and technology, fifth edn, p 11. https://doi.org/10.4018/978-1-7998-3479-3.ch007
6. Kumano S, Miyamoto K, Tamagawa M, Ikeda H, Kan K (2004) Development of container identification mark recognition system. Electron Commun Japan (Part II: Electronics) 87(12):38–50
7. Singh N, Tripathy BK (2018) Leukemia cell segmentation from microscopic blood smear image using C-Mode. SocProS 1:225–238
8. Tripathy BK, Sahu SK, Prasad MVN (2013) A support vector machine binary classification and image segmentation of remote sensing data of Chillika Lagoon. Int J Res Inf Technol 1(1):1–17

9. Tripathy BK, Sahu SK, Prasad MVN (2015) PCA classification technique of remote sensing analysis of colour composite image of Chillika Lagoon, Odisha. Int J Adv Res Comput Sci Softw Eng 5(5):513–518
10. Pal U, Tripathy N (2005) Recognition of Indian multi-oriented and curved text. In: Proceedings of the 8th international conference on document analysis and recognition (ICDAR-2005), pp 141–145
11. Tripathy BK, Chandramoulli PVSSR (2016) Computer graphics and applications. Ane Books Private limited, Chennai
12. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
13. Su B, Lu S (2014) Accurate scene text recognition based on recurrent neural network. Asian conference on computer vision (ACCV-2014). Springer, Cham, pp 35–48
14. Zhang Z, Zhang C, Shen W, Yao C, Liu W, Bai X (2016) Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR-2016), pp 4159–4167
15. Tripathy BK, Anuradha J (2015) Soft Computing - Advances and Applications. Cengage Learning India Private Limited, New Delhi

Intelligent Path Planning with Improved BFS (I-BFS) Strategy for Mobile Robot in RFID Equipped Unknown Environment



Prabin Kumar Panigrahi, Sukant Kishoro Bisoy, and Hrudaya Kumar Tripathy

Abstract This paper brings an efficient low complexity path-finding algorithm for autonomous mobile robot in a grid-based environment. Research is carried out by using a hybrid approach which combines the greedy strategy with proposed improving BFS (I-BFS) algorithm. A robot adopts localization strategy based on the environment in which it is employed. The radio frequency identification (RFID)-based localization is quite challenging and is adopted by various authors for localizing robots in indoor environment. The proposed navigation algorithm utilizes RFID technique for positioning the mobile robot which is a substitute for existing vision-based approaches. The added features like both obstacle detection and avoidance are the most promising factor of the proposed methodology. The methodology generated an optimal shortest path in an unknown grid-structured environment through repeated exploration. In this paper, both computationally effective and cost-efficient solution is made for an alternative to several mobile robot navigation algorithms. The navigation algorithm presented here can be used for autonomous vehicular robot employed in different environments like buildings, hospitals, shopping malls, etc., for effective path planning. The simulation result establishes the efficiency of the proposed navigation algorithm.

Keywords RFID · Graph and tree search strategies · Intelligent robot · Localization · Navigation · Path planning

P. K. Panigrahi (✉) · S. K. Bisoy
Department of Computer Science and Engineering, C. V. Raman Global University, Bhubaneswar, Odisha 752054, India

H. K. Tripathy
School of Computer Engineering, Kalinga Institute of Industrial Technology Deemed to be University, Bhubaneswar, Odisha 751024, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_20

1 Introduction

Environmental searching, path patrolling, objects tracking, house cleaning, tour guiding, etc., are different areas in which mobile robots are used extensively [1]. Mobile robot path planning is the key field in today's era of robotic technology. Path planning involves planning a path (or a set of paths) from source to destination [2] in open space or grid-based environment. Robot's movement direction is fixed in four or eight directions in a grid-based environment; however, there is no restriction on the direction of movement in open-space environment. This paper presents an efficient low complexity graph-based navigation algorithm to make use of the robot employed in a grid-based virtual environment. The proposed algorithm uses radio frequency identification (RFID) technology and a hybrid path-planning algorithm (greedy and I-BFS) to guide the robot for navigation.

Abundant robot navigation methodologies like Monte Carlo localization, beacon-based approach, camera-based approach, RFID system, Markov localization method, vision-based localization, a global positioning sensor, multisensory systems, etc., are proposed by different authors. An optimal recursive data processing algorithm [3] named as Kalman filtering (originated in 1960s [4]) is found to be more promising in tracking the robot [5–7]. However, due to restricted Gaussian density, different limitations like tracking failure and nonlinearities are incorporated. Kalman filter suffers from the drawback of globally localization of the robot in the case of localization failure.

Burgard et al. [8] adopt an efficient grid-based localization method named Markov localization in which arbitrary probability densities are represented at fine resolutions. In the proposed approach, memory desires and computational encumber are substantial. In topological Markov localization [9, 10], the topological formation of environment is used to set up the environment. Multimodal and non-Gaussian densities are dealt using Markov localization (grid based). In this localization model, the grid of points is numerically integrated. However, the major drawback in both of these models is that grid size has to be predetermined.

Dellart et al. [11] introduce one of the most promising probabilistic techniques named Monte Carlo localization (particle filters) method where a sampling-based representation is made for localization of a mobile robot. In contrast to Markov localization [8], this approach gives more accurate result in cells having fixed size and is mostly applicable to localize the robot globally. However, this approach results in a loss of diversity, due to the repeated selection of height weight samples. Krumm et al. [12] and Yan et al. [13] discussed the camera-based technology in mobile robot localization in a very efficient manner. In [13], the developed system is fully dependent only on cues. To improve tracking accuracy, different non-visual sensors can be used to collect more data. Dellaert et al. [14] adopt vision-based localization model for positioning the mobile robot. However, the proposed system is not tested on low sensory inputs.

The vision-based approach suffers from the huge burden of complex image processing algorithms, high computation, lack of information, etc. In addition to

this, these systems are less effective in complex environments for tracing out the mobile objects because of the line-of-sight requirement of these systems for establishing connection with objects and localizing them [15]. Additionally, there can be cumulative errors due to numerous uses of sensors. Hence, researchers diverted their center of attention toward RFID technology where the current position can be calculated in an efficient manner with limited use of sensors.

Shirehjini et al. [15] proposed a robot positioning system which uses RFID carpet and different peripherals for sensor data interpretation. Hsu et al. [16] utilized passive RFID (integrated circuit or IC) tags to generate optimal path via A^* search algorithm. Further, Park et al. [17] developed UBIRO (UBIquitous RObot) to help elderly and disabled people. This electric wheel chair uses RFID tags to calculate its orientation. RFID is also used in sensor fusion technique to estimate the robot's position [18]. To improve positioning accuracy, Hähnel et al. [19] utilized RFID scheme in an efficient manner. Gueaieb et al. [20] utilized RFID in the 3D space.

RFID tags are designated as landmarks. RFID reader is attached to the robot access information stored in nearby IC tags and utilizes these data to estimate position and orientation of the robot. Other approaches like smart carpets and smart boards are enhanced for localizing a robot. Yamada et al. [21] propose a sound-based localization in which the objects can be located through sound. However, problem will arise to localize objects which do not produce sound. Ashokraj et al. [22] introduce the multisensory system for localizing the robot. Georgiev et al. [23] adopt the global positioning sensor-based technology for localizing the robot which was very much helpful in urban areas. The proposed method integrates the Kalman filtering with the vision-based system.

Our system guides the robot to navigate in an unknown grid-structured environment by exploring it and achieve the goal with a shortest path. To some extent, the devised path is found to be optimal by reducing fuel consumption, unnecessary exploration and number of turnings. Position estimation is done through the use of IC tags and a RFID reader installed in the mobile robot. This paper addresses several modules like localization, path planning, navigation, obstacle detection and avoidance. The obstacles are seen as partially dynamic (position cannot be changed during exploration). Rest of the paper is organized as follows. A review on mobile robot navigation and related techniques is addressed in Sect. 2. In Sect. 3, we discuss on RFID communication system and then on the proposed navigation algorithm. Experimental evaluation and result analysis are addressed in Sect. 4.

2 Related Work

Recent advances on mobile robotic technology are considered to be more effective in designing efficient navigation algorithms. In this section, we discuss on different navigation algorithms proposed by various authors. Hsu et al. [16] discussed on improved A^* algorithm for mobile robot navigation which concentrates on shortest path generation with the capability of obstacles avoidance. This algorithm used RFID tags

for position estimation and a heuristic search strategy for determination of optimal min-cost path. Dave Ferguson et al. [24] utilize interpolation technique to determine cost-effective paths in grid-based environment. It computed a cheapest path from source to goal as follows:

$$g(s) = \min_{s' \in nbrs(s)} c(s, s') + g(s') \tag{1}$$

Here *nbrs*(*s*): the set of all neighboring nodes of *s*, *c* (*s*, *s*′): edge cost between *s* and *s*′, and *g*(*s*): path cost of node *s*′. Zhou et al. [25] adopt an efficient methodology in which the robot is guided to plan for its path using the known map, and then it is localized based on the iterative closest point (ICP) approach through the help of a laser range scanner. During motion, obstacle avoidance is achieved using the potential field technique. Next, the shortest path is devised through Dijkstra's algorithm. Vacariu et al. [26] use searching algorithms in discrete environments to estimate the robot's path. They adopt *breadth-first search* (BFS) for finding the shortest path. However, their system is applicable with the robots having complete knowledge about the environment which is a major drawback for robots employed in unknown environments.

Different authors utilized soft computing techniques to generate optimal path for mobile robot navigation. Tripathy et al. [27] applied fuzzy system for knowledge-based navigation of a Lego robot. They designed a Lego RCX-wheeled robot with the subsumption architecture which learns obstacle avoidance strategy through online self-adaptation. Further, a command fusion process integrates the outcome from each behavior's rule base, and crisp decision is made using a modified defuzzification scheme. In unknown and unrecognized environment, the *genetic algorithm* (GA) is used for robot movement, obstacle identification, environment learning and trajectory planning [28].

Zhang et al. [29] presented Bayesian filter of variable RF transmission power (BFVP) algorithm which estimates specific location via ultra-high-frequency (UHF)-RFID tags. The proposed strategy is proved to be suitable in real retail environment. In [30], Wu et al. presented the unwrapped phase position (UWPP) model which uses UHF-RFID tags and a RFID antenna for position estimation. Further, Hwang et al. [31] designed a neural network model for localization in RFID system. Panigrahi et al. [32] analyze different localization strategies which adopt probabilistic localization along with simultaneous localization and mapping (SLAM) and RFID for effective path planning. Various authors adopt RFID-based position estimation strategy for efficient path planning. In case of mobile robot systems, RFID systems are employed to localization rather than navigation. This paper presents a novel navigation algorithm where RFID (IC) tags mounted on a floor in equidistance manner are used for storing the current coordinate (*x* and *y*) information. Table 1 provides a mathematical analysis on discrete path-planning strategy developed by various authors.

The research in [16, 24–26] focuses on path planning in a known environment. However, in indoor environment, the environment is normally dynamic, because

Table 1 Mathematical analysis on position estimation and orientation in RFID system

Reference	Approach	Mathematical analysis
Shirehjini et al. [15]	RFID-based position estimation	Position (P_x, P_y) is computed as follows: $P_x = (m-1) * S_{\text{width}} + (x-1) * T_{\text{width}} + (T_{\text{width}}/2)$ $P_y = (n-1) * S_{\text{height}} + (y-1) * T_{\text{height}} + (T_{\text{height}}/2)$ where x, y, m and $n > 0$. T_{width}: tag width, T_{height}: tag height, S_{width}: carpet width, S_{height}: carpet height
Hsu et al. [16]	Position estimation from nearby IC tags	$X_{\text{pos}} = (x_1 + x_2 + \cdots + x_n)/n$, $Y_{\text{pos}} = (y_1 + y_2 + \cdots + y_n)/n$ where n: number of detected tags and (x_1, y_1), (x_2, y_2), … (x_n, y_n) are the detected tags
Park et al. [17]	Position estimation in polar coordinate	In xy Cartesian coordinate system, the polar coordinates (r, θ) are defined as follows: $x = r\cos\theta$, $y = r\sin\theta$, $r = \sqrt{x^2 + y^2}$, $\theta = \tan - 1(y/x)$, $x_{\text{current}} = r\cos\theta$, $y_{\text{current}} = r\sin\theta$ r: radial distance from origin, θ: angle (counterclockwise) from the positive x-axis
Choi et al. [18]	RFID sensor fusion-based localization	The state of the robot at time t is estimated as follows: $\hat{p}(t) = \left[\hat{x}(t)\hat{y}(t)\right]^{\text{T}} =$ $\left[x'(t) + a(t)y'(t) + b(t)\right]^{\text{T}}$ $a(t) = \left\lvert x'(t) - \hat{x}(t-1)\right\rvert >$ $0.5\left\lvert \text{dis}_{\text{tag}}\right\rvert$ $b(t) = \left\lvert y'(t) - \hat{y}(t-1)\right\rvert >$ $0.5\left\lvert \text{dis}_{\text{tag}}\right\rvert$ Here the errors $a(t)$ and $b(t)$ uncertainty in the measurement range and dis_{tag} are the gap between the IC tags
Hähnel et al. [19]	Probabilistic position estimation approach	$p(x\lvert z_{1:t}, r_{1:t})$ be the posterior distribution over potential distribution of a RFID tag $p(y\lvert l) = \sum_x p(y\lvert x, l)\, p(x\lvert z_{1:t}, r_{1:t})$ In accordance with the law of total probability, $p(y\lvert x, 1)$ corresponds to the sensor model, and l is the robot's position

(continued)

Table 1 (continued)

Reference	Approach	Mathematical analysis
Gueaieb et al. [20]	Intelligent orientation technique	The phase of the baseband signal (φ) can be calculated as follows: $\varphi = \tan^{-1}(I/Q)$, where I: in-phase and Q: quadrature components of the derived signal. The phase difference can be determined by $\Delta\varphi = \varphi_1 - \varphi_2$, where φ_1 and φ_2 are the phase angles. The updated direction can be computed by the following update rule: $\theta_{new} = \theta_{old} + \Delta\theta$, where $\Delta\theta$: tune-up amount, θ_{old}: old direction and θ_{new}: updated direction.

the positions of furniture and other objects may change. This requires development of an efficient methodology to be adaptable by the robot employed in dynamically changing environment. Our proposed approach devices shortest path in dynamical environment. The robot is only supplied with the boundary size, and it automatically builds the environment map, plans path and updates the map during navigation. It follows a greedy strategy there by generating best path after each detection of obstacle.

3 Proposed Approach

In this study, we proposed a greedy algorithm to generate optimal shortest path in an unknown grid-structured environment. The proposed system utilizes passive RFID tags to devise the path for the robot. The robot is only supplied with the dimension of the boundary and information about the goal (in terms of coordinate). Next it builds up a virtual map, plans shortest path from source to goal using improved BFS algorithm and updates its map during obstacle detection. The robot is applicable in indoor environment within a certain dimension.

The architecture of the navigation system consists of a RFID communication module and a software system for data communication making control decisions. The proposed technique focuses on RFID tags mounted on a 2D plane for estimating the current position. Each of the RFID tag (mounted on the floor) only contains its position in the *xy* plane. With the supplied parameters like source coordinate, destination coordinate and grid dimension, the robot generates a virtual map. It constructs a graph taken into account this map over which the shortest path algorithm runs. Prior to navigation, the robot comes with a shortest path from source to destination. The robot validates the correctness of this path through the information provided by the

RFID reader attached to it. It senses nearby tags coming within its sensing range. For instance, suppose the robot is supplied with the sequence ((1, 1), (1, 2), (1, 3), (1, 4)) and the RFID reader detects the tags: (2, 2) (1, 3) and (2, 3), then the robot only navigates to cell (1, 3) thereby discarding the other tags (2, 2) and (2, 3). This is in contrast to the position estimation technique proposed by Hsu et al. [16] where the location of the robot is estimated by taking the average of all detected tags within the sensing range. In the following, we discuss on RFID communication system and the proposed navigation algorithm.

3.1 RFID Communication Unit

For efficient position estimation, low-cost passive RFID tags are distributed in a grid-like manner (Fig. 1). During navigation, the RFID reader throws electromagnetic wave which detects nearby IC tags and reads their coordinate value. Coordinate values of all tags are checked with the next cell in the pre-supplied path sequence. If a match occurs, then the robot navigates to this cell; otherwise, it is treated as an obstacle occupied coordinate, and an alternative path is searched.

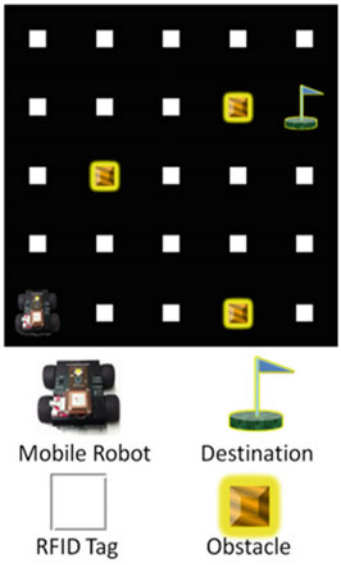


Fig. 1 Virtual world in simulated environment

3.2 Navigation Algorithm

A greedy-based navigation algorithm is proposed to maintain coordination with the RFID communication unit and to pass updated path details to the robot for making intelligent-based decision. The flowchart (Fig. 2) illustrates the navigation process, and the detail description of the algorithm is given below.

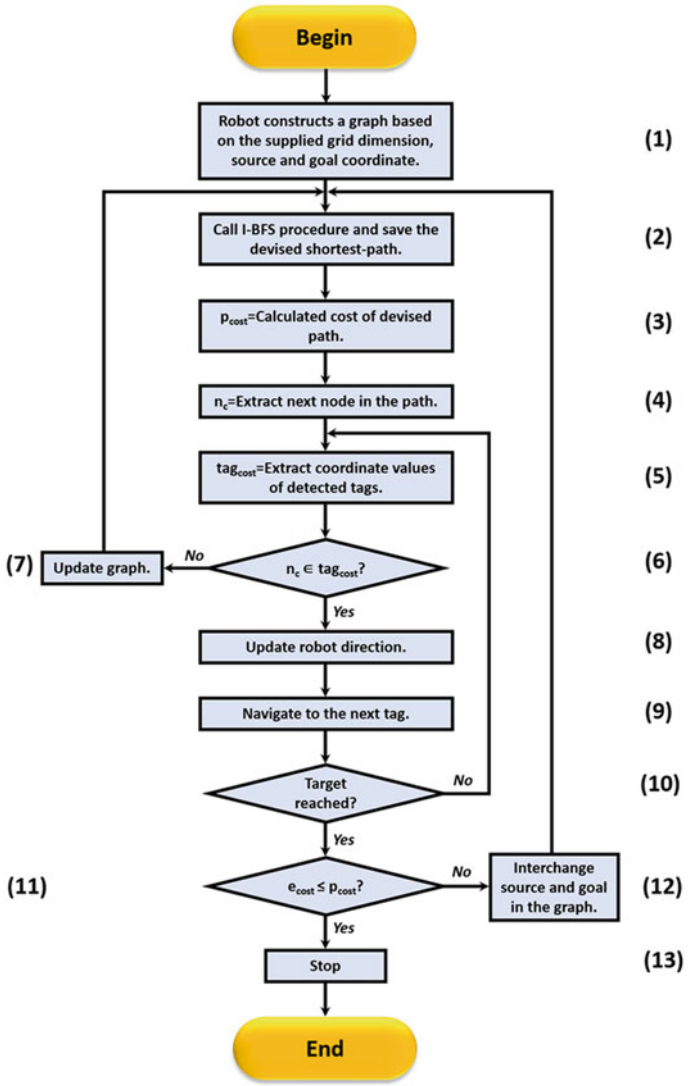


Fig. 2 Flowchart of the proposed navigation algorithm

Step 1: The robot constructs a 2D graph of the virtual environment based on supplied parameters (source, destination and grid dimension).
Step 2: I-BFS procedure is called on the graph, and devised shortest path is returned.
Step 3: Path cost (1 for each edge) is assigned to p_{cost}.
Step 4: Next n_c is assigned to the intended tag of current navigation phase.
Step 5: With the above information (Step 4), the robot throws electromagnetic wave, detects its nearby IC tags and records their contents.
Step 6 and 7: If none of the IC tag value matches fetched coordinate value than the existing graph, it is updated by removing the obstacle node (as target tag is in obstacle region) and control is passed back to Step 2.
Step 8 and 9: Robot's direction is updated, and it navigates to the next tag in the path.
Step 10: The robot validates whether it has got to the destination tag or not. This can be done by checking the last node in the path with detected IC tags which are used in the simulation described in Sect. 4. Alternatively, the robot can match its existing goal information (basically image) with the sensed image.
Step 11, 12 and 13: If e_{cost} (explored path cost) is less than or equal to p_{cost}, the robot navigates deciding the explored path as the shortest one, otherwise control passes to Step 3 after updating the graph, i.e., interchanging source and goal in the existing graph.

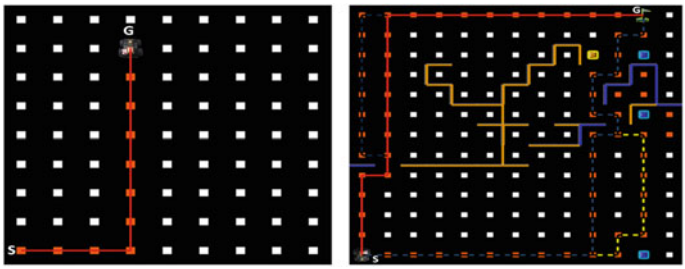
The shortest path procedure named I-BFS is an improved version of the existing BFS. I-BFS iterates same as BFS; however, the only difference lies in the adjacent node determination phase. In BFS, this phase requires n iterations to trace out the entry of the node u in the adjacency list which is in contrast to the technique used in I-BFS which uses only single calculation. Each node will be allocated a coordinate value, i.e., its position in the grid. The position of a node in the list can be calculated as follows:

$$u_i = ((r-1) \times \text{col}) + (c-1) \tag{2}$$

u_i Index of the node (say u) to be explored, r: x position of the node u. c: y position of the node u. col: Total number of columns in the grid.

4 Result Analysis

To evaluate the performance of the proposed system, we conducted two experiments: fully obstacle-free environment and complex environment. The simulation is performed using MATLAB and Java. Mobile robot is employed in a 2D grid-based virtual environment. In all experiments, direction of the robot is monitored by the mobile robot navigation algorithm.



(a) (b)

Fig. 3 Shortest path: **a** obstacle-free environment **b** environment filled with obstacles

4.1 Experiment I: Fully Obstacle-Free Environment

The first experiment is carried in an obstacle-free environment where the robot navigates in the supplied shortest path through sensing its surroundings. However, as the environment is free of obstacles, there is no need to alter the existing path, and hence, target is reached with shortest path (shown as red line in Fig. 3a) in the first iteration of the outer loop itself (Fig. 2). It can be observed from Fig. 3a that in an $m \times n$ grid-structured obstacle-free environment, maximum of $m + n - 1$ cells are required to be visited having at most two turnings, and hence, the devised path is an optimal shortest path.

4.2 Experiment II: Complex Environment

The second experiment is carried out in an environment filled with obstacles where the robot detects and avoids them through updating the existing graph each time in the navigation process. Figure 3b demonstrates a case where the robot reaches its target with a very limited number of explorations (three in this case). The navigation algorithm reduces the number of turns in comparison with the path-planning algorithm presented in software-based BFS [26] approach (Fig. 4a). The experimental analysis on numerous grid dimensions shows I-BFS runs slight speed compared to BFS (Fig. 4b). The deviation of the running is in terms of milliseconds due to the direct determination of adjacent nodes (Eq. 2). Graph construction is in the initial stage, i.e., prior to navigation, and it gets updated only during obstacle detection which decreases computation time. Our navigation algorithm removes all self-loops from the explored path, thus provides an efficient shortest path to the robot.

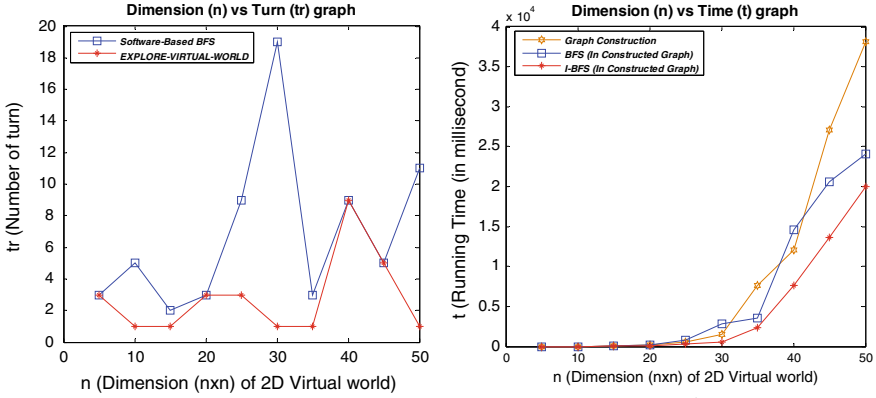


Fig. 4 Analysis on **a** turn rate and **b** running time

5 Conclusion

This paper presents an efficient path-planning algorithm with obstacle avoidance using a non-vision-based localization scheme, i.e., RFID technology. Through an intellectual-based decision-making process, the proposed methodology guides the robot to navigate in an effectual manner in unknown environment. The primary novelty of our system is to determine nearly optimal shortest path in unknown environment. This is achieved by updating the virtual map during obstacle detection. Each time the robot reaches the target, it compares previous path with current path, which guarantee the determination of final shortest path. The greedy strategy helps the robot to search best path at each iteration and which results in optimal path. This reduces fuel consumption, unnecessary exploration, unnecessary turning, etc. The proposed system can be implemented in autonomous wheel chair to help physically disabled people to move without external intervention. It can also be applied in hospitals and department campus in universities for transportation. In the future research, this paper can be extended to avoid dynamic obstacles during navigation. The capabilities of the algorithm will be extended to be able to use landmark-based navigation.

Acknowledgements Mr. Prabin Kumar Panigrahi expresses his profound gratitude to his parents for their support in this research.

References

1. Panigrahi PK, Tripathy HK (2015) Analysis on intelligent based navigation and path finding of autonomous mobile robot. In: Mandal J, Satapathy S, Kumar Sanyal M, Sarkar P, Mukhopadhyay A (eds) Information systems design and intelligent applications, 2015. AISC. Springer, New Delhi, pp 219–232. https://doi.org/10.1007/978-81-322-2250-7_22
2. Panigrahi PK, Tripathy HK (2015) Low complexicity graph based navigation andpath finding of mobile robot using BFS. In: Proceedings of the 2nd international conference on perception and machine intelligence (PerMIn'15), 2015. ACM, NewYork, NY, USA, pp 189–195. https://doi.org/10.1145/2708463.2709068
3. Maybeck PS (1979) Stochastic models, estimation control. Academic Press
4. Kalman RE (1960) A new approach to linear filtering and prediction problems. ASME J Basic Eng 82(1):35–45. https://doi.org/10.1115/1.3662552
5. Leonard JJ, Durrant-Whyte HF (1992) Directed sonar sensing for mobile robot navigation. Springer Science+Business Media, LLC, United Kingdom. https://doi.org/10.1007/978-1-4615-3652-9
6. Schiele B, Crowley JL (1994) A comparison of position estimation techniques using occupancy grids. In: Proceedings of the 1994 IEEE international conference on robotics and automation, 1994. San Diego, CA, USA, pp 1628–1634. https://doi.org/10.1109/ROBOT.1994.351357
7. Gutmann J-S, Schlegel C (1996) AMOS: comparison of scan matching approaches for self-localization in indoor environments. In: Proceedings of the first euromicro workshop on advanced mobile robots (EUROBOT'96), 1996. Kaiserslautern, Germany, pp 61–67. https://doi.org/10.1109/EURBOT.1996.551882
8. Burgard W, Fox D, Hennig D, Schmidt T (1995) Estimating the absolute position of a mobile robot using position probability grids. In: Proceedings of the thirteenth national conference on artificial intelligence (AAAI'96), 1996. AAAI Press, pp 896–901
9. Simmons R, Koenig S (1995) Probabilistic robot navigation in partially observable environments. In: Proceedings of the 14th international joint conference on artificial intelligence (IJCAI'95), 1995. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 1080–1087
10. Cassandra AR, Kaelbling LP, Kurien JA (1996) Acting under uncertainty: discrete Bayesian models for mobile-robot navigation. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems. IROS'96, 1996, pp 963–972. https://doi.org/10.1109/IROS.1996.571080
11. Dellaert F, Fox D, Burgard W, Thrun S (1999) Monte Carlo localization for mobile robots. In: Proceedings 1999 IEEE international conference on robotics and automation (Cat. No.99CH36288C), 1999. Detroit, MI, USA, pp 1322–1328. https://doi.org/10.1109/ROBOT.1999.772544
12. Krumm J, Harris S, Meyers B, Brumitt B, Hale M, Shafer S (2000) Multi-camera multi-person tracking for easy living. In: Proceedings third IEEE international workshop on visual surveillance, 2000, pp 3–10. https://doi.org/10.1109/VS.2000.856852
13. Yan W, Weber C, Wermter S (2011) A hybrid probabilistic neural model for person tracking based on a ceiling-mounted camera. J Ambient Intell Smart Environ 3:237–252. https://doi.org/10.3233/AIS-2011-0111
14. Dellaert F, Burgard W, Fox D, Thrun S (1999) Using the CONDENSATION algorithm for robust, vision-based mobile robot localization. In: Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149), 1999. Fort Collins, CO, USA, pp 588–594. https://doi.org/10.1109/CVPR.1999.784976
15. Shirehjini AAN, Yassine A, Shirmohammadi S (2012) An RFID-based position and orientation measurement system for mobile objects in intelligent environments. IEEE Trans Instrum Meas 61(6):1664–1675. https://doi.org/10.1109/TIM.2011.2181912
16. Hsu Y-F, Huang C-H, Huang W-R, Chu W-C (2013) An improved minimumcost pathfinding algorithm for mobile robot navigation. J Adv Comput Netw 1:189–193. https://doi.org/10.7763/JACN.2013.V1.38

17. Park S, Hashimoto S (2009) Indoor localization for autonomous mobile robot based on passive RFID. In: 2008 IEEE international conference on robotics and biomimetics, 2009. IEEE, Bangkok, pp 1856–1861. https://doi.org/10.1109/ROBIO.2009.4913284
18. Choi B-S, Lee J-W, Lee J-J, Park K-T (2011) A hierarchical algorithm for indoor mobile robot localization using RFID sensor fusion. IEEE Trans Ind Electron 58(6):2226–2235. https://doi.org/10.1109/TIE.2011.2109330
19. Hähnel D, Burgard W, Fox D, Fishkin K, Philipose M (2004) Mapping and localization with RFID technology. In: IEEE international conference on robotics and automation. Proceedings. ICRA'04, vol 1. New Orleans, LA, USA, pp 1015–1020. https://doi.org/10.1109/ROBOT.2004.1307283
20. Gueaieb W, Miah MS (2008) An intelligent mobile robot navigation technique using RFID technology. IEEE Trans Instrum Measur 57(9):1908–1917. https://doi.org/10.1109/TIM.2008.919902
21. Yamada T, Nakamura S, Shikano K (1996) Robust speech recognition with speaker localization by a microphone array. In: Proceeding of fourth international conference on spoken language processing ICSLP'96. Philadelphia, PA, USA, pp 1317–1320. https://doi.org/10.1109/ICSLP.1996.607855
22. Ashokaraj IAR, Silson PMG, Tsourdos A, White BA (2009) Robust sensor-based navigation for mobile robots. IEEE Trans Instrum Measur 58(3):551–556. https://doi.org/10.1109/TIM.2008.2005266
23. Georgiev A, Allen PK (2004) Localization methods for a mobile robot in urban environments. IEEE Trans Rob 20(5):851–864. https://doi.org/10.1109/TRO.2004.829506
24. Ferguson D, Stentz A (2006) Using interpolation to improve path planning: the field D* algorithm. J Field Rob 23(2):79–101. https://doi.org/10.1002/rob.20109
25. Zhou J , Lin H (2011) A self-localization and path planning technique for mobile robot navigation. In: Proceedings of the 8th world congress on intelligent control and automation, 2011, pp 694–699. https://doi.org/10.1109/WCICA.2011.5970604
26. Vacariu L, Roman F, Timar M, Stanciu T, Banabic R, Cret O (2007) Mobile robot path-planning implementation in software and hardware. In Proceedings of the 6th WSEAS international conference on signal processing, robotics and automation (ISPRA'07), 2007. WSEAS. Stevens Point, Wisconsin, USA, pp 140–145. https://doi.org/10.5555/1355681.1355705
27. Tripathy HK, Tripathy BK, Das PK (2008) A prospective fuzzy logic approach to knowledge-based navigation of mobile LEGO robot. J Convergence Inf Technol 3(1):64–70
28. Yun SC, Parasuraman S, Ganapathy V (2011) Dynamic path planning algorithm in mobile robot navigation. In: IEEE symposium on industrial electronics and applications. IEEE, pp 364–369. https://doi.org/10.1109/ISIEA.2011.6108732
29. Zhang J, Lyu Y, Patton J, Periaswamy SCG, Roppel T (2018) BFVP: a probabilistic UHF RFID tag localization algorithm using Bayesian filter and a variable power RFID Model. IEEE Trans Ind Electron 65(10):8250–8259. https://doi.org/10.1109/TIE.2018.2803720
30. Wu H, Tao B, Gong Z, Yin Z, Ding H (2019) A fast UHF RFID localization method using unwrapped phase-position model. IEEE Trans Autom Sci Eng 16:1698–1707. https://doi.org/10.1109/TASE.2019.2895104
31. Hwang C, Chen T, Hung JY (2017) Neural-network-based mobile RFID localization system. In: IECON 2017—43rd annual conference of the IEEE industrial electronics society. Beijing, pp 6051–6056. https://doi.org/10.1109/IECON.2017.8217050
32. Panigrahi PK, Bisoy SK (2021) Localization strategies for autonomous mobile robots: a review. J King Saud Univ Comput Inf Sci (Elsevier). https://doi.org/10.1016/j.jksuci.2021.02.015

Application of Deep Learning in Honeypot Network for Cloud Intrusion Detection



S. B. Goyal, Pradeep Bedi, Shailesh Kumar, Jugnesh Kumar, and Nasrin Rabiei Karahroudi

Abstract Network protection becomes the key concern of all organizations in defending them from cyber threats of all kinds. The analysis of traffic in the network is an important activity for this reason. Network traffic analysis is achieved via the implementation of various intrusion detection systems. A honeypot is one of the most critical instruments to detect network intrusions. Honeypots communicate with the attacker and gather data that can be analyzed to collect information about the attacks and network attackers. Honeypots are security tools by which the attacker targets and generates attack data logs. Honeypots provide small quantities of relevant data so that security vigilance can easily understand, and future research can be carried out. To resolve the issues of new attacks in the network, this paper is focused to focus on the integration of honeypot to identify and treat suspicious network traffic flow. This paper gives a brief overview of the implementation of honeypot in the cloud. In this work, deep learning trained IDS is proposed on the honeypot server for network traffic analysis.

Keywords Cybersecurity · Intrusion detection system (IDS) · Honeypot · Security threats · Machine learning · Deep learning · Cloud computing

S. B. Goyal (✉) · N. R. Karahroudi
City University, 46100 Petaling Jaya, Malaysia

P. Bedi
Galgotias University, Greater Noida, India

S. Kumar
BlueCrest College, Freetown, Sierra Leone

J. Kumar
St. Andrews Institute of Technology & Management, Gurgaon, Haryana, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_21

1 Introduction

Cloud computing is an evolving paradigm of programming that provides service over the internet. The main aim to develop cloud computing was to provide all technological solutions on a single platform [1]. All information and solutions are provided over the cloud and can be accessed through the Internet. Hence, this complete package of services over the Internet is termed "cloud computing." Cloud computing provides the facilities such that data of users can be stored on remote servers permanently and can be accessed by the client from anywhere on any device such as laptop, mobile, and tablet. It facilitates the customers to access structured services in an adaptive, fast, and scalable manner that only takes into consideration usage. Three perspectives are used to approach this concept. Initially, it should be noted that the advancement of several innovations consolidated over many years has culminated in cloud computing [2]. Secondly, the technological trend is so common and generally embraced. Finally, the same process methodologies also exist in the world of development-oriented cloud computing.

1.1 Security Issue in Cloud Computing

The topic of information technology protection is one of the most significant. The key concern is to preserve organizational or user data [3]. If cloud data is not reliable for the enterprise, the change from old technology to cloud technology is not useful [4, 5].

Misuse of Cloud: Cloud computing may be used by such malicious people for illicit and malicious activities. Cloud use for unlawful purposes is referred to as cloud misuse. Any malicious users may use the cloud services to provide malicious services to a number of users. Some malicious users typically create cloud ads to attract users and to ask for personal data. This personal data is then used on user addresses for illegal tasks or for sending spam. This sort of cloud advertising is a kind of cloud misuse.

Weak and Insecure Application: Cloud data and resources are accessed via the application programmable interface. Weakly built API errors appear to expose unnecessary users to services and data. A flaw in the Apache Web Server, for example, can lead to a user having complete server access. Data may often be exchanged between different users of shared cloud services due to API malfunctions and sometimes due to user privacy settings being rewritten. Most of the time, API architecture without any security measures leads to poor APIs. But often the deliberately weak API is intended for malicious operations.

Insider Theft: While consumers can rely on CSP, they cannot rely on their employees because not everyone is successful. Some employees of cloud service providers may track and steal data of customers for malicious activities. Even some CSP companies will send other companies our data to earn money [6].

Security Issues Due to Virtualization: IaaS service users can build multiple virtual machines on the same server because of the virtualization architecture. Attacks on such machines may hamper customers' data as users access the internal machines for their applications. Secondly, users should configure their virtual machine on the same server, and if their virtual machine is mounted on the same server, several attacks can be carried out and details about keystroke timings, CPU cache utilization, and network traffic rates can be easily identified.

Combined Services: Some cloud services are focused on other services and are relying on other services. User information is exchanged between all applications used by the service that the user uses. This allows the customer to share personal information with a service provider he was not allowed to share with. And without understanding the customers, back-end services may abuse their details.

Data Damage and Loss: The most important aspect of cloud services is data, and its protection must therefore be treated with great concern. The harm and loss of data can be achieved in two ways: firstly for natural causes and secondly because of problems caused by humans. Data loss due to natural causes such as servers will be destroyed due to earthquakes in the server site, fire in the server space, servers' physical damage, hardware issues so that information cannot be recovered. Harm to data from malicious problems is attacked by a malicious user, and all data can be lost or overwritten even though all data is transferred. A malicious user may modify the access rights to data to prevent original users from accessing his data equivalent to damage to his data. Any malicious users can modify services so that all data can be saved in cloud storage by the customer. A standard API inspection of cloud services is the solution to such problems and set up a protected access control list to prevent unauthorized users from accessing it easily.

Eavesdropping: According to the dictionary, the term eavesdropping meaning is to listen to other personal talks without knowing them. However, in technical terms, computer science means to listen to someone on the information without the data owner's permission. Man-in-the-middle attack (MIMA) is an example of eavesdropping. This lets a malicious person connect individually to a cloud user which looks like an original cloud-user link, thereby redirecting all your conversations and data to their storage devices. It is the same as deleting Web sites where users get the same site as their original one to enter their data and data into malicious user storage devices.

Intrusion: Intruders are acts that seek to overcome the information systems' security frameworks unclearly. This is a set of steps that jeopardize information integrity, availability, and/or privacy. Privacy means that no information can be revealed to non-accessible third parties. Integrity guarantees during the transmission that the message was not altered. When a user sends a message to another user and before the message reaches the target user, an unauthorized user changes the content of the message. The loss of reputation is recognized and triggered by changes.

The availability feature means that approved users should always be supplied with resources. Attacks such as disturbances cause a lack of resources available. The intruders are commonly triggered by a third-party applicant (middleware) vulnera-

bility that is accessing the device through the Internet or the local network or operating system on the infected computer or attempted to prevent those approved users from using the attacker. Security abuses and rights of earnings and security.

2 Intrusion Detection

IDS detects malicious behavior in computer systems and, after the attack is complete, conducts forensics. To detect intrusions and attacks that have not been blocked by preventative techniques, search network resources (firewall, router packet filtering, proxy server) [7]. Intrusion is an effort to undermine a system's privacy, credibility, or availability. It is possible to view intrusion detection devices as a rough comparison to real intruder detectors. IDS (as shown in Fig. 1) based on misuse are configured to identify breaches of predefined security policies. But with the emergence of potential unhealthy habits that cannot be predetermined, things become complicated (Fig. 2).

A developer in an enterprise, for example, that transfers large quantities of data within a short time. This may be a possible issue with data leakage, but the access policy may not detect it because it is allowed to move information. This is why

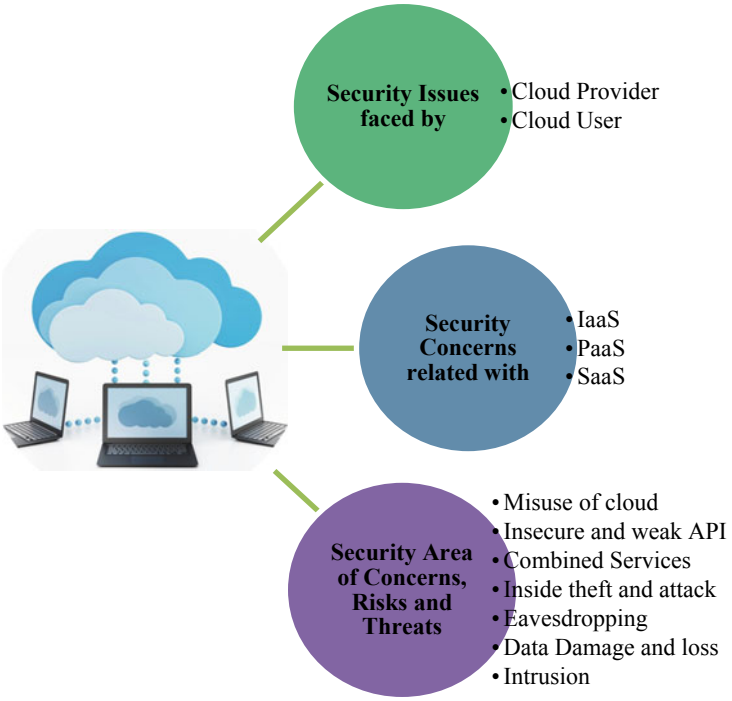


Fig. 1 Security issues in cloud

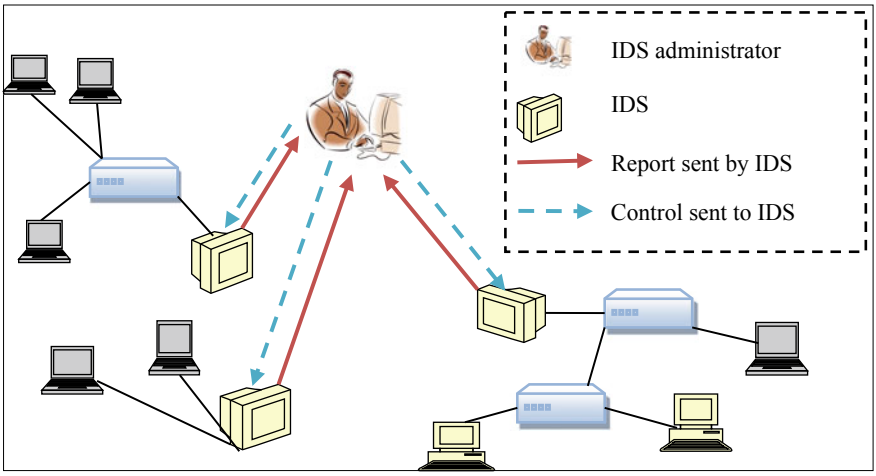


Fig. 2 Intrusion detection system

statistical anomalies are observed in which a user or device profile is generated and profile discrepancies are identified. Although both types of systems are beneficial, a combination of both will minimize the individual drawbacks, but cannot remove them.

The source of audit data is a significant factor determining the type of IDS-inherited implementation. Both primary sources are protocols used by host-based IDSs and data packets in a network used by network IDSs. Kernel logs, program logs, or device-related logs are possible for host protocols.

There are several problems with IDS as discussed below:

- The list of machine-specific detection parameters is extremely long in heterogeneous operating systems for any system.
- The increase in important nodes improves efficiency within the network.
- Degradation of host device output due to additional safety behavior such as enrollment.
- Difficulty in detecting network-level attacks.
- Host to provide a full host-based IDS with inadequate computing capacity.

Intrusion detection system or IDS is network surveillance systems which have become almost imperative in recent years due to the relentless rise in the number and risk of network attacks.

3 Overview of Honeypot

An information system comprising two key elements, decoys and captors, is a honeypot. It seeks to use its data property for the purpose of security investigations for unauthorized and unlawful access. The captor (security program) facilitates safety-related functions, e.g., attack monitoring, prevention, sensing, response, and profiling. The captor provides a comprehensive set of information systems. In addition, to avoid detection, the captors should run in stealth mode. Terminology is not consistent with existing honeypot projects and honeypot research. Some call honeypots decoys. Honey token is the word for a digital object that acts as a decoy.

Stoll used honey tokens, that is, digital files, with security software (we call it captors) to track a German hacker in the book "The Cuckoo's Egg." Thus, the buckling is a decoy, but the system of Stoll's is a buckling system. A weak device without a captor is just a decoy, not a honeypot, according to our description. We do not name anything a honeypot unless it comes with a captor. The two critical elements' organization can be divided into two categories based on respective level of coupling: loose and strong.

The combination refers to the amount of direct knowledge of one component. Loose connections are one in which each component knows little or no separate components. It makes it possible for components to be fully self-reliant and ignorant while still interacting. Tight coupling, by contrast, is if a group is highly dependent or integrated into the very same department to fulfill the mission. An autonomous sweetheart pertains to a sweetheart with close connections. A collaborative sweetheart is used as a loose connection. The decoy is intended to collect data by attack.

The design of a decoy includes several primitive features.

Fidelity: It indicates the degree of exactitude that a decoy provides the attacker with an information system resource. It divides experiences into three categories: low, medium, and high.

Low-Interaction Honeypot (LIH): Adversaries are only provided a limited amount of interaction in a low-interaction honeypot (LIH). The façade decoy is another name for the LIH decoy. A standard LIH is a scheme that resembles operating system (OS) procedures but with a tiny number of the entire features, such as Honeyd [8]. As a result, an attacker would be unable to compromise a LIH since only the fingerprints of OS functions are present, rather than the actual functionality. A LIH may also include a captor to track the façade and record system operations.

Medium-Interaction Honeypot (MIH): The adversaries have a lot more contact with the medium-interaction honeypot (MIH). Unlike LIH, however, the TCP/IP stack cannot be implemented by a MIH. MIHs such as Dionaea [9] and Cowrie [10] are instead attached to sockets, and the OS is left to handle connections. The modeling methodology of the MIHs is based on the emulation of rational implementation replies to incoming requests, contrary to LIHs that implement network protocols. The proposal to the MIH is therefore monitored and investigated, and the captors of the MIH produce fake replies.

High-Interaction Honeypot (HIH): Adversaries will absolutely compromise a high-interaction honeypot (HIH), which is a fully functioning device. Argos [11] and Cuckoo Sandbox [12] are two examples of decoy systems. The HIH must equip security toolkits for device behavior capture and outgoing traffic control because the fully operating honeypot can be breached.

Scalability: It represents the ability to deliver or expand the potential to accommodate this growth in an increasing number of decoys. There are two types of scalability: unscalable and scalable. An unnavigable squad only contains the certain amount of detonators and cannot change its number. For instance, Argos [11] can monitor only one virtual decoy. A scalable honeypot method, on the other hand, can deploy multiple decoys and its captor can control those decoys at the same time, for example, Honeyd [8] can simulate multiple OS fingerprinting constructs at the same time. A honey net is a form of honeypot device that can be scaled up or down.

Adaptability: It concerns the ability to reconfigure the decoy state to adapt to different conditions. It is static and dynamic on two levels. The security researcher must determine the configuration in advance and manually configure it and reset it manually by traditional static honeypots, e.g., Specter [13]. There are several disadvantages in this static configuration system.

- The configuration of honeypots is a complex task manually.
- The system for static configuration cannot respond to an intrusion event immediately.
- The cloned network cannot be adapted to change the goal. A dynamic honeypot can, by contrast, adapt quickly to specific events. It can change regularly or even adapt to changes in the climate in real time and react to intrusion events such as Honeyd.

Role: It describes where the decoy plays in a multifaceted architecture. Two roles can play a honeypot: server and client. This refers to a honeypot detecting or passively capturing unlawful traffic constantly or maliciously.

4 Intrusion Detection and Prevention Using Honeypot Network

Honeypots are characterized by the type of knowledge they offer to invaders. Honeypots are divided into three categories based on their degree of interactivity: low, medium, and strong interactions [14]. Lower or intermediate interface honeypots function in a manner that simulates security-sensitive services that attract intruders. They do not keep accurate and relevant data. Virtualization technology can be used to introduce honeypots on a device. There is no direct opportunity for the intrusion, hijacking the risk of the honeypots is so low. Since the intruder is not hooked up to the real system, fewer comprehensive information can be collected regarding the

attack. Simulation of the services is important in low- and medium-interaction honeypots. Fixed mistakes can assist the intruder to detect the honeypot here, during the simulation process. An unwanted risk may arise from the recognition of honeypots by assailants. When the honeypot is detected by intruders, it loses its attractiveness and aim. In contrast, high interaction offers real services to attract intruders to low-to-middle interactions [13]. However, the risk that intruders can compromise the honeypot and the provider is greater as they actually interact with true facilities. Some safety measures on the firewall side may be taken as one of the alternatives to this deformity. The intruder with highly interactive honeypots means that we can gather more crucial data about the hackers. Moreover, the recognition of the honeypot becomes more difficult, as intruders deal with actual facilities. There are also several remotely implemented applications that detect any inappropriate activity. In addition, using the virtual machines, this type of honeypot is only possible on a physical system connected to the network. Moreover, its prices are rising than its medium and low interface honeypots; the time and effort for upkeep are also greater. The cost is also higher. Moreover, these honeypots can themselves create security vulnerabilities. Strict measures should also be taken to prevent intruders from penetrating the honeypot and compromising the device, reducing the possibility of new data breaches.

Honeypots are considered to be a good technology for tracking programming and improving safety instruments' viability. Specially built to engage hackers and to mislead hackers, honeypots are often designed to detect malicious activities that occur across the Internet and can be considered an effective tool for tracking hacker behavior. Honeypots can be identified as systems or assets used to trap and track erroneous requests within a network and also to identify them [14] (Fig. 3).

They range from low-to-medium and high interactions and have their advantages and disadvantages for every single form. Their goal is to observe, comprehend, control, and track the actions of an attacker to build systems not only safe but also

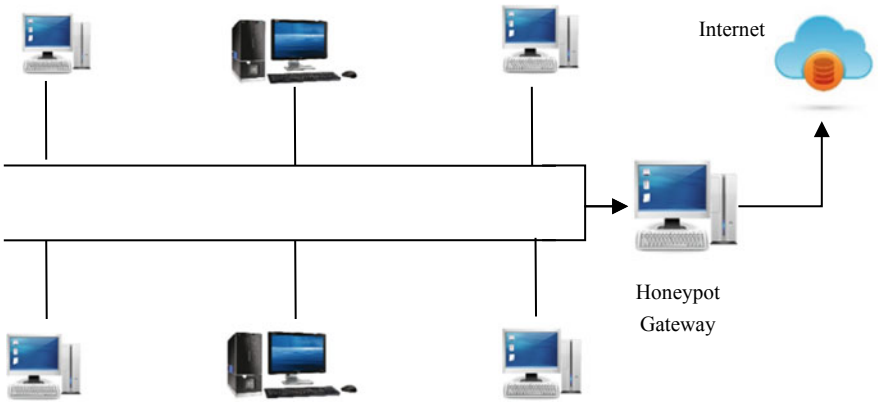


Fig. 3 Functioning of honeypots

capable of handling such traffic. We want it to be checked, attacked, or compromised by a closely monitored computer resource. "Besides, it is a resource of the information system which is valued for unauthorized or unlawful use of it."

5 Related Work

Negi et al. [7] addressed the use of honeypot for the detection of attacks in a cloud-based environment and designed a framework to handle it.

To address the issues mentioned above, Chakkaravarthy et al. [14] proposed an intrusion detection honeypot (IDH). This method consists of systems termed as (i) honeyfolder, (ii) AuditWatch, (iii) complex processing of events (CEP). Honeyfolder is an early warning system that uses the Social Leopard Algorithm (SoLA) as the decoy folder. It alerts the user during suspicious activities of the file. AuditWatch is an entropy module that checks file and folder entropy. The CEP engine is used to aggregate data from various safety systems to confirm and respond to ransomware behavior.

The SSH and telnet honeypot detection techniques were studied and evaluated by Surnin et al. [15]. Besides, the author suggested a new method for probabilistic honeypot detection estimation and automatic software that has been implemented.

Bhagat et al. [16] discussed the primary importance of network security in protecting every organization against cyber-attacks of any type. The analysis of traffic in the network is an essential task for this purpose. Network traffic analysis is done through the installation of various network intrusion detection (NID) devices. A honeypot is one of the most important devices to detect network intrusions. Honeypots interact with the attacker and gather data that can be analyzed to collect information about the attacks and network attackers. Honeypots are security tools by which the attacker targets and generates attack data logs. Honeypots provide small quantities of relevant data so that safety inspectors can easily understand and research can be carried out. This paper simulates honeypot communicating with attackers and analyzes data obtained using honeypot using different instruments.

To guarantee enhanced performance, enhanced protection in cloud computing environments, and reduced hazards in the cloud environment, Rajvi et al. [17] suggested the use of honeypot's intrusion detection and prevention system (IDPs) model, focusing on the issue of cloud-based data storage. Through the use of an organization's organized IDPS and honeypot, the proposed system is feasible and the model is successful. This mechanism is used both to recognize different attacks and deny access through the proposed intrusion prevention scheme, using both the anomaly detection (AD) and the signature detection (SD) in collaboration (IPS). The purpose of this report is the use of honeypot to emphasize, identify, and trap internal intruders.

Mehta et al. [18] spoke of architectures for systems that could predict the threat of using Honeypot as a data source and different machine learning algorithms to predict accurately using OSSEC as a host intrusion detection [HIDS], Honeyd is a honeypot open source.

The effective detection Muraleedharan et al. [19] have proposed a honeypot and swarm intelligence system. The layout of the honeypot allows agents to track the intruders strategically. This intrusion detection process decreases the rate of false alarms triggered by denial-of-service attacks. A thorough description of the attack is compiled with pattern recognition for potential attacks. Before breaching the network, the proposed system is tested based on the precision and speed of the intruder detection.

Vargas et al. [20] developed low-interaction honeypot for intruders over VoIP. Shukla et al. [21] extended the honeypot principle to the identification of insecure site URLs. The developed system is positioned on the side of the client using Python's programming language. On the client side, they installed a crawler that collects URL addresses, checks whether there is a valid reason for a visit, and then enables the sites to be accessed.

Honeypots were used to evaluate and visualize malicious activities by Koniaris et al. [22]. This study deployed two honeypots for alternative search reasons, Koniaris and his collaborators. One of the honeypots was set up for self-propagation, with the intention of discovering malicious programs, and the other was set up as a pit to capture fraudulent behaviors. Li et al. [23] developed an IDS using honeypot. The results of their work were an amount of reports attempts that could enhance the protecting power of the honeypot scheme. Chawda and Patel [24] came up with the idea of using honeypots to investigate potential threats, and they suggested a shared honeypot method to do so. The honeypot used in this study is a reduced type that is used as a front-end track item. Suo et al. [25] suggested to resolve IDS problems using honeypot. Apart from that, Paul et al. [26] proposed honeypot using signature-based architecture to defend against polymorphic worm attacks. Furthermore, the suggested technique was suitable for the analysis of transport, where suspect motorway was segregated and used to collect knowledge about particular assaults, in particular occupational assaults. In addition, if the system cannot find new attacks for unknown worm types, the developed signature-based system would produce the signature for that particular purpose. Beham et al. [27] used some advantages of virtualization technology with honeypot to detect intrusion in the network. Essentially, the research compared existing honeypot schemes, embedded virtual machines, and intrusion detection based on virtual machine introspection (VMI) (Table 1).

Table 1 Comparative analysis of existing systems

Ref.	IDS type	Technique	Attack	Results	Limitations
[14]	NIDS and HIDS	Honeypot using Social Leopard Algorithm	Ransomware	Cost and time efficient	Effective for HIDS. Needs improvement in learning constraints for NIDS
[17]	NIDS	Honeypot	Telnet	With honeypot detection rate is high	With increase in malicious packet detection rate decreases
[28]	NIDS	Neural network	Trigger	–	Signature searching failure in case of unknown attack
[29]	NIDS	Fuzzy logic	Zero-day attack	~97% accuracy	Not able to detect under changing network scenario
[30]	NIDS	Fuzzy logic	Cyber	FL-SnortIDS achieve high detection rate	Not able to detect under changing network scenario
[31]	NIDS	Fuzzy logic	Cloud attacks	Successful interaction = 20 s	Model overhead and characteristics are not evaluated
[32]	WSN	Honeypot	Poisoning DNS spoofing Denial-of-Service	Minimum false alarm rate was 15%	High false alarm rate. Deployed for small network
[33]	WSN	Trace back honeypot	–	Increased throughput	Few network features are selected
[34]	NIDS	Honeypots	Insider threat	22% improvement in detection	Disk space is an issue as large amount of data are generated
[35]	NIDS	Honeypots based on game theory	Insider threat	Decreases energy consumption	Not applicable to the real scenario

6 Proposed Methodology

Network security is critical in today's world since every organization uses the Internet. Although there is extensive research in this area, the protection of systems linked on the Internet is still not guaranteed. Attackers every day discover new and creative strategies to take advantage of network security. New tools or software are being

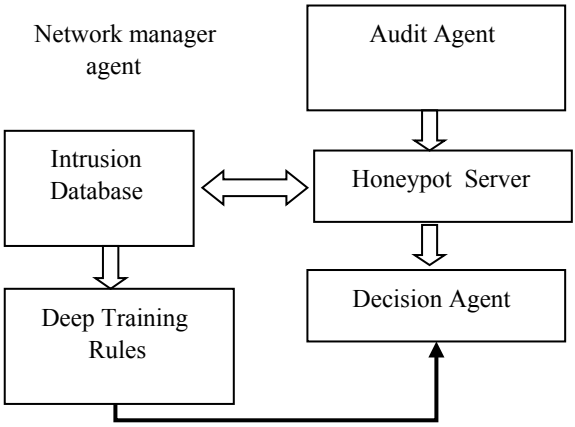


Fig. 4 Proposed honeypot server-based framework

developed to combat the attacker's strategies to solve this security issue, but still, it is a highly demanding research area. Therefore, the idea of honeypots was developed by researchers to communicate with the intruder. Honeypots are decoy systems that are used in the network to attract attackers, and thus, they get stuck in the decoy system. Their activities are further analyzed for the study. Many intrusion detection systems have trouble solving optimization issues. These issues have an increasing search space, which increases with time. In this paper, a deep learning-based honeypot framework for cloud security is proposed. The proposed algorithm flow diagram of the intrusion detection model is shown in Fig. 4.

The proposed framework consists of the following stages.

The proposed model consists of four agents that communicate with each other so that they can better identify network attacks.

Network Manager Agent: This agent controls the intrusion database training process and prepares the deep learning network for decision-making rules.

Audit Agent: This unit performs tasks of trapping the network traffic through the use of sniffing protocols by listening to the network.

Honeypot Server: The data will be analyzed by this agent. On each host, the honey agent is installed and continuously monitors the host. The agent honeypot is run on the server honeypot. The honey agent continually monitors and checks if the activity is a user activity or ransomware activity if it is changing from one state to another.

Decision Agent: After reviewing and processing uncategorized data packets, this agent will be responsible for making a decision. It allows network and security administrators to perform reporting tasks on all detected intrusions by interacting with the intruder database and deep training algorithm.

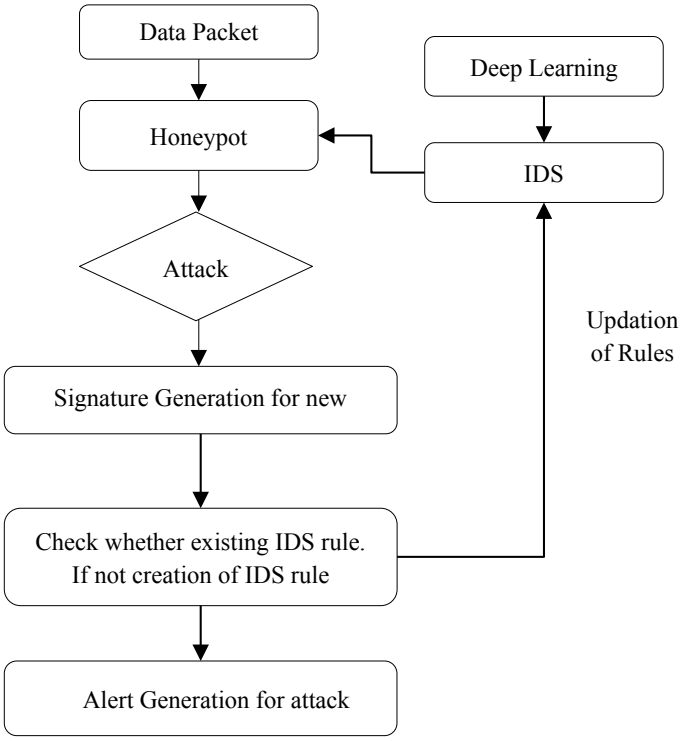


Fig. 5 Flowchart of proposed framework

The flowchart of the working of the proposed framework is shown in Fig. 5.

Algorithm: deep learning-based honeypot intrusion detection system

1: Begin
2: Input: Intrusion Feature set = $\{F_1, F_2, \ldots F_n\}$
3: Output: Intrusion Report → Data(status)
4: Agent initialization and deployment
5: Feature Extraction
6: Activate Honeypots
7: DNN training
8: Decision-making agent → Data(status)
9: Generate alert
10: Update learning model
11: End

6.1 Advantages of Proposed Framework

Zero-Day Attacks Detection: The proposed HPIDS is a combination of both the IDS and the honeypot that allows both known and zero-day attacks to be detected. This would make it better and more secure. The honeypot module is used to detect attacks of zero-days, while the IDS module is used to detect known attacks.

Cloud Supported Framework: The majority of the current IDS-based cloud protection systems as studied above have been taken into account and can handle only one or two attacks. The scope of the proposed honeypot-based IDS is not precise but can include general attacks, which have been a problem with current IDS-based solutions for one or two attacks.

Lightweight with Less Resource Usage: The proposed framework is lightweight in terms of less resource usage for its operation, given the resource constraint problems in the cloud.

High Detection Rate: With the implementation of honeypot modules into the nodes along with the IDS, the identification rate of attacks has been improved as this system includes both known and unknown attacks.

6.2 Factors Needed for Proposed Framework

Relevant factors to be taken into account during the design and development of the proposed honeypot IDS for the cloud are.

Coordination with the Honeypot Server: The coordination of the IDS database and decision unit with the honeypot server is a key concern to our proposed architecture for effective working. The honeypot server regularly updates the IDS database about malicious activities as well as automatic generation of rules against logged attack patterns, etc.

Selection of IDS Methodology: Another important factor requiring factor is the type of deep learning rules for IDS to be deployed on individual nodes to classify data packets as normal or anomaly, as different learning rules have their advantages and disadvantages.

7 Conclusion

In the network for intrusion detection honeypot is one of the important units that can interact with the attacker, collect information from them and analyze attackers in the network. In this paper, an in-depth and comprehensive description of IDS and honeypots for cloud computing is provided. Different research works are also analyzed and studied using machine learning approaches with honeypots. Furthermore, this paper includes a proposed architecture including honeypot-based IDS solutions for

enhancing security in the cloud. Deep learning is also integrated with IDS for the detection of known as well as unknown attacks. This proposed architecture will lead to a better detection rate of unknown attacks over cloud.

References

1. Huang RW, Yu S, Zhuang W, Gui XL (2010) Design of privacy-preserving cloud storage framework. In: International conference on grid and cloud computing, GCC, pp 128–132
2. Wang L, Tao J, Kunze M, Castellanos AC, Kramer D, Karl W (2008) Scientific cloud computing: early definition and experience. In: IEEE international conference on high performance computing and communications, pp 825–830
3. Wang C, Wang Q, Ren K, Lou W. Ensuring data storage security in cloud computing. https://eprint.iacr.org/2009/081.pdf
4. Sarkar MK, Kumar S (2016) Ensuring data storage security in cloud computing based on hybrid encryption schemes. In: International conference on parallel, distributed and grid computing, PDGC. Institute of Electrical and Electronics Engineers Inc., pp 320–325
5. Kandukuri BR, Ramakrishna PV, Rakshit A (2009) Cloud security issues. In: IEEE international conference on services computing, pp 517–520. https://doi.org/10.1109/SCC.2009.84
6. Goyal K, Kinger S (2013) Modified Caesar cipher for better security enhancement. Int J Comput Appl 73
7. Negi PS, Garg A, Lal R (2020) Intrusion detection and prevention using honeypot network for cloud security. In: International conference on cloud computing, data science and engineering. Institute of Electrical and Electronics Engineers Inc., pp 129–132
8. Provos N (2004) A virtual honeypot framework. Conference of USENIX security symposium. CA, USA, Berkeley, pp 1–14
9. Dionaea—Catched Bugs (2011 November) [Online]. Available: http://dionaea.carnivore.it/
10. Oosterhof M (2015) Cowrie—active kippo fork. Available at: http://www.micheloosterhof.com/cowrie/
11. Portokalidis G, Slowinska A, Bos H (2006) Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. ACM SIGOPS/EuroSys Eur Conf Comput Syst 15–27
12. CuckooFoundation (2014) "Cuckoo," [Online]. Available: http://www.cuckoosandbox.org/
13. Spitzner L (2003) Specter: a commercial honeypot solution for windows [Online]. Available: http://www.symantec.com/connect/articles/specter-commercial-honeypot-solution-windows/
14. Chakkaravarthy S, Sangeetha D, Cruz MV, Vaidehi V, Raman B (2020) Design of intrusion detection honeypot using social leopard algorithm to detect IoT Ransomware attacks. IEEE Access 8:169944–169956. https://doi.org/10.1109/access.2020.3023764
15. Surnin O, Hussain F, Hussain R, Ostrovskaya S, Polovinkin A, Lee JY, Fernando X (2019) Probabilistic estimation of honeypot detection in internet of things environment. In: International conference on computing, networking and communications, ICNC 2019. Institute of Electrical and Electronics Engineers Inc., pp 191–196. https://doi.org/10.1109/ICCNC.2019.8685566
16. Bhagat N, Arora B (2018) Intrusion detection using honeypots. In: International conference on parallel, distributed and grid computing. Institute of Electrical and Electronics Engineers Inc., pp 412–417
17. Ravji S, Ali M (2018) Integrated intrusion detection and prevention system with honeypot in cloud computing. In: International conference on computing, electronics and communications engineering, iCCECE. Institute of Electrical and Electronics Engineers Inc., pp 95–100
18. Mehta V, Bahadur P, Kapoor M, Singh P, Rajpoot S (2015) Threat prediction using honeypot and machine learning. In: International conference on futuristic trends in computational analysis

and knowledge management, ABLAZE 2015. Institute of Electrical and Electronics Engineers Inc., pp 278–282
19. Muraleedharan R, Osadciw LA (2009) An intrusion detection framework for sensor networks using honeypot and swarm intelligence. In: International conference on mobile and ubiquitous systems: networking and services, MobiQuitous
20. Vargas IRJDS, Kleinschmidt JH (2015) Capture and analysis of malicious traffic in VoIP environments using a low interaction honeypot. IEEE Lat Am Trans 13:777–783
21. Shukla R, Singh M (2014) Python Honey Monkey: detecting malicious web URLs on client side honeypot systems. In: INFOCOM technologies and optimization proceedings of 3rd international conference on reliability, pp 1–5
22. Koniaris I, Papadimitriou G, Nicopolitidis P, Obaidat M (2014) Honeypots deployment for the analysis and visualization of malware activity and malicious connections. In: IEEE international conference on communications (ICC), pp 1819–1824
23. Li S, Zou Q, Huang W (2014) A new type of intrusion prevention system. In: International conference on information science, electronics and electrical engineering, pp 361–364
24. Chawda K, Patel AD (2014) Dynamic and hybrid honeypot model for scalable network monitoring. In: International conference on information communication and embedded systems, pp 1–5
25. Suo X, Han X, Gao Y (2014) Research on the application of honeypot technology in intrusion detection system. In: IEEE workshop on advanced research and technology in industry applications (WARTIA), pp 1030–1032
26. Paul S, Mishra BK (2013) Honeypot based signature generation for defense against polymorphic worm attacks in networks. In: IEEE international advance computing conference (IACC), pp 159–63
27. Beham M, Vlad M, Reiser HP (2013) Intrusion detection and honeypots in nested virtualization environments. In: IEEE/IFIP international conference on dependable systems and networks (DSN), pp 1–6
28. Gupta D, Gupta S (2017) An efficient approach of trigger mechanism through IDS in cloud computing. In: IEEE Uttar Pradesh section international conference on electrical, computer and electronics, UPCON 2017.Institute of Electrical and Electronics Engineers Inc., pp 68–72
29. Alqahtani SM, John RA (2017) Comparative analysis of different classification techniques for cloud intrusion detection systems' alerts and fuzzy classifiers. In: Proceedings of computing conference. Institute of Electrical and Electronics Engineers Inc., pp 406–415
30. Alqahtani SM, John R (2017) A comparative study of different fuzzy classifiers for cloud intrusion detection systems' alerts. In: IEEE symposium series on computational intelligence. Institute of Electrical and Electronics Engineers Inc.
31. Sule M, Li M, Taylor G, Onime C (2017) Fuzzy logic approach to modelling trust in cloud computing. IET Cyber Phys Syst Theory Appl 2(2):84–89
32. Umamaheswari A, Kalaavathi B (2019) Honeypot TB-IDS: trace back model based intrusion detection system using knowledge based honeypot construction model. Cluster Comput 22(6):14027–14034
33. Agrawal N, Tapaswi S (2017) The performance analysis of honeypot based intrusion detection system for wireless network. Int J Wirel Inf Netw 24(1):14–26
34. Yamin MM, Katt B, Sattar K, Ahmad MB (2020) Implementation of insider threat detection system using honeypot based sensors and threat analytics. In: Lecture notes in networks and systems, vol 70. Springer, pp 801–829
35. Gill KS, Saxena S, Sharma A (2020) GTM-CSec: game theoretic model for cloud security based on IDS and honeypot. Comput Secur 92:101732

Optimal Real-time Pricing and Sustainable Load Scheduling Model for Smart Homes Using Stackelberg Game Theory

Karthikeyan Dheeraja, R. Padma Priya, and Thakur Ritika

Abstract Governments today are striving at their best to keepup with the "Energy for all" motto among their citizens. Today, majority of the electricity is generated through non-renewable resources, but these resources are getting depleted day by day. With more and more connected things around us, the drive and thrive for efficient electricity load scheduling has been an important focus, in order to ensure availability of electricity till the last mile. It is also observed that few usage hours experience over-consumptions, amid the possible presence of under-consumption usage hours. We, in this paper, propose an efficient load scheduling model for achieving a sustainable power system framework using reverse Stackelberg game technique. Since this game is played between the Electricity Department and domestic consumers, it is a single leader multi-follower problem. This reverse Stackelberg model handles a bi-level programming problem, with a single objective at the upper level (leader) and two objectives at the lower level (followers). At the upper level, we have the Electricity Department, whose aim is to propose a real-time tariff plan such that the electricity demand of his customers is load-balanced throughout the day. Contrastingly at the lower level, we have domestic users, who seek to schedule usage of their appliances in such a way that their two objectives, namely minimization of electricity bills and maximization of satisfaction, are attained. The leader's and followers' problems have been solved using genetic algorithm and greedy approach, respectively, which yielded a reduction of up to 68% domestic load consumption during peak hours. An average of 33% and markdowns of up to 72% were observed in the bimonthly bills of high power-consuming users during the experimental runs. Thus, this model has been proposed aiming for effective and sustainable benefits, as the Indian government is moving towards installation of smart meters in smart homes under smart cities development project plans.

K. Dheeraja (✉)
FireEye Technologies Pvt. Ltd., KA 560029, India

R. Padma Priya
School of Computer Science and Engineering, Vellore Institute of Technology,, TN 632014 Vellore, India

T. Ritika
Microsoft Global Service Center (India) Pvt. Ltd., Hyderabad 500032, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_22

Keywords Stackelberg game · Sustainable energy · Genetic algorithm · Electricity load scheduling · Satisfaction modeling

1 Introduction

With an installed capacity of 371.054 GW in the national electric grid (as of June 2020) [3], India is currently the third largest producer and consumer of electricity in the world [17]. However, despite low electricity tariff, the per capita consumption of electricity has been observed to be lower than that of most countries in the world [1]. This is because domestic sector contributes only 24.76% [2, 14, 15] to the overall electricity consumption in India. When too many users of different sectors (say industrial, domestic, etc.) serviced by the same electricity hub consume electricity at the same time reaching maximum power demand, the hub may reach its peak load. These peak-load consumptions lead to pitfalls in energy availability, and around 11% of such energy deficits have been observed every year in India. Further, there may even arise power-grid failures [9] if the available electricity supply is in excess than needed to meet the current demands.

Currently, India manages energy-deficit problems by rolling out power shutdowns or blackouts for low-priority consumers [10]. These low-priority consumers are mostly domestic users, because industrial users cannot compromise on electricity during peak hours of the day (11A.M–4 P.M). Sometimes, energy deficits are also managed by generating extra energy, borrowing from other providers, or using previously stored energy. These may serve as immediate solutions during emergencies, but these are not always reliable or cost-effective. This creates a pressing need to uniformly distribute the electricity consumption of domestic users across 24 hours of a day and more importantly reduce the consumption during peak hours, in order to satisfy everyone equally. This can be done by Demand Response (DR) which is a strategy to reduce or shift the electricity consumption of customers to match energy demand with the supply in a better way. In this paper, we propose one such kind of a strategy which is applicable to the state of Tamil Nadu in India.

Presently, the Indian government has adopted fixed tariff plans for its electricity supply [16] and demonstrates such a tariff proposed by the Tamil Nadu Electricity Regulatory Commission (TNERC) for domestic users in Tamil Nadu. Here, the first 100 units are considered free for all domestic users. The price to be paid for the remaining units is dependent on the total units consumed. However, the Indian government has proposed to convert all the existing meters to smart meters by 2022 [6]. This indirectly indicates a move toward attaining DR easily. But, it is only possible when the tariff is altered continuously, while otherwise the existing electricity usage pattern is bound to continue. This kind of an adapting price scheme is called real-time pricing (RTP). One type of RTP is Time of Use (ToU) tariff, wherein the electricity consumption at different times of a day is charged with different prices, which helps reduce the overall demand and shift usage from peak periods of the day to non-peak or low-demand periods. Typically, ToU tariffs are announced to customers some

duration (week or months) in advance, but this does not entirely provide a real-time control over imbalances in energy supply [10].

In [12], the authors have worked upon a possiblity of economic cost invested in building renewables-embedded microgrids for the state of Tamil Nadu, India. Similarly, in this paper, we propose a suitable tariff plan for the electricity stakeholders in Tamil Nadu. We present a day-ahead ToU electricity RTP (DA-RTP) model through a reverse Stackelberg game between the Electricity Board (1-leader) and the domestic users (n-followers). Since the primary goal of a governmental organization is not to make profit, but to provide services to citizens in a sustainable manner, the aim of our leader is to balance the domestic usage of his power grid over 24 hours. By increasing the electricity prices significantly during peak hours of demand and reducing them during non-peak hours, we can enable the consumers to shift their usage to non-peak hours. Since increasing the selling price during peak demand also increases profit for the seller (leader), in this paper, we consider the objective of the leader in our Stackelberg game as profit maximization. Although residential customers are served primarily by the electricity department of the Indian government, without loss of generality, our model is applicable for private electricity providers too.

Reverse Stackelberg games belong to the category of non-cooperative game theory since the players involved in the game act exclusively, relative to their own strategies [7]. Electricity pricing is a common application of reverse Stackelberg games, where the leader (here, the EB) can produce an objective function in a way that induces his followers to behave as he desires. He is also able to induce a collectively cooperative behavior of his multiple followers, while they behave non-cooperatively among themselves. The crucial distinction between the original and reverse Stackelberg games is that the former imposes a completely top-down-leveled scheme, whereas the latter allows much more communication from the lower to the higher levels. This kind of communication is specifically necessary in our problem because the leader's profit function is dependent on the output of his followers.

Over a decade, lot of research has been done in the field of electricity load scheduling, with an effort to continuously improve existing works in literature. Meng, F.L. and Zeng X.J. discuss a new way of proposing electricity tariff and scheduling appliances using a Stackelberg game model with 1-leader and n-followers [11]. This has been implemented using genetic algorithm for profit-maximization problem on the leader's side and linear programming for scheduling appliances on the followers' side. Similarly, we also adopt genetic algorithm to solve our leader's objective, but unlike them, we use real-valued chromosomes, tournament selection, uniform mutation, and uniform crossover in this paper. We also employ a different strategy to solve our appliances scheduling problem. Moreover, while they have not focused on considering the satisfaction parameter of electricity consumers, our model ensures that followers are satisfied with the prescribed 24-hours schedule of their appliances.

Das N.R. et al. propose a much simpler model for bill and dissatisfaction minimization problem using particle swarm optimization [5]. Unlike them, in this work, we have adopted an integrated model using Stackelberg games to consider the objectives of both the electricity provider and the consumers. Yang P. et al. have employed a game-theoretic approach to increase profit for the electricity providers and bene-

fits for the users [18]. They take into consideration all types of users like domestic, commercial, and industrial users and arrive at a Nash equilibrium using back induction. However, Huang Y. has proposed a two-layer framework for this problem [8]. The lower layer is dedicated to scheduling appliances at the users' side and generating electricity at the utility's side, whereas the upper layer employs a Stackelberg dynamic game approach to model the interaction between electricity providers and consumers. Rao M.R. et al. have proposed a model for load scheduling at the customer's side to minimize the electricity tariff considering their dissatisfaction factor [13]. They have worked with three different algorithms and have inferred that Monte Carlo simulation based on M-H algorithm gives the most optimal/minimized bill.

This paper is organized as follows. Section 2 describes the system model, and Sect. 3 defines the mathematical formulations involved in this framework. The methodology and algorithms are described in Sect. 4, followed by the results in Sect. 5. The final section portrays the conclusion.

2 System Model

Our reverse Stackelberg model handles a bi-level programming problem, with a single objective at the upper level (Leader) and two objectives at the lower level (Followers). Here, the Electricity Board (EB) acts as the sole leader of the game, while all the users coming under the provision of that particular EB are the followers. Although in reality there are various EBs covering different parts of a state or district, all of them work with same motives without mutual competition. So, we take only one such EB into consideration for simplicity, while this model is applicable to all the EBs in existence.

The government provides EB with maximum and minimum price per unit allowed for each hour of the day. Using this, the EB sets its tariff for the next day and communicates this to the smart meters of their customers. It is assumed that smart meters with inbuilt capability of two-way communication are installed at every house, which will be managing the schedule of all the appliances at each house and thus the billing. Figure 1 depicts the architecture of the system, wherein A_u denotes the total set of appliances of a user u.

Z_{u,A_u} represents the total electricity consumed by the appliances of user u. Further, let P be the set of 24-hour prices that the leader proposes, i.e., $P = [P_1, P_2, \cdots, P_{24}]$, where P_1 is the price between 12 A.M. to 1 A.M., P_2 is the price between 1 A.M. and 2 A.M., and so on. Let $H = \{1, 2, 3, \cdots, 24\}$ denote the set of total hours in a day. The government imposes a limit on the minimum and maximum prices that the leader can offer for each hour, i.e., $\forall h \in H$, $P_h^{min} \leq P_h \leq P_h^{max}$.

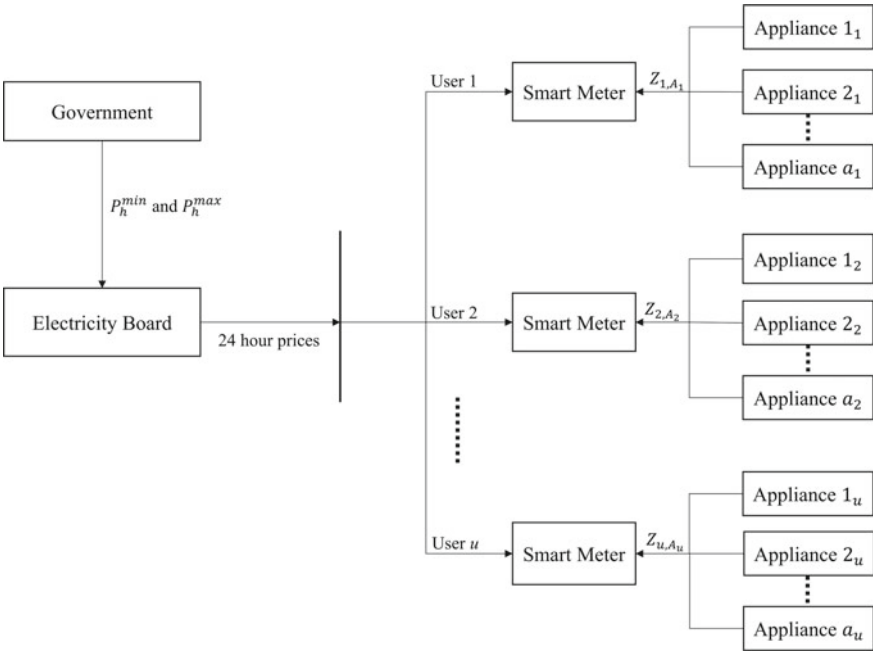


Fig. 1 System architecture

3 Problem Formulation

The proposed system consists of two goals: (i) minimizing the followers' bill amount and dissatisfaction and (ii) maximizing the leader's profit.

3.1 Follower's Problem

Let U denote the total set of users or homes (followers) using electricity under our leader. All the appliances are represented as a set A_u, $\forall u \in U$. Further, we categorize these appliances into three types: *reschedulable, non-reschedulable*, and *curtailable appliances*. *Reschedulable appliances* are those whose usage can be shifted to a different time period within H, where the prices are relatively low. High power-consuming appliances like washing machines, dish washers come under this category, and we represent the set of these appliances as R_u, $\forall u \in U$. Contrastingly, *non-reschedulable appliances* are those whose operation is mandatory in the preferred time period, irrespective of the prices. Appliances like clocks and usage of tubelights in the evening cannot be rescheduled, and the set of these appliances is represented as N_u, $\forall u \in U$. *Curtailable appliances* can be defined as those whose usage period cannot be shifted, but energy consumption can be controlled based

on the tariff for each hour. These can be devices with dynamic voltage frequency scaling (DVFS) technology, air conditioners, space heaters, etc. We represent the set of these appliances as $C_u, \forall u \in U$. Hereby, we have $A_u = R_u \bigcup N_u \bigcup C_u$ and $R_u \bigcap N_u \bigcap C_u = \phi, \forall u \in U$.

Each user $u \in U$ has to provide a preferred starting and ending time within which they wish to operate each appliance $a \in A_u$. These times are represented as $\alpha_{u,a}$ and $\beta_{u,a}$, respectively, where $\alpha_{u,a}, \beta_{u,a} \in H$. Every appliance has a power rating that implies how much electricity it uses. Low power-consuming appliances like fans and bulbs commonly have a constant value of this rating, while most of other appliances can operate over a range of values. We assume this rating to be in kWh since the electricity bills in India are calculated in terms of the number of units used, where 1 unit = 1 kWh. Each user $u \in U$ provides the minimum and maximum power rating $\theta_{u,a}^{min}$ and $\theta_{u,a}^{max}$, respectively, $\forall a \in A_u$.

Arriving at the best schedule for these appliances involves equal consideration of two separate problems: (i) user satisfaction and (ii) bill calculation.

3.1.1 Modeling Reschedulable Appliances

User satisfaction for reschedulable appliances

It is important for the operating time of an appliance to be convenient to the users as important as it is to minimize their payment bills. So, we use a satisfaction factor along with the bills to determine the schedule of a reschedulable appliance. The users provide a satisfaction rate vector S that determines the level of satisfaction the users want for each of their appliances. Here, $\forall s_{r,u} \in S$, $s_{r,u} > 1$, $where \forall r \in R_u$ and $u \in U$. For higher satisfaction of appliance $r \in R_u$, the value of $s_{r,u}$ should be set higher.

Consider that the smart meter wants to schedule an appliance $r \in R_u$, of user $u \in U$, within the interval $\left[t, t + \left(\beta_{u,r} - \alpha_{u,r}\right)\right]$. The farther an appliance is rescheduled from its preferred starting time $\alpha_{u,r}$, higher is the dissatisfaction caused to the user. This dissatisfaction is determined by the factor $\omega_{u,r} = d/d_{\max}$, where $d = \left|\alpha_{u,r} - t\right|$ is the absolute difference between preferred starting time and rescheduled starting time of appliance $r \in R_u$. d_{max} is the maximum possible value of d (which is a constant value of 23). Using this, the dissatisfaction caused by scheduling appliance $r \in R_u$ at a time $t \in H$ is calculated as:

$$D_{u,r} = {S_{u,r}}^{\omega_{u,r}}$$

Bill calculation for reschedulable appliances

We define a power-consumption vector $Z_{u,r} = [z_{u,r}^1, z_{u,r}^2, \cdots, z_{u,r}^{24}], \forall u \in U$ and $r \in R_u$, where $z_{u,r}^h$ represents the units of power consumed by appliance r of user u at any hour $h \in H$. For simplicity and to clearly distinguish between reschedulable, non-reschedulable, and curtailable appliances, we assume that these type of appliances

operate over a constant power rating, i.e., $\theta_{u,r}^{min} = \theta_{u,r}^{max}$, $\forall r \in R_u$. So, the power-consumption vector $Z_{u,r}$ can be generated as,

$$z_{u,r}^{h} = \begin{cases} \theta_{u,r}^{\min}, & \text{where } t \leq h \leq t + \left(\beta_{u,r} - \alpha_{u,r}\right) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Therefore, the contribution of these appliances to the electricity bill of user $u \in U$ is:

$$BR_u = \sum_{r \in R_u} \left[\sum_{h=t}^{t+(\beta_{u,r}-\alpha_{u,r})} \left(P_h \times z_{u,r}^{h}\right) \right], \text{ where } r \in R_u \text{ and } P_h \in P \tag{2}$$

Thus, the best schedule for a reschedulable appliance is $\left[t, t + \left(\beta_{u,r} - \alpha_{u,r}\right)\right]$ which has the least value for the equation:

$$ER_u = \delta_{1,u,r} \times BR_{u,r} + \delta_{2,u,r} \times D_{u,r} \tag{3}$$

where $\delta_{1,u,r}$ and $\delta_{2,u,r}$ are values that determine the importance level of minimized bill and satisfaction, respectively, provided by the user u for appliance r,. It is also important to note that $\delta_{1,u,r} + \delta_{2,u,r} = 1$.

3.1.2 Modeling Non-reschedulable Appliances

Bill calculation for non-reschedulable appliances

We define an electricity-consumption vector $Z_{u,n} = [z_{u,n}^{1}, z_{u,n}^{2}, \cdots, z_{u,n}^{24}]$, $\forall u \in U$ and $n \in N_u$, where $z_{u,n}^{h}$ represents the electricity consumed by appliance n of user u at any hour $h \in H$, and it is assumed to remain constant during each hour of operation, i.e., $\theta_{u,n}^{min} = \theta_{u,n}^{max}, \forall n \in N_u$ Thus, the power-consumption vector $Z_{u,n}$ can be generated as,

$$z_{u,n}^{h} = \begin{cases} \theta_{u,n}^{\min}, & \text{where } \alpha_{u,n} \leq h \leq \beta_{u,n} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Therefore, payment calculation problem for these appliances can be solved as:

$$BN_u = \sum_{n \in N_u} \left[\sum_{h=\alpha_{u,n}}^{\beta u,n} \left(P_h \times z_{u,n}^{h}\right) \right] \tag{5}$$

User satisfaction for non-reschedulable appliances

Since there is not any rescheduling (time-shifting) involved in modeling non-reschedulable appliances, the users set the values of $\delta_{1,u,n}$ and $\delta_{2,u,n}$ as one and

zero, respectively. This indicates that there is not any dissatisfaction caused to the user, and he is completely focused just on the bill. Thus, to obtain the best schedule for non-reschedulable appliances, we write an equation similar to (3) as:

$$EN_u = \delta_{1,u,n} \times BN_{u,n} + \delta_{2,u,n} \times D_{u,n} \tag{6}$$

where $\delta_{1,u,n} = 1$ and $\delta_{2,u,n} = 0, \forall u \in U$ and $n \in N_u$.

3.1.3 Modeling Curtailable Appliances

Bill calculation for curtailable appliances

The power-consumption vector for these appliances is defined as $Z_{u,c} = [z^1_{u,c}, z^2_{u,c}, \cdots, z^{24}_{u,c}], \forall u \in U$ and $c \in C_u$, where $z^h_{u,c}$ represents the electricity consumed by appliance c of user u at any hour $h \in H$. For each of these curtailable appliances, there exists a minimum power level $\theta^{min}_{u,n}$ and maximum power level $\theta^{max}_{u,n}$ within which they can operate. Consider that the price vector during the desired operation interval $[\alpha_{u,c}, \beta_{u,c}]$ of user u for appliance c and $h \in H$ is $P_{u,c} = [p_{\alpha_{u,c}}, p_h, \cdots, p_{\beta_{u,c}}]$, where $P_{u,c}$ is a subarray of P. Let $p^{min}_{u,c}$ and $p^{max}_{u,c}$ represent the minimum and maximum prices within $P_{u,c}$. Using this, the energy consumption of appliance c of user u at hour $h \in H$ with price p_h can be modeled as a function of the prices like,

$$z^h_{u,c} = p^{\max}_{u,c} - \left[\frac{\left(\theta^{\max}_{u,c} - \theta^{\min}_{u,c}\right) \times \left(p_h - p^{\min}_{u,c}\right)}{p^{\max}_{u,c} - p^{\min}_{u,c}} \right] \tag{7}$$

This way, the minimized bill for curtailable appliances of user $u \in U$ is calculated as:

$$BC_u = \sum_{c \in C_u} \left[\sum_{h=\alpha_{u,c}}^{\beta_{u,c}} \left(p_h \times z^h_{u,c}\right) \right] \tag{8}$$

User satisfaction for curtailable appliances

Similar to non-reschedulable appliances, the users set the values of $\delta_{1,u,c}$ and $\delta_{2,u,c}$ as one and zero, respectively, since modeling of curtailable appliances does not involve time-shifting but only usage reduction. Thus, the best schedule for curtailable appliances can be obtained as:

$$EC_u = \delta_{1,u,c} \times BC_{u,c} + \delta_{2,u,c} \times D_{u,c} \tag{9}$$

where $\delta_{1,u,c} = 1$ and $\delta_{2,u,c} = 0, \forall u \in U$ and $c \in C_u$.

Therefore, minimized bill of satisfactory schedule for all appliances of user u is:

$$BA_u = BR_u + BN_u + BC_u \tag{10}$$

where $BR_u \Rightarrow \min ER_u$, $BN_u \Rightarrow \min EN_u$ and $BC_u \Rightarrow \min EC_u$

3.2 Leader's Problem

Let M represent the total load used by all the followers of this leader and X represent the total cost of generating a unit of power (or its wholesale price). Therefore, the profit-maximization problem can be modeled as:

$$\max \left\{ \sum_{u \in U} BA_u - XM \right\} \tag{11}$$

where $M = \sum_{u \in U} \sum_{a \in A_u} z^h_{u,a}$ and $BA_u = BR_u + BN_u + BC_u$ It is important to note that every substation of an EB has a maximum load capacity installed, higher than which it cannot handle demand from customers at any point of time. Also, while calculating the profit, we have only considered the generating cost and selling price of electricity, not the other expenses that may be involved such as subsidiaries, taxes, distribution costs.

3.3 The Stackelberg Game

Since the Stackelberg game model used in this work is similar to that proposed by Meng F.L. and Zeng X.J., the proof of existence of Stackelberg strategy for this problem can be referred in [11]. We use a 1-leader and n-followers reverse Stackelberg game to solve our problem objectives. EB acts as the leader whose strategy, denoted as p_L, is the set of 24-hour prices for a day, and let P_L denote his strategy space. The followers' reactions are mutually exclusive. Let p_{F_u} and P_{F_u} denote the strategy and strategy space of follower F_u, where $u = 1, 2, \cdots, U$.

For each of the leader's strategy $p_L \in P_L$, the followers react by minimizing their objective function, independent of each other.

$$\min_{p_{F_u} \in P_{F_u}} O_{F_u}\left(P_L, R_{F_u}\right) = \min \sum_{a \in A_u} \delta_{1,u,a} \times BA_{u,a} + \delta_{2,u,a} \times D_{u,a}$$

This is the best response strategy $R_{F_n}(p_L)$, for $p_L \in P_L$, of the followers.

In turn, the leader maximizes his objective function by taking into account every follower's best response strategy.

$$\max_{p_L \in P_L} O_L\left(p_L, R_{F_1}(p_L), \ldots, R_{F_u}(p_L)\right) = \max\left\{\sum_{h=1}^{24} p^h \times \sum_{u \in U}\sum_{a \in A_u} z_{u,a}^h - XM\right\}$$

This way, we can obtain a Stackelberg strategy, using the most optimal solution p_L^* for the leader's profit-maximization problem and the follower's solution $p_{F_u}^*$ obtained using their best response strategy to p_L^*, i.e., $p_{F_u}^* = R_{F_u}\left(p_L^*\right)$, as $\left(p_L^*, p_{F_1}^*, p_{F_2}^*, \ldots, p_{F_u}^*\right)$

4 Methodology

In this work, we use genetic algorithm to solve the leader's problem and employ a greedy approach to that of the followers. Our system is driven by the genetic algorithm which also encapsulates the followers' problem.

4.1 Solving Leader's Problem

The profit-maximization problem for the leader has been implemented using genetic algorithm. Table 1 indicates the parameters set for our genetic algorithm. It is important to note that each gene i in the chromosomes is real-valued, not binary and represents the price (in Rupees) per unit of electricity consumed at the hour $i \in H$. The fitness function of every genetic algorithm is different and depends entirely on the problem being solved. In our system, the fitness of each chromosome is evaluted based on the profit it generates for the leader. This genetic algorithm employs

Table 1 Description of parameter values in genetic algorithm

Parameter	*Value*
Size of population	20
Length of each chromosome	24
Number of generations	500
Tournament size	2
Mutation rate	0.2 ($\approx$ 4 genes)

tournament selection, uniform crossover, and uniform mutation to perform selection, crossover, and mutation operations, respectively. Algorithm 1 explains how the leader's problem has been solved using genetic algorithm.

Algorithm 1: Profit-maximization algorithm using Stackelberg and evolutionary (GA) approach

Input: P_h^{min} and P_h^{max} vectors of length 24 each, provided by the government as the permitted price range for each hour.

Output: Profit gained

1 Initialize a population of N chromosomes randomly, of length 24 each. Each chromosome is a strategy of the leader. Every gene in the chromosomes should be within the range specified by P_h^{min} and P_h^{max}.

2 **for** $i = 1$ *to* N **do**

3 Leader plays his strategy $p_{L,i}$ by communicating the prices for the next 24 hours to the smart meters.

4 Every follower reacts to the leader's strategy with an optimal schedule of his appliances, i.e., response strategy $p_{F_u,i} = R_{F_u}(p_{L,i})$ Fitness of strategy $p_{L,i}$ is evaluated by solving (8).

5 Select parents for next generation using tournament selection.

6 Perform crossover operation on the parents using uniform crossover.

7 Mutate the child chromosomes obtained above using uniform mutation.

8 Repeat Steps 2–7 until a maximum number of generations are obtained.

4.2 Solving Followers' Problem

The users provide various inputs regarding their appliances to the smart meter which in turn follows Algorithm 2 to schedule the appliances and compute the minimized bill for the day. As seen in Sect. 3, each type of appliance is scheduled in a different manner. While non-reschedulable and curtailable appliances cannot be time-shifted, reschedulable appliances need to be shifted to a time period that gives as much satisfaction and minimization to the bill as required. For this process, we employ a greedy approach. Not much computation is required for non-reschedulable appliances as the power consumption is constant. Scheduling of curtailable appliances has been solved using Formula (7).

Algorithm 2: Scheduling algorithm for optimal usage of appliances

Input: For each user $u \in U$:
(i) The set of electricity prices for 24 hours, i.e., $P = [P_1, P_2, \cdots, P_{24}]$.
(ii) The total set of appliances A_u.
(iii) The preferred scheduling interval $[\alpha_{u,a}, \beta_{u,a}]$, for each $a \in A_u$.
(iv) The minimum and maximum power levels $\theta_{u,a}^{min}$ and $\theta_{u,a}^{max}$ for each $a \in A_u$.
(v) A satisfaction rate vector S, for each $r \in R_u \in A_u$.
(vi) Importance factor $\delta_{1,u,a}$ for minimized bill, for each $a \in A_u$.
Output:
(i) Sum of Bill amount of all users, BA.
(i) Total power consumed, $Z(24)$.

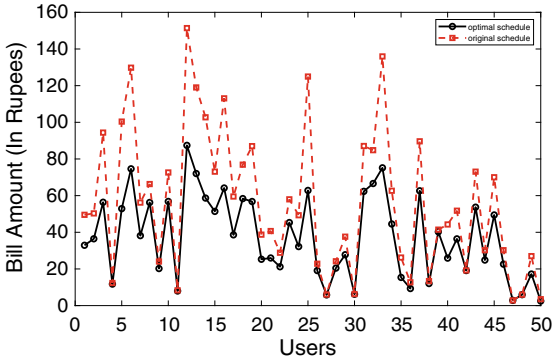
```
1  BA := 0, Z(24) := 0
2  for i = 1 to U do
3      BA_u := 0
4      for j = 1 to A_u do
5          if a_j ∈ N_u then
6              Generate Z_{u,n} using (4)
7              Calculate BN_u by applying Z_{u,n} in (5)
8              BA_u+ = BN_u, Z+ = Z_{u,n}
9          else if a_j ∈ C_u then
10             Generate Z_{u,c} using (7)
11             Calculate BC_u by applying Z_{u,c} in (8)
12             BA_u+ = BC_u, Z+ = Z_{u,c}
13         else
14             if δ_{1,u,a} > 1 − δ_{1,u,a} then
15                 Sort P based on price
16                 Minimize E as in (3)
17             else
18                 Sort P based on magnitude of difference between α_{u,a} and indices of P
19                 Minimize E as in (3)
20             BA_u+ = BR_u, where BR_u ⇒ min E
21             Z+ = Z_{u,c}
22     BA+ = BA_u
```

5 Results

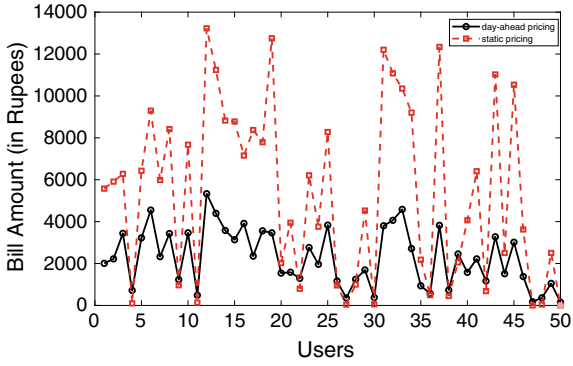
The electricity price limits in our model for each of 24 hours were chosen from value ranges in the existing electricity tariff of Tamil Nadu [16], but they were set differently according to peak and non-peak hours. About 25 different common appliances were accounted for, and the program was simulated among a locality of almost 1000 users.

Figures 2a and b are great indications as to how much money was saved by the followers using our scheduling model. Figure 2a compares the bill amounts to be paid by each of the first 50 users for their original and optimal schedules, both calculated using our day-ahead tariff, whereas Fig. 2b compares the total bill for two months using Tamil Nadu's original tariff and our day-ahead tariff, for the same 50 users. It

Fig. 2 Bill amounts



(a) Bill amounts of each user in a day

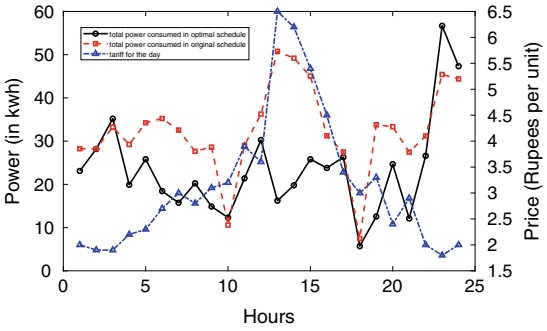


(b) Bimonthly Bill amounts of each user

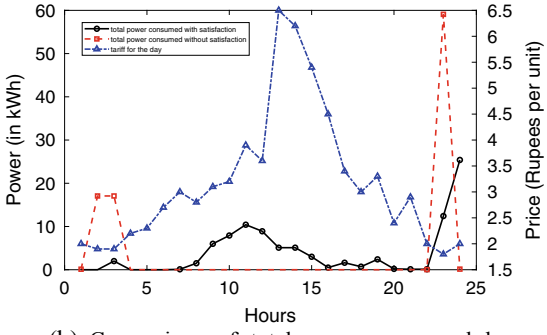
can be noticed in Fig. 2b that few users have lesser bills with the static pricing tariff than day-ahead pricing. This is because the first 100 units of electricity consumed is free for all domestic users in Tamil Nadu. However, the average and standard deviation of bimonthly bill amounts with day-ahead pricing are observed to be Rs. 2285 and Rs. 1377, respectively, which are much lower than those observed with static pricing (Rs. 5365 and Rs. 4206, respectively). This indicates a significant decrease (57%) in the average bimonthly bills with day-ahead tariff. A survey conducted in 2016 reveals that the average household electricity consumption per month in Tamil Nadu was 80–90 units [4], which in turn would add up to more than 100 units for two months. This number is bound to have increased significantly by 2021 due to increase in the number of electrified households and dependency on electrical devices. So, it is safe to conclude that majority of the population, which consumes more than 100 units of electricity in two months, will find our day-ahead electricity pricing model more beneficial than the existing static pricing model in Tamil Nadu.

Although higher prices in the peak hours have proven to shift the usage of reschedulable appliances significantly to non-peak hours in our model, the importance levels for minimized bill and satisfaction provided by the user also affect the

Fig. 3 Power consumption



(a) Comparison of total power consumed by Reschedulable & Curtailable appliances of all users in a day, before (original) and after (optimal) scheduling

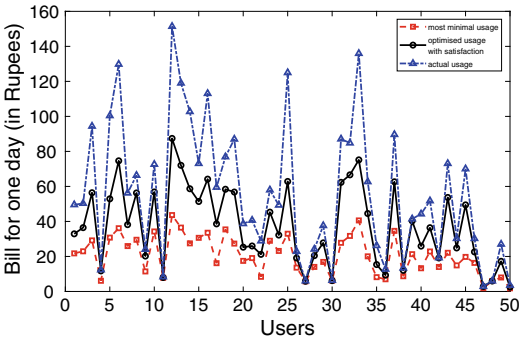


(b) Comparison of total power consumed by Reschedulable appliances of all users in a day, with and without satisfaction

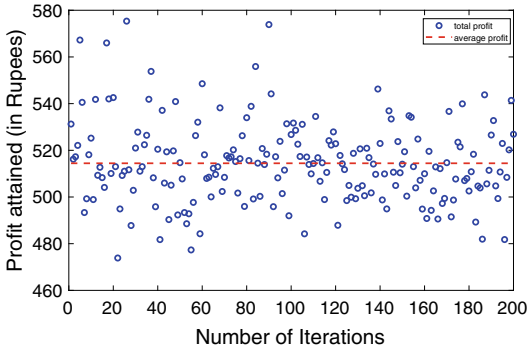
usage schedule computed for an appliance. This is evident from Fig. 3a, where originally more power was consumed in peak hours, but our model has reduced the power consumption during these hours drastically. This figure represents the total power consumed by reschedulable and curtailable appliances of all users in a day. It can be observed that the consumption of reschedulable appliances has been shifted to non-peak hours when the prices are comparatively lower. The energy consumption of curtailable appliances has been controlled depending on the price at that hour. Figure 3b compares the overall power consumption of reschedulable appliances in a day when and not user satisfaction is given importance. The difference in both will be more significant if the users provide higher values to satisfaction rate vector S and extreme values to $\delta_{1,u,r}$ and $\delta_{2,u,r}$. It can be observed that the reschedulable appliances have been completely shifted to non-peak hours when user satisfaction is not given importance. Although this is beneficial for DR and electricity conservation, it may not be very practical to implement.

Figure 4a depicts how different the electricity bill for a day is for most minimal usage, optimal usage, and original usage of the appliances, for the first 50 users.

Fig. 4 Efficiency of the algorithm



(a) Comparison of bill amounts of users with most minimal (without satisfaction), optimised (with satisfaction) and original (without scheduling) usage of appliances



(b) Results of leader's profit obtained over 200 runs of the proposed algorithm

Although least bill amounts are obtained with most minimal usage, this would mean compromising on the users' satisfaction, always running appliances on low power, and at impractical schedules. The highest bills are obtained when the appliances are operated at the user's original or most preferred schedules. It can be observed that there are few users whose bills amounts are nearly same in all three cases. This is because, these users do not use curtailable or reschedulable appliances significant enough to cause a difference. When the system is simulated 200 times with 100 generations each, it can be seen in 4b that the final result changes with every call, depicting the nature of evolutionary algorithms. It conveys a range of profit attained over several calls with an average of Rs. 514.4 in a day.

Table 2 describes the results obtained from common curtailable appliances in our model. Since air conditioners are present in majority of households, their contribution to the electricity provider's revenue is the highest among curtailable appliances. It is also important to note that air conditioners are mostly used during peak hours when the weather is sultry, especially in Tamil Nadu. This creates a need to manage the

Table 2 Curtailable appliances

	Air conditioner	*Space heater*	*Desktop*
Minimum power (kWh)	1	2	0.1
Maximum power (kWh)	4	5	0.45
Contribution to bills before scheduling (Rs.)	1507.2	217	72.855
Contribution to bills after scheduling (Rs.)	869.75	145.6	41.1
Typical usage period	10 P.M–12 A.M, 12 P.M–3 P.M	3 A.M–5 A.M.	Daytime

Table 3 Reschedulable appliances

Appliance	*Power rating (kWh)*	*Typical usage period*
Washing machine	0.5	Daytime
Vacuum cleaner	0.675	Daytime
Dish washer	1.5	Daytime
Iron box	1	8A.M–9A.M

use of air conditioners in peak hours, both for the electricity providers to balance the load and for the users to minimize their bills. It can be observed that the usage and contribution of air conditioners to the eclectricity bills have been reduced by almost half the original. Space heaters are not a common appliance among the citizens of Tamil Nadu, but it may be used in the winters. Although used in non-peak hours, it is important to use them responsibly as they are very high power-consuming. Table 3 lists few of the common reschedulable appliances used by the followers. Table 4 demonstrates the results obtained by using genetic algorithm for the profit-maximization problem.

Table 5 compares the bill amounts of the lowest and highest power-consuming users in different scenarios. It can be observed that the least power-consuming user who consumes less than 100 units receives a bimonthly bill of zero. This can be beneficial to such extremely low power-consuming users, because the first 100 units consumed in two months are free of cost in the static pricing model. Since the population of such users are significantly less in urban areas, RTP models could be overall more beneficial to both the users and the electricity providers.

6 Conclusion

Nowadays, there is a dire need in many countries including India to change the existing fixed-pricing electricity model to real-time pricing (RTP), facilitated by smart meters. Without RTP, customers will continue to use power during peak hours

Table 4 Genetic algorithm efficiency

	Profit in a day (Rupees)
Without genetic algorithm	481.6
With genetic algorithm	542.1
Minimum profit in 200 runs	473.9
Maximum profit in 200 runs	575.3
Average profit in 200 runs	514.4

Table 5 Efficiency of the proposed model

	Bill amount of the highest power-consuming user (Rs.)	*Bill amount of the lowest power-consuming user (above 100 units) (Rs.)*
333 without scheduling day-ahead pricing in one day	151.4	3.501
With scheduling day-ahead pricing in one day	87.42	2.494
Static pricing (bimonthly)	13230	0
Day-ahead pricing (bimonthly)	5332	152.1

which has various disadvantages as discussed in Sect. 1. Through our proposed model, the electricity department can identify tariff plans in which users can shift their usage to non-peak hours and also be satisfied with the schedule. With our reverse Stackelberg game-based technique, we have also noticed reductions in the bimonthly bills, thus paving way for the government to increase the existing price of electricity to compensate for losses. Also, by enabling shifting of domestic power consumptions significantly to non-peak hours, the electricity providers can efficiently manage the load demands at all times and also avoid plausible powercuts to anyone.

Our work in this paper can be extended further in various dimensions. Firstly, in this work, we have considered reschedulable and non-reschedulable appliances to operate over a constant power rating. This can be enhanced by applying the power-consumption logic of curtailable appliances to handle control tuning of the power usage of reschedulable and non-reschedulable appliances. Secondly, the set of peak and non-peak hours in our system is fixed, which in reality can change depending on seasons, weather conditions, lockdowns, holidays, etc. Therefore, our model can be made to dynamically modify electricity prices according on the time of the year or use machine learning to study and predict the electricity consumption of users. Thirdly, our proposed work may also be incorporated with power supplies from microgrid system establishments.

References

1. CEA, G.o.I.: Tariff & Duty of Electricity Supply in India (Mar 2014), http://www.cea.nic.in/reports/others/enc/fsa/tariff_2014.pdf
2. CEA, G.o.I.: Growth of Electricity Sector in India from 1947-2019 (May 2019), http://www.cea.nic.in/reports/others/planning/pdm/growth_2019.pdf
3. CEA, G.o.I.: All India Installed Capacity (in MW) of Power Stations (Jun 2020), http://www.cea.nic.in/reports/monthly/installedcapacity/2020/installed_capacity-06.pdf
4. Chunekar, A., Varshney, S., Dixit, S.: Residential electricity consumption in india: What do we know? Prayas (Energy Group), Pune **4** (2016)
5. Das, N.R., Rai, S.C., Nayak, A.: A Model for Optimizing Cost of Energy and Dissatisfaction for Household Consumers in Smart Home. In: Advances in Intelligent Computing and Communication, pp. 523–531. Springer (2020)
6. ETGovernment: Budget 2020: Govt to convert all electricity meters into smart prepaid meters by 2022. ETGoverment.com (Feb 2020), https://government.economictimes.indiatimes.com/news/smart-infra/budget-2020-govt-to-convert-all-electricity-meters-into-smart-prepaid-meters-by-2022/73866539
7. Groot, N., De Schutter, B., Hellendoorn, H.: Reverse stackelberg games, part i: Basic framework. In: 2012 IEEE International Conference on Control Applications. pp. 421–426. IEEE (2012). 10.1109/CCA.2012.6402334
8. Huang, Y.: Optimal Demand Response and Supply Schedule Under Market-Driven Price: A Stackelberg Dynamic Game Approach. arXiv preprint arXiv:1911.07365 (2019)
9. Kothari, S.: Power Grid Failure (Mar 2017), https://cdgi.edu.in/pdf/Power%20Grid%20Failure.pdf
10. Kumar, S., Sodha, N., Wadhwa, K.: Dynamic tariff structures for demand side management and demand response: An approach paper from India. ISGAN (2013)
11. Meng FL, Zeng XJ (2013) A Stackelberg game-theoretic approach to optimal real-time pricing for the smart grid. Soft Computing 17(12):2365–2380. https://doi.org/10.1007/s00500-013-1092-9
12. Padma, Priya, R., Rekha, D.: Sustainability modelling and green energy optimisation in microgrid powered distributed FogMicroDataCenters in rural area. Wireless Networks pp. 1–14 (2020). 10.1007/s11276-019-02207-z
13. Rao, M.R., Kuri, J., Prabhakar, T.V.: Towards optimal load management with day ahead pricing. In: 2015 7th International Conference on Communication Systems and Networks (COMSNETS). pp. 1–8. IEEE (2015)
14. Rogers, J., Suphachasalai, S.: Residential consumption of electricity in India: documentation of data and methodology. The World Bank (2008)
15. Statistics, E.: Central statistics office, Ministry of Statistics and Programme Implementation. Govt. of India, New Delhi (26th Issue)(www. mospi. gov. in) (2019)
16. TANGEDCO: One Page Statement on Tariff Rates as in the TNERC (Aug 2017), https://www.tangedco.gov.in/linkpdf/ONE_PAGE_STATEMENT.pdf
17. Tripathi, B.: Now, India is the third largest electricity producer ahead of Russia, Japan. Business Standard India (Mar 2018), https://www.business-standard.com/article/economy-policy/now-india-is-the-third-largest-electricity-producer-ahead-of-russia-japan-118032600086_1.html
18. Yang P, Tang G, Nehorai A (2012) A game-theoretic approach for optimal time-of-use electricity pricing. IEEE Transactions on Power Systems 28(2):884–892. https://doi.org/10.1109/TPWRS.2012.2207134

Unsupervised Spectral Feature Extraction for Hyperspectral Image Classification



Ratna Sravya Sajja and B. L. N. Phaneendra Kumar

Abstract The modern criterion of remote sensing is the acquisition of enormous dimensions of hyperspectral images. The transformation of such enormous multi-dimensional images, object unmasking, feature drawing, forecasting, and hyperspectral image classification is several progressive affairs in the present scenario. Countless mechanisms have been refined to classify hyperspectral images. Hyperspectral image (HSI) classification has been a very dynamic area in remote sensing and other applications like agriculture, eye care, food processing, mineralogy, surveillance, etc. HSI gathers and summons information cross the electromagnetic spectrum. The objective of HSI is to earn the spectrum for all pixels in the image for the reason of discovering things, analyzing materials, and recognizing processes. It includes varying bands of images. HSI often dispenses with an inseparably nonlinear connection linking the recorded spectral data and the similar materials. Recently, deep learning antiquated as a robust feature extraction tool to effectively disclose irregular problems and extensively utilized in several image processing tasks. In the beginning, fast independent component analysis (FICA) is applied for dimensionality reduction. Convolutional neural network (CNN) is the persistently used deep learning-based technique for observable data processing. Initially, fast independent component analysis is executed to reduce dimensions and then CNN is implemented to the reduced data. The results are tested using different spectral features with CNN for HSI. The overall accuracy of the suggested approach is 99.83 and 100% for Indian Pines, Salinas Scene which shows the reliability of the suggested CNN method for HSI classification.

Keywords Classification · Hyperspectral · Convolutional neural networks · Fast-independent component analysis

R. S. Sajja (✉) · B. L. N. Phaneendra Kumar
Department of Information Technology, V R Siddhartha Engineering College, Vijayawada, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_23

1 Introduction

HSI is the combination of conventional imaging, spectroscopy, and radiometry. It creates images where the spectral signature is grouped with each pixel. The datasets are data cubes that are three-dimensional where spatial data is collected in X–Y Plane, and spectral data is presented in the Z-direction. The spectral resolution is termed as the interval between different wavelengths measured in a specific wavelength range [1]. The exploration of HSI is supreme as it produces multiple bands that affect analysis due to enlarged volume. The data comprise spectral along with spatial information of contrasting bands. HSI classification is a surprising mechanism to examine diversified land cover in remote sensed hyperspectral images. HSI classification is utilized to delegate a class label to every pixel. The investigation of HSI is predominant due to its potential applications. In agriculture, HSI is used to obtain the spectral and spatial information from land, and these are helpful in an extensive variety of fields to inspect the issues [2] and suggest a mode for finding exceptional results in practice such as Dale et al. [3] have reviewed a technique to characterize soil based on the properties like moisture, organic matter, and salinity. Zhang [4] suggested a technique for the extraction of minerals like copper, silver, gold, and lithium and to classify them. The HSI also has medical applications Calin [5] has reviewed tumor delimitation and recognition, estimating tissue perfusion and its pathological circumstances, hyperspectral imaging gives accurate results. Feng [6] presented an approach for food safety. HSI has been fortunately utilized in food safety inspection and control. Negi [7] extended a procedure for collecting spectral signature for different items and other environmental things [8] fungal infection [9] wetland classification [10] forestry applications, etc.

Numerous hyperspectral image classification mechanisms have been developed. Akbari [11] employed a hierarchical segmentation-based technique for spatial extraction and multilayer perceptron neural network, as it has inflated smoothness of the image accuracy is dropped. Binge [12] suggested a segmentation algorithm and classification with SVM. Yang et al. have presented a mechanism based on hyperspectral image classification with deep learning models [13]. Here the authors have used 2D CNN, 3D CNN, R-2D-CNN, and R-3D-CNN to obtain patch extraction and label identification. Roy has proposed an algorithm based on three and two-dimensional CNN features [14]. At first, the authors implemented principal component analysis (PCA) to acquire the neighboring patches and to minimize the dimensionality. Lin has proposed a method based on spatial as well as spectral band reduction and CNN [15]. Here, the authors used CNN, pseudo-synthesis, Shannon entropy for information measure. If the information is scattered, then the considered three spectra may not have the important information. Ahmad has proposed a technique based on a fast 3D CNN for hyperspectral image classification [16] using incremental PCA and CNN without data augmentation and batch normalization. Yu has presented a framework for CNN where both spectral–spatial features are included [17], and the auto-encoder is implemented for dimensionality reduction. Today, researchers are concentrating on the categorizing hyperspectral images upon deep neural networks

[18]. Vaddi has presented a mechanism based on HSI classification using spectral and spatial features integration in CNN [19], Initially, probabilistic principal component analysis is performed preceded by the Gabor filter, and the derived images are integrated to train the convolution neural network (CNN) with 2D convolution layer for the accomplishment of feature maps. Alkha Mohan has introduced a method based on hybrid CNN-based HSI classification using multiscale spatial-spectral features [20]. To utilize the spectral information, the authors have implemented kernel principal component analysis. Makantasis has proposed a mechanism based on deep supervised learning [21]. A Gabor feature embedded CNN is trained in [22], and a whale optimization-based method is implemented in [23].

2 Proposed Approach

The design of the suggested method is shown in the figure. The fast-independent component analysis (FICA), which is an unsupervised learning algorithm, is used initially for dimensionality reduction, and then the obtained image patches are given to CNN to produce the classification map.

Let us take the input hyperspectral cube which is represented in three-dimensional space with proportions as RXCXB where *R* represents the rows, *C* represents the columns, and *B* represents the spectral bands. After reading the input, hyperspectral cube dimensionality reduction technique is applied. The dimensionality reduction technique which is proposed is fast ICA. The main purpose of the independent component analysis is to locate a straight presentation of non-Gaussian data, so that, the elements are statistically independent. The traits of fast ICA are parallel, computationally effortless, and it takes little memory. The fast ICA looks for directly independent components which have non-Gaussian distribution and uses nonlinearity for optimizing the solution. The independent components will be generated without any loss of data. The dimensions of the spectral bands are reduced to 20 without any loss of spatial dimensions. The HSI data cube is partitioned into small 3D patches with WXW as the window size and *B* as the spectral bands. The schematic representation of the proposed method is expressed in Fig. 1.

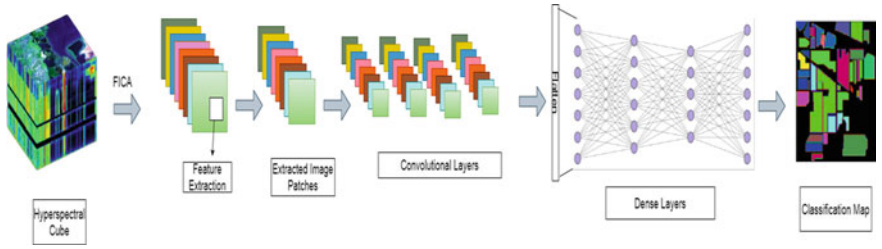


Fig. 1 Proposed architecture of CNN

2.1 Fast-Independent Component Analysis (FICA)

Independent component analysis (ICA) is a statistical technique that constitutes a multi-dimensional linear vector which is expressed as a linear mixture of non-Gaussian random variables that are independent. As suggested in [24], the estimation of ICA can be done by applying the Fast ICA algorithm, which is computationally powerful. It utilizes a fixed point echoing scheme which is quicker than the gradient descent methods. It can perform projection pursuit, which is a technique used for exploratory data analysis for visualization and can also be used to approximate the independent components. Likewise, it can find low-dimensional projections to show highly non-Gaussian distributions. There is an inbuilt package named fast ICA to reduce the dimensionality of the given input and produces the low dimension data with independent components. The procedure of feature extraction is as follows:

Step-1: Initially to perform feature extraction, the FICA method is defined which has the number of the arguments, components, and the input data.
Step-2: The input data is reshaped, and then the method is applied to the reshaped data, and it is transformed.

The obtained new shape is split into overlapping 3D patches.

2.2 Convolutional Neural Network (CNN)

The framework of the suggested CNN is depicted in Fig. 2. Initially, the hyperspectral dataset is loaded into memory. Firstly, the FICA is applied to the dataset to obtain the independent components and then image patches are extracted from the dataset and the obtained independent components are given to CNN as input.

The CNN architecture comprises convolutional layers accompanied by fully connected layers. The convolutional layers are three-dimensional layers with kernels of three dimensions height, width, and depth. The CNN consists of an input layer with 25×25 as window size and $B = 20$, which are the spectral bands. The first layer of 3D CNN is added with a filter size of 8 and the output shape is $23 \times 23 \times 14 \times 8$. The second layer of 3D is added with a filter size of 16 followed by the third and fourth layers with filter sizes like 32, 64, respectively. After that, a few flatten layers are added to the model. Four dense layers with different units are added followed by the dropout layers with different units. These layers are added with nonlinear activation function as rectified linear unit (ReLU). At the last layer, i.e., output layer, the softmax activation function is applied. This function has the probability scores for the classification of multi-class hyperspectral image classification.

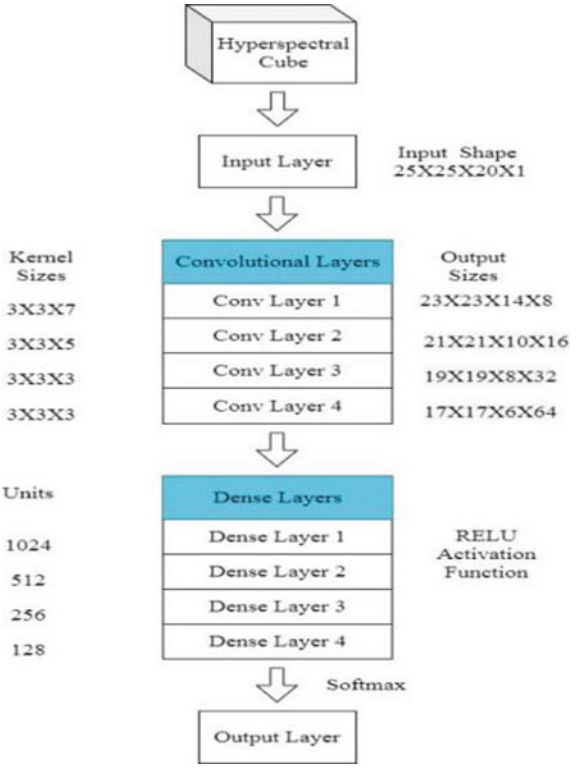


Fig. 2 Recommended 3D CNN architecture

3 Datasets and Experimental Setup

The achievement of the designed architecture is tested on two standard datasets. Indian Pines as well as Salinas Scene. Indian Pines (IP) was collected over the test site in northwestern Indiana. The ground truth image is shown in Fig. 3a.

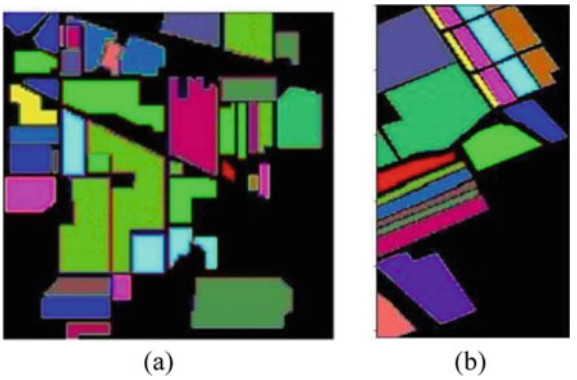


Fig. 3 **a** Indian Pines dataset **b** Salinas dataset

Salinas Scene (SA) were collected over Salinas Valley, California, using an AVIRIS sensor. Its ground truth image is represented in Fig. 3b.

4 Results and Observations

4.1 Classification Maps

The ground truth image of Indian Pines and the predicted output images are shown in Fig. 4. It is clear that the predicted image has a loss at the corn-notill class and soybean notill class.

The ground truth image of Salinas Scene and the predicted output images are shown in Fig. 5. It is clear that the predicted image is more accurate.

From the figures, it is noticeable that the classification maps of the advised methods clearly show the proper attribution of a class label to associating labels. The achievement of the considered framework concerning individual classes is represented in the classification maps for the two datasets. Table 1 represents the training and test samples of the standard datasets used in the proposed method.

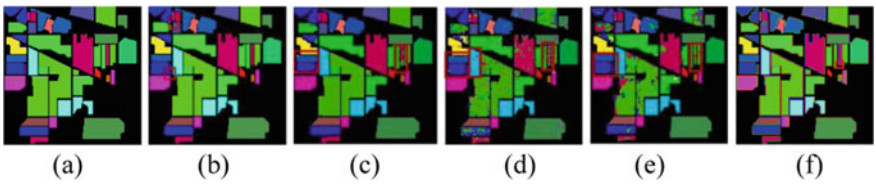


Fig. 4 **a** Ground truth-Indian Pines **b** Hybrid SN **c** GFDN **d** SVM **e** CNN **f** proposed

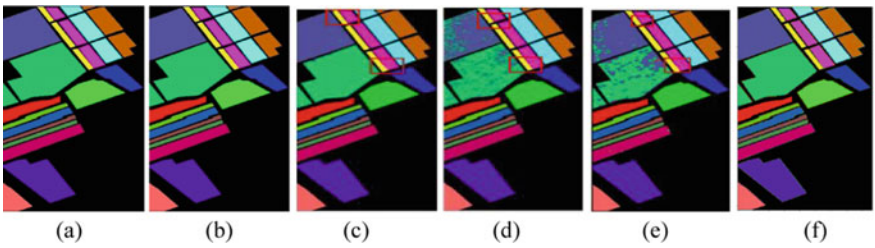


Fig. 5 **a** Ground truth-Salinas **b** Hybrid SN **c** GFDN **d** SVM **e** CNN **f** proposed

Table 1 Training and test samples of the Indian Pines and Salinas Scene

Dataset	% of training samples	% of test samples
Indian Pines	30	70
Salinas	30	70

Table 2 OA, AA, Kappa coefficient values using the proposed method

Datasets	Overall accuracy	Average accuracy	Kappa
Indian Pines	99.83	99.90	99.80
Salinas Scene	100.00	100.00	100.00

4.2 Comparison of Performance Measures

The proposed method uses three accuracy measures, overall accuracy (OA), which depicts the correctly classified samples out of the total test samples. The second metric is average accuracy (AA), which depicts the average of the classwise classification accuracies. The third metric is Kappa coefficient (Kappa), which depicts the statistical measurement for mutual information between the ground truth map and classification map. Table 2 exhibits the performance metrics OA, AA, Kappa coefficient of the proposed method.

The three metrics are evaluated for five mechanisms under deliberation. These values are represented as bar graphs in Fig. 6a for Indian Pines. The CNN is trained

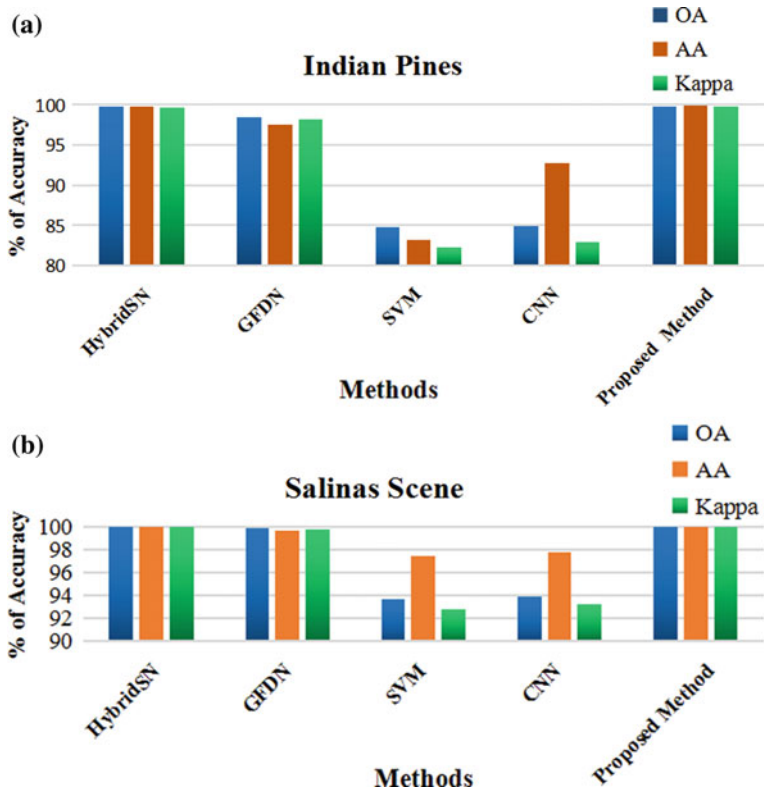


Fig. 6 Performance metrics **a** Indian Pines dataset **b** Salinas

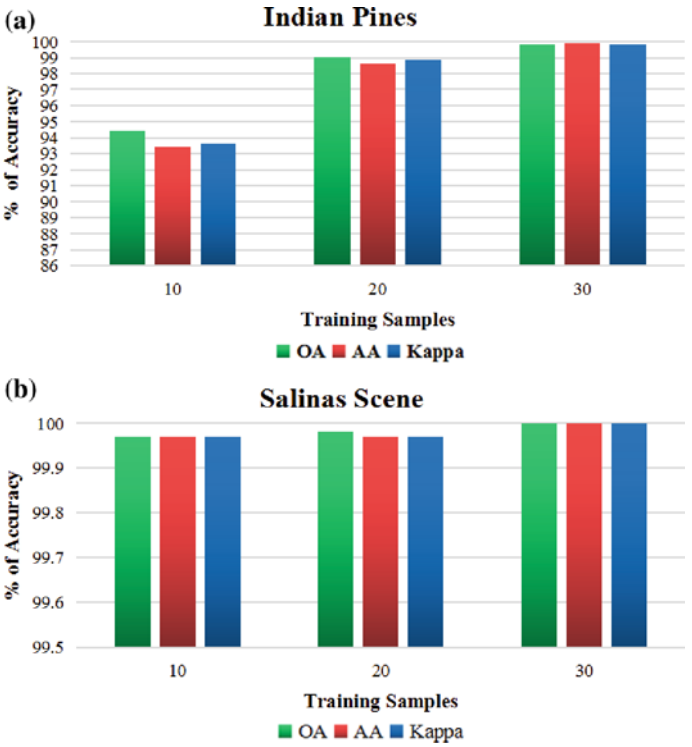


Fig. 7 Accuracy for different training samples **a** Indian Pines **b** Salinas

with distinct % of training samples. The accomplishment with the distinct% of the training samples on the Indian Pines is depicted in Fig. 7a. The proposed algorithm achieved good accuracy at 30%.

The performance metric values are represented in Fig. 6b for the Salinas Scene. Figure 7b shows Salinas dataset achievement with the different % of samples.

5 Conclusion

The selection of indicative bands from a specified hyperspectral image is one of the leading tasks in the HSI classification. Here, the indicative features are extracted from one dimensionality reduction technique, viz. the fast independent component analysis. Then trained the algorithms using the convolution neural network. In the framework, four convolutional layers are implemented accompanied by four dense layers with distinct kernels and filters. For the last layer, a softmax activation/classifier is utilized for categorizing the pixels. The overall, average, and Kappa accuracies on Indian Pines are 99.83%, 99.90%, and 99.80%; on Salinas are 100.0%, 100.0%,

and 100.0% correspondingly. The results exhibit the reliability of the prescribed technique and surpass other state-of-the-art approaches.

References

1. Boggavarapu LNPK, Manoharan P (2020) Hyperspectral image classification using fuzzy-embedded hyperbolic sigmoid nonlinear principal component and weighted least squares approach. J Appl Remote Sens 14(2):024501 (2 April 2020). https://doi.org/10.1117/1.JRS.14.024501
2. Hughes GF (1968) On the mean accuracy of statistical pattern recognizers. IEEE Trans Inf Theor 14:55–63
3. Dale L, Thewis A, Boudry C, Rotar I, Dardenne P, Baeten V, Fernández Pierna J (2013) Hyperspectral imaging applications in agriculture and agro-food product quality and safety control: a review. Appl Spectrosc Rev 48:142. https://doi.org/10.1080/05704928.2012.705800
4. Zhang T, Liu F (2012) Application of hyperspectral remote sensing in mineral identification and mapping. In: Proceedings of 2nd international conference on computer science and network technology, ICCSNT 2012, pp 103–106. https://doi.org/10.1109/ICCSNT.2012.6525900
5. Calin M, Parasca S, Savastru D, Manea D (2014) Hyperspectral imaging in the medical field: present and future. Appl Spectrosc Rev 49. https://doi.org/10.1080/05704928.2013.838678
6. Feng Y, Sun D-W (2012) Application of hyperspectral imaging in food safety inspection and control: a review. Crit Rev Food Sci Nutr 52:1039–1058. https://doi.org/10.1080/10408398.2011.651542
7. Negi H, Shekhar C, Singh S (2015) Snow and glacier investigations using hyperspectral data in the Himalaya. Curr Sci 108:892–902
8. Olmanson LG, Brezonik PL, Bauer ME (2013) Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: the Mississippi river and its tributaries in Minnesota. Remote Sens Environ 130:254–265
9. Lu Y, Wang W, Huang M, Ni X, Chu X, Li C (2020) Evaluation and classification of five cereal fungi on culture medium using Visible/Near-Infrared (Vis/NIR) hyperspectral imaging. Infrared Phys Technol 105:103206. ISSN 1350-4495
10. Samiappan S et al (2017) Classifying common wetland plants using hyperspectral data to identify optimal spectral bands for species mapping using a small unmanned aerial systems—a case study. In: 2017 IEEE international geoscience and remote sensing symposium (IGARSS), Fort Worth, TX, pp 5924–5927
11. Zarco-Tejada PJ, Berjón A, López-Lozano R, Miller JR, Martín P, Cachorro V, González MR, de Frutos A (2005) Assessing vineyard condition with hyperspectral indices: leaf and canopy reflectance simulation in a row-structured discontinuous canopy. Remote Sens Environ 99:271–287
12. Akbari D (2019) Spectral-spatial classification of hyperspectral imagery using a hybrid framework. In: ISPRS—international archives of the photogrammetry, remote sensing and spatial information sciences. XLII-3/W6, pp 541–544. https://doi.org/10.5194/isprs-archives-XLII-3-W6-541-2019
13. Binge C, Ma X, Xie X, Ren G, Ma Y (2017) Classification of visible and infrared hyperspectral images based on image segmentation and edge-preserving filtering. Infrared Phys Technol 81:79–88, ISSN 1350-4495. https://doi.org/10.1016/j.infrared.2016.12.010
14. Yang X, Ye Y, Li X, Lau RYK, Zhang X, Huang X (2018) Hyperspectral image classification with deep learning models. IEEE Trans Geosci Remote Sens 56(9):5408–5423. https://doi.org/10.1109/TGRS.2018.2815613
15. Roy SK, Krishna G, Dubey SR, Chaudhuri BB (2020) HybridSN: exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci Remote Sens Lett 17(2):277–281. https://doi.org/10.1109/LGRS.2019.2918719

16. Lin L, Chen C, Xu T (2020) Spatial-spectral hyperspectral image classification based on information measurement and CNN. J Wireless Com Netw 59
17. Ahmad M (2020) A fast 3D CNN for hyperspectral image classification. arXiv:2004.14152 [eess.IV] (or arXiv:2004.14152v1 [eess.IV]
18. Yu C, Han R, Song M, Liu C, Chang C (2020) A simplified 2D–3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion. IEEE J Sel Top Appl Earth Observations Remote Sens 13:2485–2501. https://doi.org/10.1109/JSTARS.2020.2983224
19. Vaddi R, Manoharan P (2020) CNN based hyperspectral image classification using unsupervised band selection and structure-preserving spatial features. Infrared Phys Technol 103457. https://doi.org/10.1016/j.infrared.2020.103457
20. Vaddi R, Manoharan P (2020) Hyperspectral image classification using CNN with spectral and spatial features integration. Infrared Phys Technol 107. https://doi.org/10.1016/j.infrared.2020.103296. ISSN 1350-4495
21. Mohan A, Venkatesan M (2020) HybridCNN based hyperspectral image classification using multiscale spatiospectral features. Infrared Phys Technol. https://doi.org/10.1016/j.infrared.2020.103326ISSN1350-4495
22. Makantasis K, Karantzalos K, Doulamis A, Doulamis N (2015) Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: 2015 IEEE international geoscience and remote sensing symposium (IGARSS). Milan, pp 4959–4962
23. Boggavarapu LNPK, Manoharan P (2020) A new framework for hyperspectral image classification using Gabor embedded patch based convolution neural network. Infrared Phys Technol 110(103455):1350–4495. https://doi.org/10.1016/j.infrared.2020.103455
24. Boggavarapu LNPK, Manoharan P (2021) Whale optimization-based band selection technique for hyperspectral image classification. Int J Remote Sens 42(13):5105–5143. https://doi.org/10.1080/01431161.2021.1906979

Distributed Ledger Framework for an Adaptive University Management System



Yaswanth Lingam Sivadanam, R. Ramaguru, and M. Sethumadhavan

Abstract The true power and the state of any technology are best known during difficult times. The yet-to-over COVID-19 lockdown has unleashed the power of online and remote modes of learning, evaluation, and working. The universities and educational institutions worldwide faced their big-time challenge in delivering, managing, evaluating their student's learning progress. The need for an online-virtual-remote learning platform has become inevitable, which comes with a multitude of challenges, including usability, security, privacy, and real-time adoption. In this paper, we propose an Adaptive University Management System Framework (UMS) based on Distributed Ledger Technology (DLT) like Ethereum Blockchain where we record the schedules, the delivered classes, the materials and track the student's learning progress with the use of ERC-721 standard to tokenize the submissions & evaluations. To provide efficient and effective utilization, the learning materials and submissions made onto the system are stored in Distributed File Storage system - InterPlanetary File System (IPFS) and hosted in InterPlanetary Name System (IPNS) with files securely split using a secret sharing algorithm.

Keywords University management system · Blockchain technology · Ethereum · ERC-721 · IPFS · Secret sharing · Color tokens

Y. L. Sivadanam (✉) · R. Ramaguru · M. Sethumadhavan
TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: cb.en.p2cys19017@cb.students.amrita.edu

R. Ramaguru
e-mail: r_ramaguru@cb.amrita.edu

M. Sethumadhavan
e-mail: m_sethu@cb.amrita.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_24

1 Introduction

During the COVID-19 pandemic, the entire world was held on lockdown, and most operations were required to switch to online and remote mode. UNESCO's observation shows that "Approximately 60% of the world's student population is affected by these countrywide closures". Overall, close to 200 countries closed their schools, thereby more than 1.5 billion young students education has been interrupted. Gender, disability, immigration, mother tongue, learning challenges, socioeconomic factors contribute to widespread educational inequities during the pandemic. In recent months, approximately 40% of the world's poorest countries have been unable to support their deprived learners, with the unintended impact of school closures being particularly severe for deprived children and their families, as well as all learners with learning difficulties and special needs. Currently, about 320 million students in Indian schools and universities are affected [1]. Every single country and industry, organization, institution tried their level best to adopt the available technologies to continue its operation; Especially educational institutions started delivering classes and conduct examinations through virtual mode. Though the online, remote and virtual mode of working and learning was not completely new for certain countries and segments of participants but for many, this was the new normal. Students adopted independent learning, which helps them to get state-of-the-art methodologies and experiencing the new way of study even in a critical situation. Most of the students and teachers across the globe are habituated to offline modes of teaching and learning for years. The biggest hurdle and challenge many faces is the lack of the necessary tools & technologies and the technical knowledge on how to use them. Moreover, practical learning like using machinery and equipment is not possible in an online mode completely as the simulators for practical learning are still developing.

A File Management System (FMS) is administering the system & files like images, videos, documents, and process involved in storing, managing, and efficient retrieval. In the absence of an FMS, it is arduous to manage a large volume of data. The integrity of the file is not guaranteed due to the chances of duplication and easy manipulation in case of security miss-configuration. Besides, recovering lost data is quite arduous as it is a single system. Distributed File Management System (DFMS) deals with distributing the files across multiple nodes. In DFMS, there are two main components, namely location transparency and redundancy. The challenge here is access privilege and duplication of data. Data retrieval takes time even though the data is small as it is distributed over the network, which is not linked in any specific structure.

Amrita University Management System (AUMS) is an innovative & comprehensive system for effective management, learning management services, assessment management services, and understanding administration services that practically cover the entire area of university operations. It has high scalability and performance at the same time, ensuring it meets all the needs of an educational institution [2]. University Management System (UMS), which covers different operations of the university with high scalability and performance, is necessary for the days to come. In the traditional UMS managing large data requires huge storage as well as

computation power that too restricted to a single system. Since it is regulated by a central authority, there is a high risk of data theft, and it necessarily requires other high-end security protocols and interventions to ensure its safe operation.

Blockchain Ledger is a data structure identical to a linked list, with the exception that each block is cryptographically linked to its next block. Blockchain Technology is a decentralized computing and distributed ledger platform that uses a logical decision-making process among several participants in an open and public network to store transactions immutably in a verifiable manner, which supports private network configuration as well [3]. Blockchain supports greater transparency, enhanced security, traceability, auditability, and moreover, it provides immutability of the transaction data. Blockchain Technology which is the underlying technology of Bitcoin and other cryptocurrencies has far more applications to larger domains. Blockchain Technology is used in Automotive [4], Waste Management [5], Digital Rights Management [6], Supply Chain & Logistics monitoring, e-voting, Identity Management and much more. Blockchain technology could be the best choice for a system that requires high scalability and efficient management. University Management System (UMS), which requires transparency, auditability, traceability with efficient storage management and retrieval, can be implemented with Blockchain Technology and distributed storage platforms like IPFS.

The remaining portion of the paper is arranged as follows: Section 2 describes the related works in Blockchain with IPFS. Section 3 proposes the distributed ledger framework for university management, Sect. 4 concluding the paper. Section 5 highlights the scope for future work.

2 Related Work

In this section, we briefly look into various Blockchain and IPFS based educational file management platforms. In the last few years, multiple researchers worked and contributed to this area.

A credit-based private blockchain is hosted in a Public-Private Partnership (PPP) model to securely store and authenticate educational documents. This Blockchain uses Proof of Stake (PoS) consensus and a private IPFS to store the documents over the Ethereum Blockchain where only the private vendors are allowed to take part in the consensus. The system allows universities to register and issue mark sheets that are notified to the registered students. The credits are issued on a prepaid basis and can be purchased like cryptocurrencies once exhausted [7].

Open Educational Resources (OER) based on Blockchain and IPFS [8] is believed to address the drawbacks in the traditional teaching-learning process. Ethereum blockchain and IPFS are used for adequate storage and retrieval of Educational Resources. This solution addresses the attribution, copyright, and update issues related to the OER materials.

This paperwork [9] addressed the impact caused by COVID-19 in the education system and parallelly discussed various reports that reveal the impact of COVID-19

on E-learning, current solution security issues, and many more. This work provided an E-learning solution with a 5-layered framework based on Blockchain. By comparing other available sources during COVID-19, this work tried to adopt Blockchain technology to E-learning with the help of cloud computing that will make the solution expensive.

Massive open online courses (MOOCs) are gaining excessive attention as the education system is trying to adopt the E-learning process to provide and explore the out-of-classroom concepts. But the concern of this type of system with security-related issues like privacy and data manipulation is dragging the users to continue with existing techniques. This paper [10] believes that blockchain technology can help solve the problems confronting traditional methods by storing the student's data, time of creation, and other results in Blockchain. Since the Blockchain is a distributed ledger, the data present in it can be verifiable publicly from any place to avoid fraud and fake testimonials.

3 Proposed Work

This section proposes a distributed ledger framework for an Adaptive University Management System (UMS) using IPFS, secured by secret sharing and color tokens created logically at the front-end that deals with standard ERC tokens. Any university shall adopt the proposed framework to manage its online-virtual-remote teaching-learning process and evaluate its effectiveness.

3.1 System Architecture

The proposed system architecture as seen in Fig. 1 shows the various entities and components.

Web API is a browser-provided API that is used to communicate with the browser to solve front-end requirements. Using these APIs, anyone can easily build a user-friendly and attractive front-end for the applications deployed on the blockchain. With the help of Web3.js, smart contracts deployed on the blockchain can be conveniently managed and integrated with the front-end.

Secret Sharing Module splits the files like Open Educational Resource (OER), Question Papers, Assignments, Answer Sheets, Lab Reports, Thesis, which are upload to the system. We are using the Non-Linear Secret Sharing (NLSS) scheme (1,2,5) that splits any given file into '5' shares, where '2' shares are required to get back the original file with one essential share (referred to as secret) held by the owner of the file. These files (shares) are stored in IPFS [11].

Inter Planetary File System (IPFS) is a hypermedia protocol and distributed network for storing, sharing, and accessing files, websites, application data from a dis-

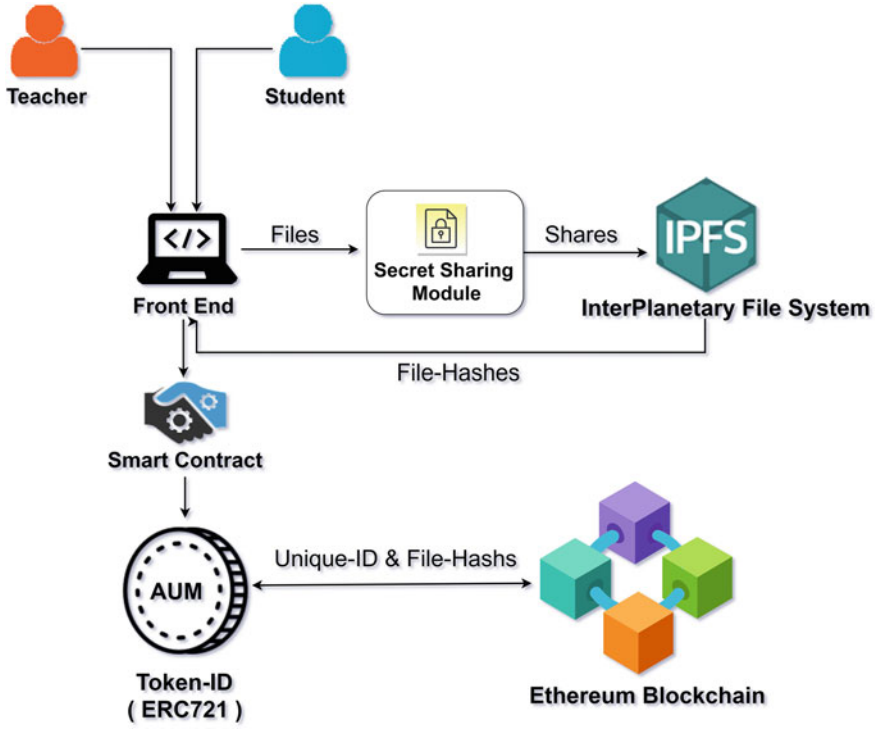


Fig. 1 System architecture

tributed file system. IPFS is a content-addressed network used to store data between peer-to-peer (p2p), so its content is accessible through peers, anywhere, anytime across the globe. Anyone with the hash value will be able to access the file [12]. In the proposed framework, we are using Private IPFS accessible only by the nodes running within the university campus.

InterPlanetary Name System (IPNS) is a naming system that is used to host the contents stored in the IPFS concerning the user's node ID [13]. We have used IPNS to host the web-based exams through remote-virtual mode. The files are added to the IPFS and then published to IPNS.

Metamask is the bridge between the traditional browser to the decentralized Smart Contracts deployed and running on Ethereum. Metamask also acts as a wallet through which the transaction is signed, and transaction fees are paid to the underlying Ethereum network [14].

ERC-721 is an Ethereum Standard for Non-Fungible Tokens (NFTs) [15]. NFTs could be a diverse universe of assets like Physical property, Virtual collectibles, "Negative value" assets. NFTs are distinguishable, and the ownership of each one is tracked separately. In the proposed framework, the files created by the teacher

and submissions made by the students during the stipulated time of the course are tokenized as the ERC-721 tokens. In our system, we've also incorporated bitcoin's definition of colour tokens [16]

ERC-725 which is an Ethereum Standard for Identity Management. ERC-725 allows for self-sovereign identity (SSI), where the users are allowed to manage their own identities. The teachers, students and heads of departments are identified in the blockchain through a SSI and they will be able to take their identity across different DApps.

Smart Contract is a self-executing digital agreement between two or more parties written and executed on a blockchain which eliminates the need for any intermediaries or third parties. Transactions of a smart contract are easily traceable and irreversible. Smart contracts are executed in EVM and requires fee to be paid in the units of gas.

Ethereum is a smart contract based blockchain platform that allows end-user to build distributed applications. The native cryptocurrency of Ethereum is Ether. Ethereum uses sandboxed virtual machines to store and execute its transactions and smart contracts. Ethereum has Mainnet and multiple Testnets available. We have deployed our proposed framework in the Ropsten Test network [17].

3.2 Usecases

File management in educational institutions is a preeminent task, and it deals with every aspect of the institution. File management covers various files from technical documentation to learning materials, official announcements to open resources, assignment submissions to exams. We have considered the continuous assessment and end-semester examination as the use-case to showcase the proposed framework's functionality.

3.2.1 Continuous Assessment Mode (CAM)

Continuous Assessment Mode (CAM) is widely adopted learning management model to evaluate the progress of the student continuously during the course. Adapting distributed technology to this methodology will bring state-of-the-art outcome that prevent cyber attacks.

Teacher Assignment Creation

Teacher or Faculty is the in-charge of a course taught during the given semester. The Teacher's responsibility, as known, is to deliver the lectures, provide assignments,

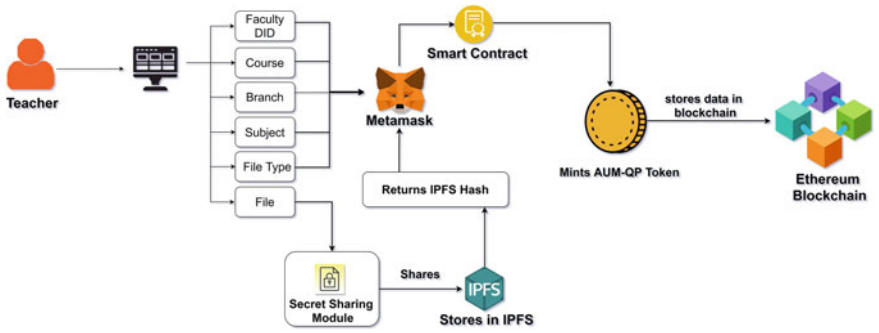


Fig. 2 Assignment by teacher and creation of AUM-QP token

lab exercises and evaluate the student's learning progress. Each Teacher or faculty is identified on the blockchain through a Decentralized Identifier (DID) created using ERC-725. The teacher can upload any assignment or periodical exam question paper through the dedicated user interface provided by the UMS. This uploaded file is securely split into multiple shares using the secret-sharing module, and the shares are then stored in the IPFS. These shares are referenced in the blockchain using the IPFS-hash value. The smart contract generates a *File-ID*, which is used to tokenize this assignment using ERC-721 standard represented as AUM-QP (Gold Token). The File-ID generated will be the ERC-721 Token ID referred to as ***AUM-QP Token-ID***. The smart contract automatically notifies the student about the assignment along with *File-ID*. The workflow and the parameters are shown in the Fig. 2. The smart contract automatically monitors the deadline of the assignment submission.

Student Assignment Submission

Students are classified using an ERC-725-based DID in the same way as teachers. A dedicated user interface provided to students allows them to view their learning progress, assignment notifications, and their submissions. Once the teacher uploads and assign anything against a particular class or student immediately it is notified via communication channels supported by the university (SMS, e-Mail, Student Portal). The student access the assignment through the ***AUM-QP Token-ID***. The student must complete the task and submit the answer sheet by the deadline. This file is also split into several shares and saved in IPFS after submission using the secret-sharing module. The smart contract generates a *File-ID* which is the Token-ID for the answer sheet, which is represented as ***AUM-AS Token-ID*** (Blue Token). The smart contract automatically notifies the teacher via mail, sms or any other notification medium that is being used in the campus. The workflow and the parameters are shown in the Fig. 3.

Figure 4 shows the detailed interaction between the student and the teacher and the head of the department in case of any mark sheets are to be an issue for the exams.

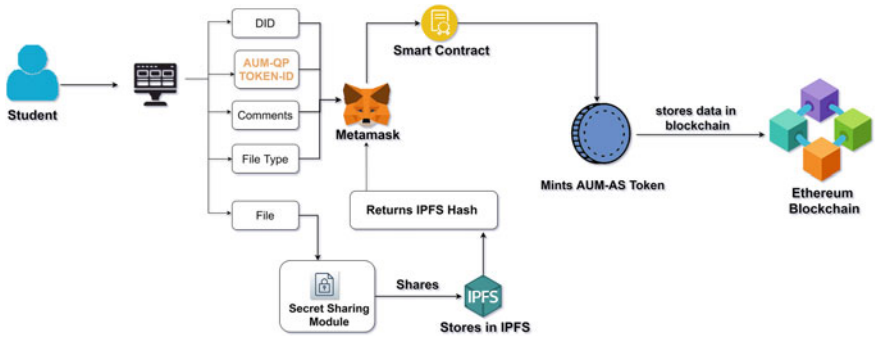


Fig. 3 Submission by student and creation of AUM-AS token

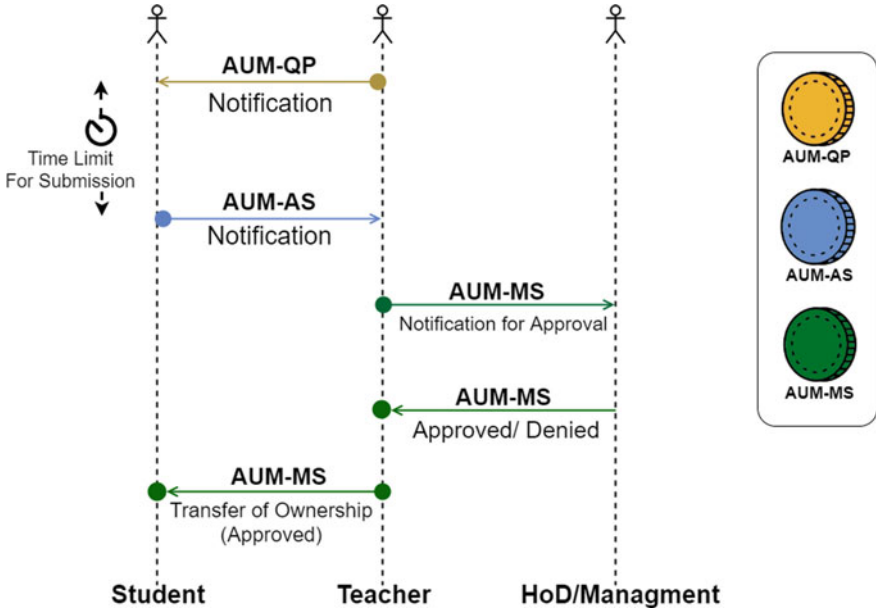


Fig. 4 Ownership of AUM-QP, AUM-AS, AUM-MS tokens

We have already discussed the assignment created by the teacher and submission by the student. The ownership of the AUM-QP Token is owned by the teacher who creates the question paper; similarly, the AUM-AS Token ownership is with the student. Other stakeholders notify about the Token-IDs with which they can view the submissions. In case of evaluations that require the issuance of mark sheets, then the head of departments are also notified, and the mark sheet tokenized using ERC-721 standard referred as ***AUM-MS*** (Green Token). The notification is sent to the Head of Department for approval. The head can either Approve or Deny the mark sheet based on the decision appropriately notified to the teacher. In case if it is approved, then

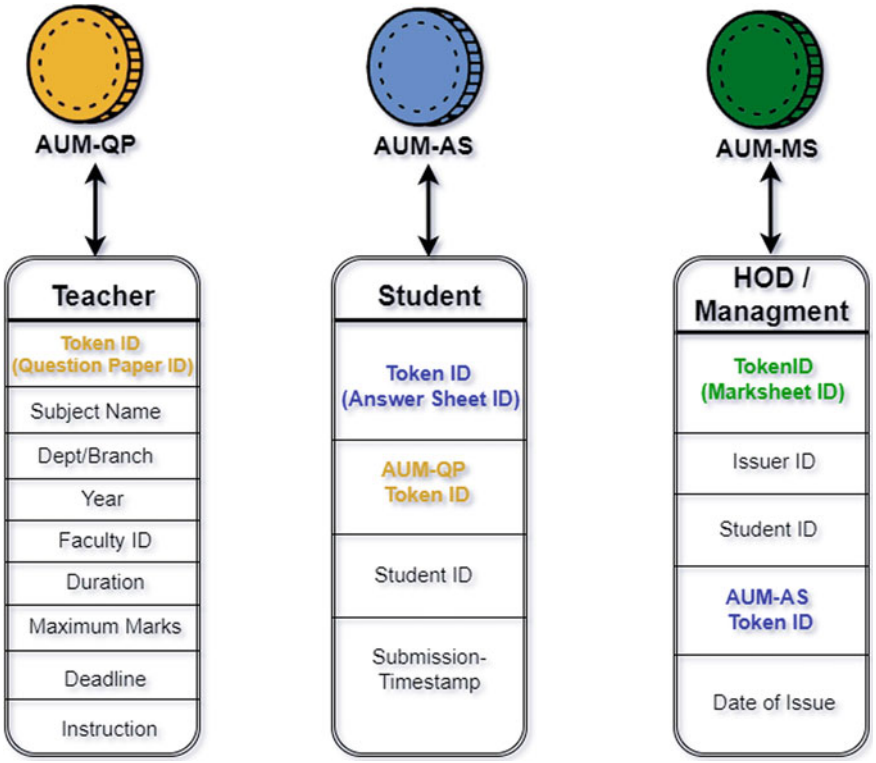


Fig. 5 Detailed view of ERC-721 tokens

the token AUM-MS Token is transferred to the student. Figure 5 elaborates about the three color tokens used in the above use case.

3.2.2 Online Semester Examination

Major universities conduct their final examination at the end of each semester with a certain weightage given to Continuous assessment and final assessment. We are using the InterPlanetary Name System (IPNS) to conduct the End Semester Examination, which can be either a Multiple Choice Questions (MCQ) or Subjective as shown in the Fig. 6.

The Examination cell or the faculty in charge can submit the final question papers, which are tokenized following the method explained earlier as ***AUM-QP Tokens*** (Gold Token). All the different question papers for different branches, years are hosted in the IPNS. The students can take the examination in the given time period. The answers submitted are tokenized as *AUM-AS Token* (Blue Token) and stored in the blockchain. The UMS can be configured either to perform the automatic

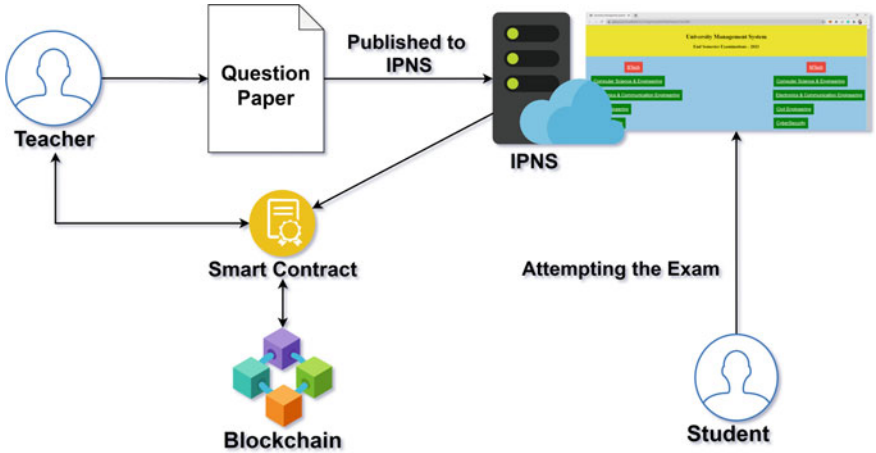


Fig. 6 End semester examination hosted in IPNS

evaluation or manual evaluation, followed by which the results can be published, which is tokenized as ***AUM-MS*** (Green Token). After approval from the Exam cell and/or management, this token can be transferred to the student. The project is hosted in GitHub under the name Distributed Ledger Framework for an Adaptive University Management System [18].

4 Conclusion

We have introduced the challenges we faced because of the COVID-19 pandemic and the sheer need for an Adaptive University Management System. We have discussed Blockchain Technology's potential in handling the challenges and specific requirements put forth by the present scenario in the teaching-learning process. We looked over some of the current Blockchain and IPFS-based implementations. We presented our architecture built on the Ethereum Blockchain and supported by IPFS, IPNS, and secret sharing. The proposed work uses two Ethereum standards of identity management and tokenization: ERC-725 and ERC-721. We strongly believe this framework would bring greater accountability to stakeholders, transparency on the learning and evaluation process, auditability by the regulatory organization for ranking like the National Institutional Ranking Framework (NIRF), Atal Ranking of Institutions on Innovation Achievements (ARIIA). The cost-effectiveness of this proposed system depends on the choice of the Blockchain and the required feature-set.

5 Future Work

We can extend this UMS to an Inter-UMS that can help expand the scope and ease of use by implementing an oracle, an agent that collects real-world external data from various third parties then verifies and shares to the smart contract on-demand basis. That can be used for advertising and promoting with the help of real-world data in the application. The UMS can also monitor and document Library management, Leave management, Publication management, and Innovation management within the university.

References

1. UNESCO's Education and COVID-19: challenges and opportunities (2020). https://en.ccunesco.ca/idealab/education-and-covid-19-challenges-and-opportunities
2. Amrita University Management System (AUMS). Available: https://www.amrita.edu/aums/about
3. Ramaguru R, Minu M (2021) Blockchain terminologies. NamChain open initiative research lab. https://github.com/NamChain-Open-Initiative-Research-Lab/Blockchain-Terminologies
4. Ramaguru R, Sindhu M, Sethumadhavan M (2019) Blockchain for the internet of vehicles. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T., Kashyap R. (eds) Advances in Computing and Data Sciences. ICACDS 2019. Communications in Computer and Information Science, vol 1045. Springer, Singapore. https://doi.org/10.1007/978-981-13-9939-8_37
5. Gopalakrishnan P, Ramaguru R (2019) Blockchain based waste management. Int J Eng Advan Technol 8(5):2632–2635
6. Kripa M, Nidhin Mahesh A, Ramaguru R, Amritha PP (2021) Blockchain framework for social media DRM based on secret sharing. In: Senjyu T, Mahalle PN, Perumal T, Joshi A (eds) Information and communication technology for intelligent systems. ICTIS 2020. Smart Innovation, Systems and Technologies, vol 195. pp. 451–458. Springer, Singapore. https://doi.org/10.1007/978-981-15-7078-0_43
7. Shrivastava AK, Vashistth C, Rajak A, Tripathi AK (2019) A decentralized way to store and authenticate educational documents on private blockchain. 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, vol 1, pp 1–6, https://doi.org/10.1109/ICICT46931.2019.8977633
8. Marjit U, Kumar P (2020) Towards a decentralized and distributed framework for open educational resources based on IPFS and blockchain. 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, pp 1–6, doi: https://doi.org/10.1109/ICCSEA49143.2020.9132841
9. Gajendran N (2020) Blockchain-based secure framework for elearning during COVID-19. Indian J Sci Technol 13(12):1328–41
10. Sun Han, Xiaoyue Wang, Xinge Wang (2018) Application of blockchain technology in online education. Int J Emerg Technol Learn 13.10
11. Aishwarya Nandakumar, Amritha PP, Lakshmy KV, Vijaya Sai Talluri (2012) Non linear secret sharing for gray scale images. Procedia Engineering, vol 30, pp 945–952. https://doi.org/10.1016/j.proeng.2012.01.949
12. InterPlanetary File System (IPFS)—Content Addressed, Versioned, P2P File System. https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf
13. InterPlanetary Name System (IPNS). Available: https://docs.ipfs.io/concepts/ipns/
14. Ramaguru R (2020) Metamask usage guide. NamChain Blog. Available: https://namchain.blogspot.com/p/metamask.html

15. ERC-721 Token Standard. Available: http://erc721.org/
16. Bitcoin Colored Coins. Available: https://en.bitcoin.it/wiki/Colored_Coins
17. Wood Gavin (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum project Yellow Paper. 151.2014 (2014): pp. 1–32. Available: https://ethereum.github.io/yellowpaper/paper.pdf
18. Distributed Ledger Framework for an Adaptive University Management System GitHub Page. https://github.com/Amrita-TIFAC-Cyber-Blockchain/Distributed-Ledger-Framework-for-an-Adaptive-University-Management-System

Fuzzy Model-Based Robust Clustering with Multivariate *t*-Mixture Distributions



Wajid Ali, Ishtiaq Hussain, and Miin-Shen Yang

Abstract In general, multivariate Gaussian distribution (MGD) is the most used probability model. However, MGD is not a good probability model for clustering under the circumstance with outliers or noisy points. In this case, multivariate *t* distribution (MtD) should be a better choice than MGD because MtD has more heavy tail than MGD. In this article, we propose a fuzzy model-based clustering on multivariate *t*-mixture distributions, called fuzzy model-based *t*-distribution (F-MB-T) clustering, which is a feasible choice for robustness in clustering data sets with outliers or noisy points. To show the novelty of the proposed F-MB-T algorithm, we use numerical as well as real data to compare the proposed F-MB-T with some existing methods, such as fuzzy model-based Gaussian mixture, EM with *t*-mixture (EM-T) and fuzzy classification maximum likelihood with *t*-mixture (FCML-T). The experimental results and comparisons demonstrate that the F-MB-T gives good aspects. This presents the proposed F-MB-T as a good model-based clustering algorithm.

Keywords Clustering · Fuzzy sets · Model-based clustering · Fuzzy model-based multivariate *t* (F-MB-T) · Outlier · Noisy data

1 Introduction

Clustering is an important tool for data analysis to group data into homogeneous clusters. It has various applications in the literature. Clustering has been applied in natural sciences, intelligent system, data mining, image processing, database acquisition, social science, agriculture, economics, life sciences, and biology [1–7]. In general, clustering methods include two areas. One is a model-based approach.

W. Ali · I. Hussain · M.-S. Yang (✉)
Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li, Taoyuan 32023, Taiwan
e-mail: msyang@math.cycu.edu.tw

W. Ali
Department of Mathematical Sciences, Karakoram International University, Gilgit, Gilgit-Baltistan, Pakistan

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_25

Another is a non-parametric approach. The model-based approach uses a mixture distribution of probability models, where the expectation and maximization (EM) algorithm is a commonly used method [8]. For a non-parametric approach, a clustering method follows an objective function with similarity or dissimilarity measures. In these non-parametric clustering methods, the prototype-based clustering, such as k-means, fuzzy c-means, and possibilistic c-means, is well-used [9–11]. Banfield and Raftery [12] proposed a model-based clustering to improve these existing algorithms proposed by Scott and Symons [13] and Symons [14] under a mixture of multivariate Gaussian distributions. Although multivariate Gaussian distribution is a commonly used model, it is not a good choice for clustering under the circumstance with outliers or noisy points. As we know that multivariate t-distribution should be a better choice than multivariate Gaussian distribution because multivariate t-distribution has heavy tail so that it will be robust for outliers or noisy points. In this sense, Peel and McLachlan [15] considered the robust mixture model by using multivariate t-distributions with the EM algorithm, called EM-T. In Peel and McLachlan [15], they also gave more properties and compared them with discussing their strength and weaknesses. Afterwards, Liu [16], Liu and Rubin [17], McLachlan and Peel [18], and Lo and Gottardo [19] made more studies on these EM-T methods. Yang et al. [20] considered the fuzzy type of classification maximum likelihood (FCML) procedure with multivariate t-distributions, called FCML-T. Yang et al. [21] considered a fuzzy model-based Gaussian (F-MB-Gauss) clustering in which they combined the model-based Gaussian [12] with fuzzy c-partition memberships [22, 23] for the F-MB-Gauss clustering. Clustering analysis is an important tool to separate data points into similar groups. In this paper, our goal is to extend the F-MB-Gauss proposed by Yang et al. [21] on multivariate t-mixture distributions, called fuzzy model-based t (F-MB-T) clustering. The proposed F-MB-T should be more robust than the F-MB-Gauss when outliers or noisy points present in data sets. To show the novelty and usefulness of the proposed F-MB-T algorithm, we use simulated data with outlier and noisy points and also real data sets to compare with the existing algorithms of F-MB-N, EM-T, and FCML-T algorithms. The experimental results show that the proposed F-MB-T has better aspects than these existing methods.

2 Fuzzy Model-Based Clustering Using Multivariate t-Distribution

Model-based Gaussian clustering proposed by Banfield and Raftery [12] is an essential tool to partition data into similar and dissimilar clusters by using probability mixture models. Let $X = \{x_1, ..., x_n\}$ be randomly selected from a d-variate mixture model with the Gaussian probability distribution $N(x; a_k, \Sigma_k) = (2\pi)^{-d/2}|\Sigma_k|^{-1/2}e^{-(1/2)(x-a_k)^T\Sigma_k^{-1}(x-a_k)}$. Let $W = \{W_1, \ldots, W_c\}$ be a hard c-partition on X with indicator functions $\{z_1, \ldots, z_C\}$ with $z_k(x) = 1$ as $x \in W_k$, and $z_k(x) = 0$ otherwise. The model-based Gaussian objective function is given by

$J(z,\theta;X)=\sum_{i=1}^{n}\sum_{k=1}^{c}z_{ki}\ln f_k(x_i;\theta_k)$, where $f_k(x_i;\theta_k)=N(x_i;\mu_k,\Sigma_k)$, and $z_{ki}=z_k(x_i)$ is the membership function with $z_{ki}\in\{0,1\}$. The model-based Gaussian clustering algorithm is iterated by the updating equations to minimize the objective function $J(z,\theta;X)$ with $\hat{a}_k=\sum_{i=1}^{n}z_{ki}x_i/\sum_{i=1}^{n}z_{ki}$ and $\hat{\Sigma}_k=\sum_{i=1}^{n}z_{ki}(x_i-\hat{a}_k)(x_i-\hat{a}_k)^T/\sum_{i=1}^{n}z_{ki}$. Yang et al. [21] gave a F-MB-Gauss clustering by combining the model-based Gaussian mixture model with fuzzy c-partition memberships. The objective function of F-MB-Gauss proposed by Yang et al. [21] is with

$$J_{\text{F - MB - Gauss}}(u,a,\sum;X)=\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln N(x_i;a_k,\Sigma_k)-w\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln u_{ki}$$

where u_{ki} is a fuzzy c-partition with the condition $\sum_{k=1}^{c}u_{ki}=1\forall i$, and the entropy term $-w\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln u_{ki}$ of membership functions u_{ki} is considered in $J_{\text{F - MB - Gauss}}(u,a,\sum;X)$. We next use multivariate t-distributions to extend the F-MB-Gauss to fuzzy model-based clustering with multivariate t-mixture distributions (F-MB-T).

The multivariate t-distribution $f(x;a,\sum,v)$ is with

$$f(x;a,\Sigma,v)=\frac{\Gamma\left(\frac{v+d}{2}\right)|\Sigma|^{-\frac{1}{2}}}{(\pi v)^{\frac{d}{2}}\Gamma\left(\frac{v}{2}\right)\left[1+\frac{\delta(x,a;\Sigma)}{v}\right]^{\frac{v+d}{2}}}$$

where $\delta(x\ ;a,\Sigma)=(x-a)^T\Sigma^{-1}(x-a)$. Σ is the covariance matrix, and $\Gamma(v)$ is the Gamma function with $\Gamma(v)=\int_0^\infty s^{v-1}e^{-s}\mathrm{d}s$. We know that the mixture model of multivariate t-distributions can be used as a scale mixture model of multivariate Gaussian distributions. Let Y be the latent variable with $X|y\ N(x;a,\Sigma/y)$ and $Y\ G(y;v/2,v/2)$, where $N(x;a,\Sigma/y)$ is the multivariate Gaussian distribution, and $G(y;\alpha,\beta)$ denotes the gamma distribution. For multivariate t-mixtures, we consider the multivariate t-distribution $f(x;a_k,\sum_k,v_k)$ as the kth component of distributions where a_k is the mean of the kth cluster; $\sum_k$ is covariance matrix;$v_k=(v_1,\ldots,v_c)$ is the degree of freedom for the kth cluster. Thus, a new F-MB-T objective function is proposed as

$$\begin{aligned}J_{\text{F - MB - T}}(u,a,\sum,v;X)&=\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln f(x;a_k,\sum_k,v_k)-w\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln u_{ki}\\&=\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln\left[N\left(x_i;a_k,\sum_k/y_{ki}\right)G(y_{ki};v_k/2,v_k/2)\right]\\&\quad-w\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\ln u_{ki}\end{aligned}$$

where Y is the latent variable over the gamma distribution $G(y; v/2, v/2)$ such that $X|y\ N(x; a, \Sigma/y)$ and $Y\ G(y; v/2, v/2)$. We know that $w \geq 0$ is a learning parameter. According to our simulation results, we use the learning rate with $w = 0.999^t$, where the t is the iteration number. To get the updating equations for minimizing the F-MB-T objective function, the following Lagrangian is used:

$$\tilde{J}_{\text{F - MB - T}}(u, a, \sum, v, \lambda; X) = \sum_{i=1}^{n}\sum_{k=1}^{c} u_{ki} \ln f(x; a_k, \sum_k, v_k) - w\sum_{i=1}^{n}\sum_{k=1}^{c} u_{ki} \ln u_{ki} - \lambda\left(\sum_{k=1}^{c} u_{ki} - 1\right)$$

The necessary conditions of y_{ki} to maximize $\tilde{J}_{\text{F - MB - T}}(u, a, \sum, v, \lambda; X)$ are as follows:

$$y_{ki} = \frac{(v_k + d - 2)}{\{(x_i - a_k)^T \sum_k^{-1}(x_i - a_k) + v_k\}} \tag{1}$$

Differentiating $\tilde{J}_{\text{F - MB - T}}(u, a, \sum, v, \lambda; X)$ with respect to the fuzzy membership function, u_{ki}, we get the updating equation with $\frac{\partial \tilde{J}_{\text{F - MB - T}}}{\partial u_{ki}} = \ln f(x; a_k, \sum_k, v_k) - w \ln u_{ki} - w - \lambda$. Set $\frac{\partial \tilde{J}_{\text{F - MB - T}}}{\partial u_{ki}} = 0$. We get $w \ln u_{ki} = \ln f(x; a_k, \sum_k, v_k) - \ln e^{(w+\lambda)}$, and then, we have

$$\hat{u}_{ki} = \left[f(x; a_k, \sum_k, v_k)\right]^{\frac{1}{w}} / \sum_{s=1}^{c}\left[f(x; a_k, \sum_k, v_k)\right]^{\frac{1}{w}} \tag{2}$$

Differentiate $\tilde{J}(u, a, \sum, v, \lambda; X)$ w.r.t a_k we obtain

$$\hat{a}_k = \sum_{i=1}^{n} \hat{u}_{ki} y_{ki} x_i / \sum_{i=1}^{n} \hat{u}_{ki} y_{ki} \tag{3}$$

Thus, the updated equation of $\sum_k$ after differentiating $\tilde{J}_{\text{F - MB - T}}(u, a, \sum, v, \lambda; X)$ with respect to $\sum_k$ becomes

$$\hat{\sum}_k = \sum_{i=1}^{n} u_{ki} y_{ki} (x_i - a_k)(x_i - a_k)^T / \sum_{i=1}^{n} u_{ki} y_{ki} \tag{4}$$

We consider initialization for covariance matrix $\sum_k^{(0)}$ as follows. Suppose $D_k = \text{sort}\left\{d_{ki}^2 = ||x_i - a_k||^2 : d_{ki}^2 > 0,\ i \neq k\right\} = \left\{d_{k(1)}^2, d_{k(2)}^2, \ldots, d_{k(n^*)}^2\right\}$ and

$$\sum_{k}^{(0)} = d_k^2([\sqrt{n}])I_d \tag{5}$$

where I_d is a $d \times d$ identity matrix. Similarly, we can differentiate $\tilde{J}_{\text{F - MB - T}}(u, a, \sum, v, \lambda; X)$ with respect to v_k for obtaining the degree of freedom $v_k = (v_1, \ldots, v_c)$. Thus, the following equation is obtained:

$$\ln\left(\frac{v_k}{2}\right) - \psi\left(\frac{v_k}{2}\right) + 1 + \sum_{i=1}^{n} u_{ki}(\ln y_{ki} - y_{ki}) / \sum_{i=1}^{n} u_{ki} + \psi((v_k + p)/2) - \log((v_k + p)/2) = 0 \tag{6}$$

where $\psi(v) = \frac{\mathrm{d}}{\mathrm{d}v} \ln \Gamma(v)$ is the digamma function with the correction term $\psi((v_k + p)/2) - \log((v_k + p)/2)$ for imputing the conditional mean value y_i in $\ln y_i$. For the learning parameter w, we use the decreasing learning function with

$$w^{(t)} = 0.999^t \tag{7}$$

We can summarize the F-MB-T algorithm as follows:

The F-MB-T clustering algorithm

S1: Fix $2 \leq c \leq n$ and $\varepsilon > 0$. Give the initials $w^{(0)} = 1$, $v^{(0)}$, $a_k^{(0)}$ and $y_{ki}^{(0)}$.
S2: Compute $\sum_k^{(0)}$ by Eq. (5) and let $t = 0$.
S3: Update $\hat{u}_{ki}^{(0)}$ using $w^{(0)}$, $a_k^{(0)}$, $y_{ki}^{(0)}$ and $\sum_k^{(0)}$ by Eq. (2) and let $t = t + 1$.
S4: Compute $\hat{a}_k^{(t)}$ using $\hat{u}_{ki}^{(t-1)}$ and $y_{ki}^{(t-1)}$ by Eq. (3).
S5: Update $w^{(t)}$ by Eq. (7).
S6: Update $\hat{\sum}_k^{(t)}$ using $\hat{a}_k^{(t)}$, $y_{ki}^{(t-1)}$ and $\hat{u}_{ki}^{(t-1)}$ by Eq. (4).
S7: Update $v_k^{(t)}$ using $\hat{u}_{ki}^{(t-1)}$ and $y_{ki}^{(t-1)}$ by Eq. (6).
S8: Update $y_{ki}^{(t)}$ using $\hat{a}_k^{(t)}$, $v_k^{(t)}$ and $\hat{\sum}_k^{(t)}$ by Eq. (1).
S9: Compute $\hat{u}_{ki}^{(t)}$ using $w^{(t)}$, $\hat{a}_k^{(t)}$, $\hat{\sum}_k^{(t)}$ and $y_{ki}^{(t)}$ by Eq. (2).
S10: Update $\hat{a}_k^{(t+1)}$ using $\hat{u}_{ki}^{(t)}$ and $y_{ki}^{(t)}$ by Eq. (3).
If $\max \left\| \hat{a}_k^{(t+1)} - \hat{a}_k^{(t)} \right\| < \varepsilon$, then STOP.
Else $t = t + 1$ and return to S3.

3 Numerical Examples and Comparisons

In this section, we demonstrate the performance of the proposed F-MB-T clustering algorithm. Several synthetic and real data sets are used to show the novelty of the F-MB-T algorithm. We also give the comparisons of the proposed F-MB-T with

EM-T [15], FCML-T [20], and F-MB-Gauss [21]. The accuracy rate (AR) is used as a criterion for evaluating the performance of a clustering algorithm. AR, defined as $\text{AR} = \sum_{i=1}^{k} r_i/n$, is the percentage of data points in which they are correctly identified by the clustering algorithm, where n is the number of data points, and r_i is the number of points in C_i' that are also in C_i in which $C = \{C_1, C_2, \ldots, C_c\}$ is the set of c clusters for the given data set, and $C' = \left\{C_1', C_2', \ldots, C_c'\right\}$ is the set of c clusters generated by the clustering algorithm.

3.1 Synthetic Data Set

Here, we demonstrate the novelty of our proposed algorithm using simulation data sets. We also compare our proposed algorithm F-MB-T with F-MB-Gauss, EM-T, and FCML-T algorithms and show the novelty by considering accuracy rate (AR).

Example 1 A data set is used in this example to demonstrate the novelty of the proposed F-MB-T algorithm over the F-MB-Gauss, EM-T, and FCML-T algorithms. A data set, called 2-cluster Gaussian data, includes 1250 data points that are generated from a Gaussian mixture with the following parameters: $\alpha_1 = \alpha_2 = \frac{1}{2}$, $a_1 = (5\ \ 2)$, $a_2 = (-5\ \ 3)$, $\sum_1 = (1\ \ 0; 0\ \ 1)$, $\sum_2 = (1\ \ 0; 0\ \ 1)$, to which 150 noisy points were added using the uniform distribution over the interval $(-20, 20)$ on each variable. The original data set with noisy points is shown in Fig. 1a. Figure 1b–d represents the clustering results of F-MB-Gauss, EM-T, and FCML-T with two cluster centers after 19, 69, and 16 iterations. The final clustering result of the proposed F-MB-T algorithm with four iterations is shown in Fig. 1e. Results of ARs from the F-MB-Gauss, EM-T, FCML-T, and F-MB-T are shown in Table 1. It is seen that the proposed algorithm F-MB-T has a higher accuracy rate (AR = 1.00) as compared to F-MB-Gauss (AR = 0.601), EM-T (AR = 0.969), and FCML-T (AR = 0.981). This shows the novelty of our method.

Example 2 Here, we used another simulated data set to demonstrate the novelty of the proposed F-MB-T algorithm over the F-MB-Gauss, EM-T, and FCML-T algorithms. A data set, called 3-cluster Gaussian data, has 550 data points that are generated from the two-variate Gaussian mixture model with the following parameters: $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$, $a_1 = (0\ \ 4)$, $a_2 = (4\ \ 0)$ and $a_3 = (-4\ \ 0)$ while $\sum_1 = (5\ \ 1.5; 1.5\ \ 2.5)$,$\sum_2 = (1\ \ 0; 0\ \ 0.2)$ and $\sum_3 = (2\ \ -1.5; -1.5\ \ 1.5)$. We add 50 noisy points in two variables using the uniform distribution over the interval $(-10, 10)$. The original data set with the outlier is shown in Fig. 2a. Figure 2b–d represents the clustering results of F-MB-Gauss, EM-T, and FCML-T with three cluster centers after 25, 57, and 351 iterations. The final clustering result of the proposed F-MB-T algorithm with 11 iterations is shown in Fig. 2e. Results of ARs from the F-MB-Gauss, EM-T, FCML-T, and F-MB-T are shown in Table 2. We find that our proposed algorithm F-MB-T has a higher accuracy rate (AR = 0.921) as compared to F-MB-Gauss (AR = 0.601), EM-T (AR = 0.881), and FCML-T (AR = 0.901). This shows the novelty of our method.

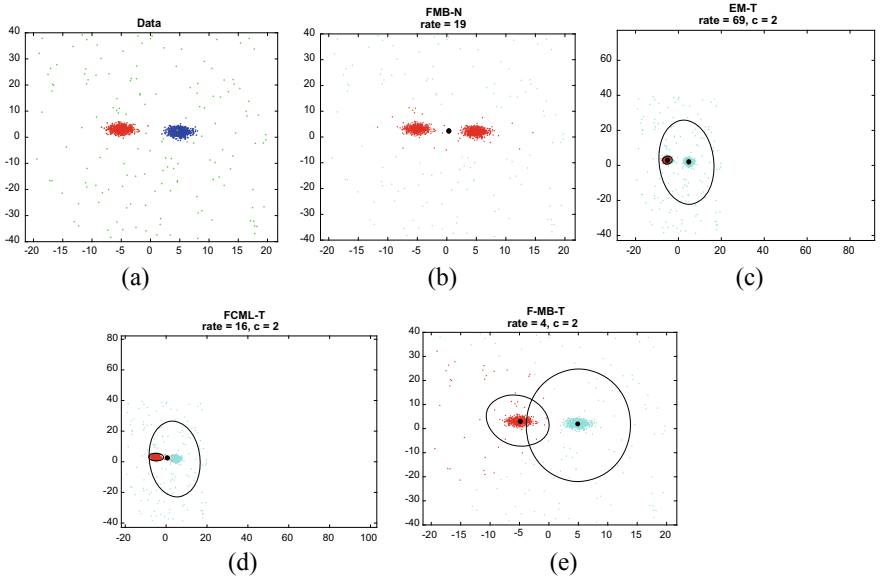


Fig. 1 **a** Original 2-cluster data set; **b** F-MB-Gauss results after 19 iterations; **c** EM-T results after 69 iterations; **d** FCML-T results after 16 iterations; **e** F-MB-T results after four iterations

Table 1 Comparison of F-MB-Gauss, EM-T, FCML-T with F-MB-T using accuracy rate for example 1

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.601	0.969	0.981	1.00

Example 3 We use a data set in this example to demonstrate the novelty of the proposed F-MB-T algorithm over the F-MB-Gauss, EM-T, and FCML-T algorithms. A data set, called 4-cluster Gaussian data, includes 900 data points that are generated from the two-variate Gaussian mixture with the following parameters: $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{4}$, $a_1 = (2\ 4)$, $a_2 = (4\ 0)$, $a_3 = (-4\ 0)$, $a_4 = (0\ -5)$, $\sum_1 = (1\ \ 0;\ 0\ \ 1)$, $\sum_2 = (1\ \ 0;\ 0\ \ 1)$ $\sum_3 = (2\ -1.5; -1.5\ \ 1.5)$ and $\sum_4 = (1\ \ 0; 0\ \ 0.5)$, to which 100 noisy points were added using the uniform distribution over the interval $(-10, 10)$ on each variable. The original data set with noisy points is shown in Fig. 3a. Figure 3b–d represents the clustering results of F-MB-Gauss, EM-T, and FCML-T with four cluster centers after 32, 500, and 49 iterations. The final clustering result of the proposed F-MB-T algorithm with 20 iterations is shown in Fig. 3e. Results of ARs from the F-MB-Gauss, EM-T, FCML-T, and F-MB-T are shown in Table 3. It is seen that our proposed algorithm F-MB-T has a higher accuracy rate (AR = 0.898) as compared to F-MB-Gauss (AR = 0.681), EM-T (AR = 0.771), and FCML-T (AR = 0.571). This shows the novelty of our method.

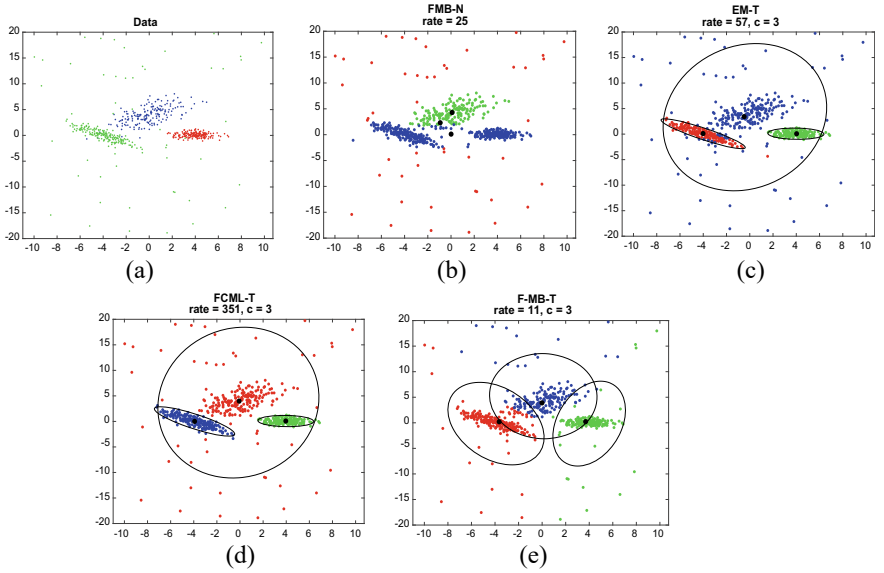


Fig. 2 **a** Original 3-cluster data set; **b** F-MB-Gauss results after 25 iterations; **c** EM-T results after 57 iterations; **d** FCML-T results after 351 iterations; **e** F-MB-T results after 11 iterations

Table 2 Comparison of F-MB-Gauss, EM-T, FCML-T with F-MB-T using accuracy rate for example 2

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.601	0.881	0.901	0.921

3.2 Real Data Sets

In this section, we use some data sets from well-known database UCI [24] which consists of real sets related with classification, clustering, and regression analysis. To show the novelty, we compare our proposed F-MB-T algorithm with F-MB-Gauss, EM-T, and FCML-T.

Example 4 The real data set *Iris* is considered in this example. This data set consists of four dimensions, namely as sepal length, sepal width, petal length, and petal width. The data set has three components *Iris setosa*, *Iris virginica*, and *Iris versicoclor* [24]. By using F-MB-T, we obtain the average accuracy rate (AR = 0.966) while using the other three methods, we get (AR = 0.802) from F-MB-Gauss, (AR = 0.901) from EM-T, and (AR = 0.911) from FCML-T after taking the average of 50 different initializations. This shows the significant aspect and novelty of the proposed F-MB-T. Table 4 shows the ARs comparison of four methods.

Example 5 We consider the flea beetle data from Lubischew [25], where *heikertingeri* represents cluster one, *heptapotamica* as cluster two, and *concinna* in cluster

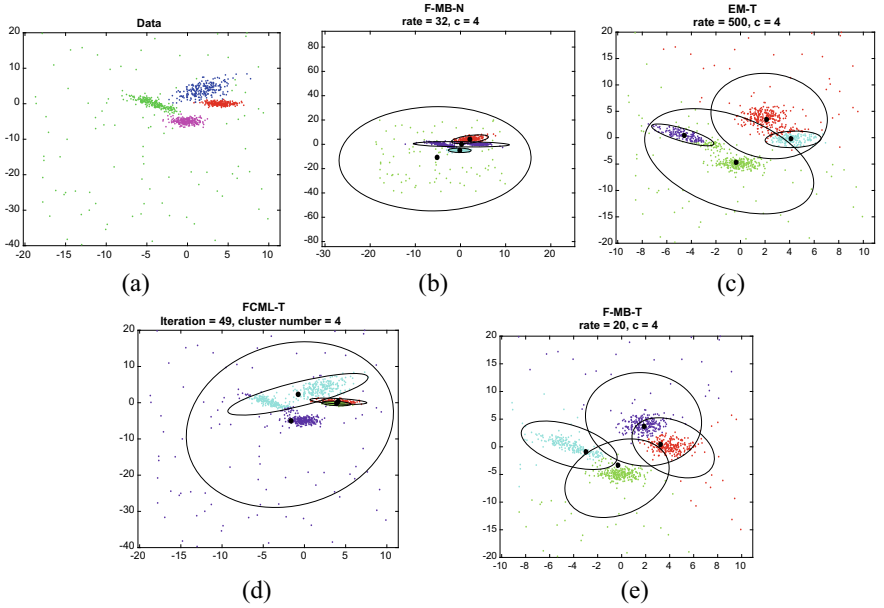


Fig. 3 **a** Original 4-cluster data set; **b** F-MB-Gauss results after 32 iterations; **c** EM-T results after 500 iterations; **d** FCML-T results after 49 iterations; **e** F-MB-T results after 20 iterations

Table 3 Comparison of F-MB-Gauss, EM-T, and FCML-T with F-MB-T using accuracy rate for example 3

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.681	0.771	0.571	0.898

Table 4 Comparison of F-MB-Gauss, EM-T, and FCML-T with F-MB-T using average accuracy rate for example 4

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.802	0.901	0.911	0.966

three. The data set consists of 74 data points with three species: *concinna*, *heikertingeri*, and *heptapotamica*. Points were obtained by measuring two characteristics of a beetle: the maximal width of the aedeagus in the fore-part in microns and the front angle of the aedeagus (1 unit 7.5′). By using F-MB-T, we obtain the average accuracy rate (AR = 0.932) while using the other three methods we receive (AR = 0.881) from F-MB-Gauss, (AR = 0.891) using EM-T and (AR = 0.901) from FCML-T after taking the average of 50 different initializations. This shows the good aspect and significance of the proposed F-MB-T. Table 5 shows the ARs comparison of four methods.

Table 5 Comparison of F-MB-Gauss, EM-T, and FCML-T with F-MB-T using average accuracy rate for example 5

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.881	0.891	0.901	0.932

Example 6 Breast cancer is most frequently found in vast majority countries females, and leading countries are Australia/New-Zealand, United Kingdom, Sweden Finland, Denmark Belgium (Highest rate), the Netherlands, and France. (causes, methods). According to World Health Organization, the common risk factors for cancers are due of tobacco use, alcohol intake, dietary factors, including insufficient fruit and vegetable consumption, overweight and obesity, lack of physical activity, chronic infections from helicobacter pylori, hepatitis B virus (HBV), hepatitis C virus (HCV) and some human papillomavirus (HPV), and environmental and occupational risks including ionizing and non-ionizing radiation. We consider real data set breast cancer [24] which consists of 699 samples and the following eight features, namely clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, bare nuclei, bland chromatin, normal nucleoli, mitoses and one output variable. By implementing F-MB-T, we get a better average accuracy rate (AR = 0.935) while using the other three methods we obtain (AR = 0.908) from F-MB-Gauss (AR = 0.901) using EM-T and (AR = 0.661) through FCML-T after taking the average of 50 different initializations results are shown in Table 6. This shows the novelty and application of our proposed algorithm.

Example 7 (Leukemia data set [26]).

Leukemia is one type of cancer related with blood. Generally, it has four types, namely acute lymphoblastic leukemia (ALL), acute myeloid leukemia (AML), chronic lymphocytic leukemia (CLL), and chronic myeloid leukemia (CML). According to global cancer statistics, it has 2.4% number of cases seen in 2018 around the world. Leukemia cancer is mostly found in children, but another age, gender can also suffer from this disease. We consider Golub et al. [26] where they collected leukemia data set which consists of 38 leukemia patients with their biological sample array and genes as features. From these genes, 27 are acute lymphoblastic leukemia, and 11 are acute myelogenous leukemia. Golub et al. [26] divided these patients according to separate clinical treatment. We use 700 genes after sorting their variance for our algorithm setting, and we use standardization, so each variable has zero mean and one variance. We implement the F-MB-T and F-MB-N, EM-T and FCML-T algorithms to the Leukemia data set where the clustering results are shown in Table 7. It is seen that the F-MB-T algorithm obtains the highest average accuracy

Table 6 Comparison of F-MB-Gauss, EM-T, and FCML-T with F-MB-T using average accuracy rate for example 6

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.908	0.901	0.661	0.935

Table 7 Comparison of F-MB-Gauss, EM-T, and FCML-T with F-MB-T using average accuracy rate for example 7

AR	F-MB-Gauss	EM-T	FCML-T	F-MB-T
	0.651	0.662	0.662	0.904

rate AR = 0.904 as compared to F-MB-N = 0.651, EM-T = 0.662, and FCML-T = 0.662 after taking 50 different initializations. This shows the novelty of our proposed algorithm.

4 Conclusions

Multivariate *t*-distribution is vigorous choice and works efficiently when the situations are with heavy-tailedness in data settings [27]. It is the novelty of the multivariate *t*-distribution where it involves the parameter of degree of freedom that is useful for outlier accommodation and noisy point detection. In this article, we proposed the method of F-MB-T with a new objective function to overcome the weaknesses of Gaussian mixture model which is not robust choice due to presence of outliers/noisy points. We can see that the accuracy rate of our proposed F-MB-T algorithm is higher than existing methods, such as F-MB-Gauss, EM-T, and FCML-T, for both artificial and real data sets. This shows the good aspect of our proposed F-MB-T. In the future, we will extend the proposed method to robust type of clustering algorithms under unknown cluster number and not dependent on the selection of parameters.

References

1. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice Hall, New Jersey
2. Kaufman L, Rousseeuw J (1990) Finding groups in data: an introduction to cluster analysis. Wiley, New York
3. Yang MS, Hung WL, Chang-Chien SJ (2005) On a similarity measure between LR-type fuzzy numbers and its application to database acquisition. Int J Intell Syst 20:1001–1016
4. Jang D, Chun T, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16:1370–1386
5. Chang ST, Lu KP, Yang MS (2015) Fuzzy change-point algorithms for regression models. IEEE Trans Fuzzy Syst 23:2343–2357
6. Yang MS, Ali W (2019) Fuzzy Gaussian Lasso clustering with application to cancer data. Math Biosci Eng 17:250–265
7. McLachlan GJ, Basford KE (1988) Mixture models: inference and applications to clustering. Marcel Dekker, New York
8. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B 39:1–22
9. Yang MS (1993) A survey of fuzzy clustering. J Math Comput Model 18:1–16
10. Krishnapuram R, Keller JM (1993) A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1:98–110

11. Pal NR, Pal K, Keller JM, Bezdek JC (2005) A possibilistic fuzzy c-means clustering algorithm. IEEE Trans Fuzzy Syst 13:517–530
12. Banfield JD, Raftery AE (1993) Model-based Gaussian and non-Gaussian clustering. Biometrics 49:803–821
13. Scott AJ, Symons MJ (1971) Clustering methods based on likelihood ratio criteria. Biometrics 27:387–397
14. Symons MJ (1981) Clustering criteria and multivariate normal mixtures. Biometrics 37:35–43
15. Peel D, McLachlan GJ (2000) Robust mixture modelling using the t distribution. Stat Comput 10:339–348
16. Liu C (1997) ML Estimation of the multivariatet distribution and the EM algorithm. J Multivar Anal 63:296–312
17. Liu C, Rubin DB (1995) ML estimation of the t distribution using EM and its extensions. ECM ECME Stat Sin 5:19–39
18. McLachlan GJ, Peel D (1998) Robust cluster analysis via mixtures of multivariate t-distributions. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 1451:658–666
19. Lo K, Gottardo R (2012) Flexible mixture modeling via the multivariate t distribution with the Box-Cox transformation: an alternative to the skew-t distribution. Stat Comput 22:33–52
20. Yang MS, Tian YC, Lin CY (2014) Robust fuzzy classification maximum likelihood clustering with multivariate t-distributions. Int J Fuzzy Syst 16:566–576
21. Yang MS, Chien SJ, Nataliani Y (2019) Unsupervised fuzzy model-based Gaussian clustering. Inf Sci 481:1–23
22. Yang MS, Su CF (1994) On parameter estimation for the normal mixtures based on fuzzy clustering algorithms. Fuzzy Sets Syst 68:13–28
23. Yang MS, Tian YC (2015) Bias-correction fuzzy clustering algorithms. Inf Sci 309:138–162
24. Blake CL, Merz CJ (1988) UCI machine learning repository, UCI repository of machine learning database, a huge collection of artifical and real-world data sets. [Online]. Available: https://archive.ics.uci.edu/ml/index.php
25. Lubischew AA (1962) On the use of discriminant functions in taxonomy. Biometrics 18:455–477
26. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
27. Lin TI, Lee JC, Hsieh WJ (2007) Robust mixture modeling using the skew t distribution. Stat Comput 17:81–92

Tackling COVID-19 Infodemic Using Deep Learning



Prathmesh Pathwar and Simran Gill

Abstract Humanity is battling one of the most deleterious virus in modern history, the COVID-19 pandemic, but along with the pandemic there's an infodemic permeating the pupil and society with misinformation which exacerbates the current malady. We try to detect and classify fake news on online media to detect fake information relating to COVID-19 and coronavirus. The dataset contained fake posts, articles and news gathered from fact checking websites like politifact whereas real tweets were taken from verified twitter handles. We incorporated multiple conventional classification techniques like Naive Bayes, KNN, Gradient Boost and Random Forest along with Deep learning approaches, specifically CNN, RNN, DNN and the ensemble model RMDL. We analyzed these approaches with two feature extraction techniques, TF-IDF and GloVe Word Embeddings which would provide deeper insights into the dataset containing COVID-19 info on online media.

Keywords Recurrent neural network (RNN) · Convolutional neural network (CNN) · Deep learning · COVID-19 · Fake news · Word embeddings · RMDL

1 Introduction

Misinformation is defined as objectively incorrect information that is not supported by scientific evidence and expert opinion [1]. Since COVID-19 was declared as a pandemic of global scale in March 2020, there has been speculations and theories about the cause, symptoms, treatment, number of cases, fatality levels, the age of

P. Pathwar (✉) · S. Gill
Indian Institute of Information Technology, Allahabad, India
e-mail: iit2016028@iiita.ac.in

S. Gill
e-mail: iit2016088@iiita.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_26

infected, conspiracy theories and much more ersatz information revolving in social media causing panic and contention among people. The problem does not engender just from presence of fake news, but the rapid pace at which it spreads among 3.6 billion social media users in 2020 [2] with numbers exponentially increasing in future. According to Economic Times, Whatsapp (which is now part of Facebook), limited forwarding viral messages to one person down from five [6], when Indian government advised the need to control hoaxes on social media platforms as it eventually poses a threat to society [3]. In an article by Forbes, other companies followed suite by deploying some mechanism to tackle dissemination of fake content and give priority to reliable, credited sources like WHO and government offices [5]. Further, studies indicated that misinformation was more frequently tweeted than science based evidence [7, 8], concluding that misinformation can stymie the efforts for preventive measures [9], which could exacerbate the damage done by pandemic.

Globally institutions began to realize that fake news regarding COVID-19 poses a similar if not higher threat that the virus itself, with news like "*Alcohol is a cure for COVID-19*" which led to multiple deaths in Iran [4] and in recent times hate and attack on Asian communities have increased several fold given to the baseless conspiracies regarding China. According to Business Today [10], WHO SEARO released a statement in April 2021 that a video of it *predicting 50,000 deaths* in India is fake, pointing to the fact even after a year of battling the *Infodemic*, much is to be done.

In this paper, we aim to detect the fake news surrounding COVID-19 and Coronavirus on social media platforms. We analyze the dataset containing a 10,700 fake and real tweets. We analyze different conventional and deep learning models using Term-Frequencey Inverse-Document-Frequency (TFIDF) [11] and Glove word embeddings [12] to detect fake COVID-19 information efficiently. This analysis would aid institutions to filter information and tag fake news and misinformation at early stage so that it's pernicious effects can be mitigated. Further, this underscores the importance of proper models required to espouse society and medical institutions to save resources and time on tackling misinformation and concentrate towards the pandemic, acting as an anodyne for the infected. ML researchers can obtain greater insight regarding the deep learning and feature extraction paradigms with respect to COVID-19 pandemic.

2 Literature Review

There is a lot of research done regarding fake news classification and detection on social media and multiple approaches have been suggested to improve the models. We look into some of the works done by researchers worldwide on the topic of text classification, fake news detection and approaches using deep learning .

Hadeer Ahmed, Issa Traore, Sherif Saad [13] used n-gram analysis and different feature extraction techniques: SVM, LSVM and Logistic Regression on the 2 datasets obtained from kaggle.com, tripadvisor.com and other sources to detect fake news

and spam opinions (as both are closely related). It was observed that with increase in number of features the accuracy of classification models increased achieving a best accuracy of 90%.

Jamal Abdul Nasir, Osama Subhani Khan, Iraklis Varlamis [14] created a hybrid model of CNN and RNN on FA-KES and ISOT dataset to detect fake news. They compared their results with general techniques like Random forest, Logistic Regression, Naive Bayes, KNN and more and also with standalone DNN and CNN models. Their model achieved an accuracy of 0.60 ± 0.007 on FA-KES and 0.99 ± 0.02 on ISOT.

Jiangfeng Zenga, Yin Zhangb, Xiao Ma [15] introduced a novel approach to use semantic correlations across multimodalities using images in FND-SCTI and enhance training the learning of text representation via Hierarchical Attention Mechanism. They explored two more approaches using multimodal variational autoencoder and multimodal fusion eigenvector which and tested with Twitter and Weibo fake news dataset to obtain best accuracy among seven competetive approaches.

Mykhailo Granik, Volodymyr Mesyura [16] used naive bayes algorithm to detect fake news from a dataset created using facebook posts and news pages. They obtained an accuracy of 74% which indicates that a simple model can still be used to detect and classify fake news.

Yonghun Jang, Chang-Hyeon Park and Yeong-Seok Seo [17] proposed a fake news analysis modelling method to get best features from Quote retweet that would positively impact fake news detection. They also employed a method to collect, tweets, posts and user details from twitter and format it so that it can be easily used for fake news analysis.

Aswini Thota, Priyanka Tilak, Simeratjeet Ahluwalia, Nibhrat Lohia [18] used Dense Neural Network along with a custom tuned TFIDF and BoW approach to detect classify fake news. There accuracy using Word2Vec word embeddings was comparatively lower to TFIDF and BoW, for which they explored possible reasons but eventually there best accuracy was 94.21% on FNC-1 dataset.

Álvaro Ibrain RodríÃ¯Â¿Â½guez, Lara Lloret Iglesias [19] compared the performance of convolutional based model, LSTM based model and BERT on fake news to attain an accuracy of, after hyper-parameter tuning, 0.91 with LSTM based model, 0.94 with Convolutional based model and 0.98 using BERT.

HT Le, C Cerisara, A Denis [20] explored the efficiency of deep CNNs in classification domain, they discovvered that the models with short and wide CNNs are most efficient at word level, on the other hand deep models perform better when input text is encoded as a sequence of characters but still are worse on average. They also introduced a new model derived from DeepNet for text inputs.

Table 1 Dataset distribution

Attributes	Real	Fake	Total
Train	3360	3060	6420
Validation	1120	1020	2140
Test	1120	1020	2140
Total	5600	5100	10700



(a) All samples (b) Real samples (c) Fake samples

Fig. 1 Wordclouds for label distribution

3 Dataset and Experimental Setup

The analysis and prediction of fake news is conducted on the dataset originally provided by Parth et al. [21] as part of their research and contains fake news regarding *COVID-19* from various fact-checking websites like Politifact, NewsChecker , Boomlive and from tools like Google fact-check-explorer and IFCN chatbot in the form of posts, tweets, articles. The real news is collected from Twitter using verified twitter handles. The dataset is divided into three parts, train, validate and test. The train dataset contains total 6420 samples, whereas validation and test each contain a data 2140 samples. The distribution is provided in Table 1.

The dataset has comparable number of Fake and Real samples for both test and train sets, making the training and testing unskewed which would give better insights about the performance of classifiers (Fig. 1).

4 Methodology

4.1 Data Cleaning

Data Cleaning increases the validity and quality of the model by removing incorrect information and refining the data. Raw data might contain null or missing values

which might skew the results for our model. Following methods were be taken into account while cleaning the data:

- **Renaming and removing the columns** : To get name of columns in more readable form, the names are renamed.
- **Renaming the values of a feature**: To form data in consistent and homogeneous format, all data is converted into numerical format.

4.2 Data Pre-processing

The data needs to be preprocessed and refined before before using it to test model through punctuation removal, removing stop words, stemming, lower casing comments, lemmatization and tokenization. A generic function was used to remove punctuation, hyperlinks and lower-casing the text.
Original sample: `Our daily update is published. States reported 734k tests 39k new cases and 532 deaths. Current hospitalizations fell below 30k for the first time since June 22. https://t.co/wzSYMe0Sht`

4.3 Removing Stop Words

Stop words are words that create unnecessary noise when training models. These words may be pronouns, conjunctions, prepositions, hyperlinks or any other words that might be inconsistent with the dataset and hamper the results and training process. We used the nltk's stopwords which contains 179 stopwords specifically for english language.
After removing stopwords: `daily update published states reported 734k tests 39k new cases 532 deaths current hospitalizations fell 30k first time since June 22`

4.4 Stemming

Stemming essentially is morphing the words into their original form, such as "swimming" is converted to "swim", this removes the multiple variation of words in different formats and boils them down to their meaning for efficient and better learning. We use nltk's Snowball stemmer which is an improved version of Porter stemmer.
After stemming: `daili updat publish state report test new case death current hospit fell first time sinc June`

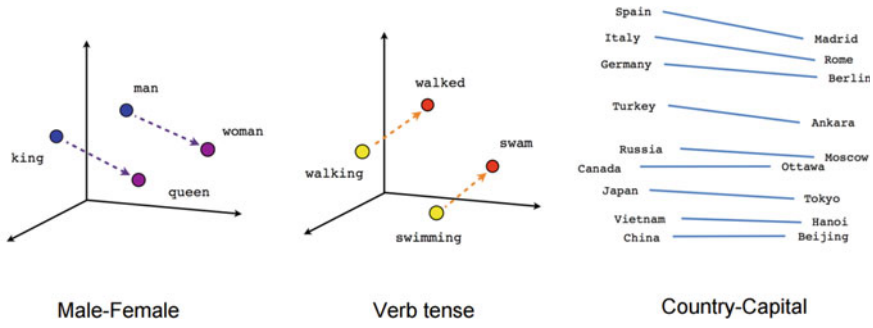


Fig. 2 Relationship between vectors

4.5 Feature Extraction

Feature Extraction is a major step as it adversely affect the performance of the model. Words cannot be used in raw format and hence need to be converted to vectors to enable models to learn. We take two approaches for this, word embeddings and TF-IDF.

4.5.1 *Word Embeddings*

Word Embeddings are pretrained multidimensional vectors which represent relationship among words as vectors. The vectors for "airplane", "aeroplane", "plane", and "aircraft" would be similar as they represent almost the same meaning whereas "airplane" and "book" would have dissimilar vectors. It can be considered that they define relationship between the words, but need large external corpus to obtain the relationships, and are computationally intensive to use as dimensions increase. We use the 50 dimensional twitter GloVe (Global Vectors for Word Representation) [12] word vectors which were made by evaluating Wikipedia 2014 and Gigawords 5 translating to 6B tokens, and 400K vocab which were used to obtain embedding with 50, 100, 200 and 300 dimensions. The objective function for it can be defined as

$$g(v_i - v_j, \tilde{v_j}) = \frac{P_{ik}}{P_{jk}} \tag{1}$$

where v_i refers to the word vector of word i, and P_{ik} refers to the probability of word k to occur in context of i (Fig. 2).

4.5.2 *Term Frequency—Inverse Document Frequency*

Term Frequency—Inverse Document Frequency (TF-IDF) [11] is common tool in NLP for converting a list of text documents to a matrix representation. It is obtained by calculating Term Frequency and Inverse Document Frequency to obtain normalized vectors. The sparse matrix formed measures the times a number appeared in the document and as the count can be large it is also divided by the length of the document as a way of basic normalization. Though obtaining TF-IDF is not computational intensive, it doesn't capture the relationship between words. Term Frequency can be defined as the ratio of occurrence of term t in sentence to the number of words in the sentence.

$$TF(d,t) = \frac{\sum_{t \in d} t}{|d|} \tag{2}$$

Inverse Document Frequency has a characteristic that it normalizes effect of TF , while raising the significance of rare terms. For example, words like "*the*","*where*" might dominate if only Term Frequency is used but using IDF with TF reduces this impact and gives a normalized output. Here N denotes the number of documents and $f(t)$ denotes the number of documents containing term t.

$$IDF(d,t) = log\left(\frac{(N_d)}{f(t)}\right) \tag{3}$$

Together we get the TF-IDF matrix.

$$W(d,t) = TF(d,t) * log\left(\frac{(N_d)}{f(t)}\right) \tag{4}$$

4.6 *Data Classification Using Conventional Models*

In this paper, we can broadly categorize the models used. At one part we would use conventional algorithms and on the other hand we will explore deep learning models and networks, to make sure that the dataset is evaluated from multiple perspectives. We used Multinomial Naive Bayes, Gradient Boosting classifier, K Nearest Neighbours and Random Forest classifier, with optimum hyperparameters to conduct the experiment.

4.7 Deep Learning Models

Since the advent of Perceptron and Neural Network , there has been progress in the field of machine learning with some models been widely used as the basis for complex and highly specific models. Deep Learning models usually require test and training set from which the models can obtain information and "learn" to be later applied to new inputs. We used 4 different deep learning models - CNN, RNN using GRU and LSTM recurrent units, DNN and RMDL (which is an ensemble model of CNN, RNN and DNN) for our text classification objective.

5 Experimental Results

The dataset was pre divided into train, test and validation sets. It contained 6420 training samples, 2140 validation samples and 2140 test samples. The samples contains "tweet" and their respective label "fake" or "real". The labels were changed to 0 for "fake" and 1 for "real". Further the data contained, hyperlinks, emojis and other noise which was cleaned as part of data pre-processing. The testing is done using two feature selection techniques - TFIDF and Glove word embedding. The models used accuracy as the criteria to optimize and minimize loss function. For deep learning models, sparse categorical crossentropy was used as the loss function and Adam optimizer was used.

- ***Sparse Categorical Crossentropy*** : Derived from cross entropy, this is used when the classes are mutually exclusive, so the values for ith label would be present and other would be zero (hence the term "sparse").

$$J(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)\right] \tag{5}$$

  where w is the model parameters, y_i is true label and $\hat{y}_i$ is predicted label.
- ***Accuracy*** : It is the calculated as the proximity between the predicted class label and actual class label or the ratio of correctly predicted (True Positive + True Negative) to total number of samples. Accuracy can be calculated as:

$$Accuracy = \frac{TruePositive + TrueNegative}{Total} \tag{6}$$

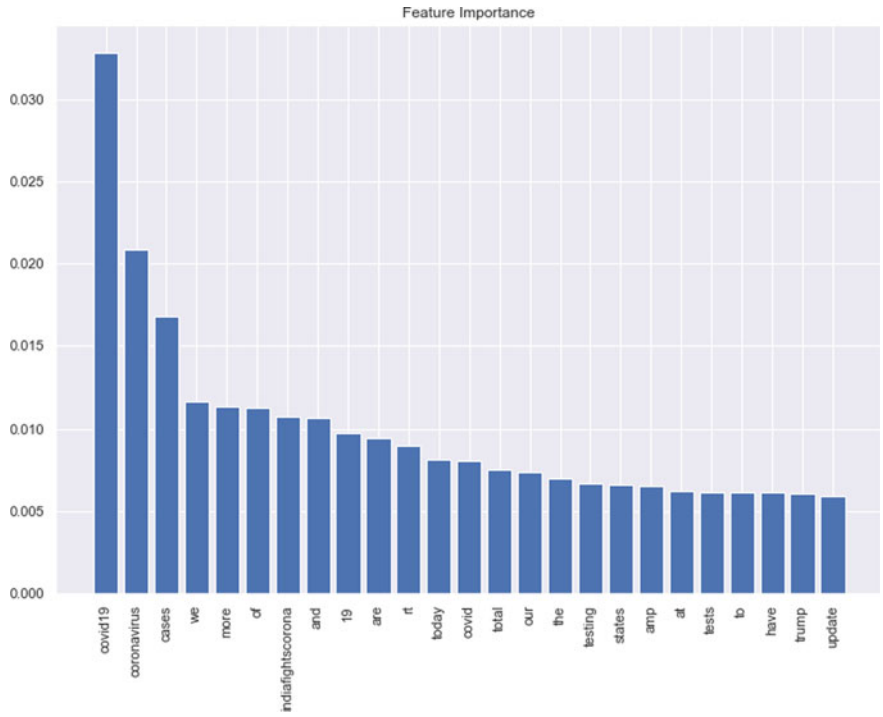


Fig. 3 Feature importance using TFIDF

Table 2 Accuracy of random forest using TFIDF and word embedding

Metric (in percentage)	TFIDF	Word embedding
Accuracy	91.30	87.05
Precision	93.64	90.64
Recall	89.46	83.92
F1-Score	91.50	87.15

5.1 Classification Algorithms for Predicting Fake COVID-19 News

5.1.1 *Random Forest Analysis*

The two feature extraction techniques were used along-with Random Forest, the number of decision trees (n estimators) used were 64. An analysis was done on the accuracy by using TFIDF and Word Embedding.

Table 3 Accuracy of multinomial naive bayes using TFIDF

Metric (in percentage)	TFIDF
Accuracy	90.88
Precision	88.57
Recall	94.82
F1-Score	91.59

Table 4 Accuracy of gradient boost using TFIDF and word embedding

Metric (in percentage)	TFIDF	Word embedding
Accuracy	86.86	84.90
Precision	90.92	85.86
Recall	83.21	85.17
F1-Score	86.89	85.52

Thus, the best performance of **Random Forest** using TFIDF to obtain vectors was 91.30% and with word embedding it was observed as 87.05%. Hence TFIDF provided better performance with **91.30%** accuracy (Fig. 3 and Table 2).

5.1.2 *Multinomial Naive Bayes Analysis*

In the case of Multinomial Naive Bayes, the analysis was only done with TFIDF due to the fact that word embedding contained negative vector values which couldn't be used with Naive Bayes as it works on conditional probabilities which are non-negative (Table 3).

Thus, the best performance of **Naive Bayes** was **90.88%** but word embedding wasn't compatible and hence wasn't used but it obtained a good accuracy when compared to an ensemble model such as Random Forest.

5.1.3 *Gradient Boost Analysis*

The two feature selection extraction were used along-with Gradient Boosting Classifier. The number of estimators used were 100, and the boosting took place for 100 epochs.

As seen in Table 4, the best performance of **Gradient Boost** using TFIDF to obtain vectors was 86.86% and with word embedding it was observed as 84.90%. Hence TFIDF provided better performance with **86.86%** accuracy but word embeddings provided higher recall indicating lower false negatives (Table 5).

Table 5 Accuracy of KNN using TFIDF and word embedding

Metric (in percentage)	TFIDF	Word Embedding
Accuracy	90.32	86.54
Precision	93.02	84.89
Recall	88.12	90.35
F1-Score	90.50	87.54

Table 6 Accuracy of DNN using TFIDF and word embedding

Metric (in percentage)	TFIDF	Word Embedding
Accuracy	93.87	92.47
Precision	94.03	95.10
Recall	94.28	90.26
F1-Score	94.15	92.62

5.1.4 *K Nearest Neighbours Analysis*

The two feature extraction techniques were used along-with K Nearest Neighbours. The number of neighbours used for this analysis was 6 for both TFIDF and word embedding.

As seen in the given table, the best performance of **KNN** using TFIDF to obtain vectors was 90.32% and with word embedding it was observed as 86.54%. Hence TFIDF provided better performance with **90.32%** accuracy but word embeddings provided higher recall indicating lower false negatives.

5.2 Deep Learning Models

The standalone models used here were each trained for 10 epochs wheras each model in RMDL was trained for 8 epochs. Training required substantial time ranging from 2 minutes for DNN to 30 h for RNN.

5.2.1 *Dense Neural Network Analysis*

DNNs can be modeled in a variety of ways by introducing different layers of dropout, convolutional and dense layers, in summary we used 4 hidden dense layers with dropout after each layer and this was used to train 9.4M parameters with TFIDF and 3.2M parameters with Glove word embeddings after an addition of flattening layer before the output layer (Table 6).

Table 7 Accuracy of DNN using TFIDF and word embedding

Metric (in percentage)	TFIDF	Word embedding
Accuracy	89.11	93.92
Precision	90.42	93.80
Recall	88.57	94.64
F1-Score	89.49	94.22

As seen in table VI, the best performance of **DNN** using TFIDF to obtain vectors was 93.87% and with word embedding it was observed as 92.47%. Hence TFIDF provided better performance with **93.87%** accuracy but word embedding provided higher precision indicating higher true positives.

5.2.2 *Convolutional Neural Network Analysis*

Similar to DNN, CNN hosts a variety of models that can be used for our task, we used 6 convolutional layers with varying filter sizes along with a single concatenation layer. Also 5 average pooling layers were used was used to extract important features. For TFIDF 6.5M parameters were used to train compared to 2.2M used with word embedding (Table 7).

We obtained an accuracy of 89.11% using TFIDF by training 6.5M parameters and 93.92% using word embeddings by training 2.2M parameters. The case we see with CNN is quite different from all the models we analyzed, the performance with word embeddings was much better compared to TFIDF, this can be due to a few reasons. Firstly the input layer used with word embeddings was an Embedding Layer which provided advantage in terms of reducing sparse vectors being created (as a by product of TFIDF) and maintaining word relationships intact during feature selection by convolutional layer. Another factor weighs in is that the dimensions are fixed for TFIDF whereas we can define the dimensions for word embedding which (in this case) proved to perform better. Hence the best accuracy of the two was **93.92%**

5.2.3 *Recurrent Neural Network*

GRU and LSTM are two generally used Recurrent units in RNN. To gain an insight about the performance for our use case both were deployed having similar model structure with just the recurrent layers changing between the two where for GRU, all recurrent layers were GRU and for LSTM, the recurrent layers were LSTM. Training of both models was also computationally expensive with taking almost **30 h** to train a the model in both cases. But here we observe that the performance of both models varies by a hairline for both feature extraction techniques. One reason that there is not much difference is that the dataset used isn't huge and hence doesn't have much

Table 8 Accuracy of RNN using GRU

Metric (in percentage)	TFIDF	Word embedding
Accuracy	92.75	92.14
Precision	92.66	94.15
Recall	93.67	90.62
F1-Score	93.11	92.35

Table 9 Accuracy of RNN using LSTM

Metric (in percentage)	TFIDF	Word Embedding
Accuracy	92.71	92.24
Precision	92.57	93.12
Recall	93.57	91.96
F1-Score	93.07	92.54

vocabulary from which can long sequences can be created and put to use by LSTM. On the other hand lack of data size doesn't use the advantages of GRU having no memory cells. Hence, both perform similarly, but the best performance comes out to be **92.75%** for RNN with GRU and TFIDF. We can also notice that the precision for both approaches was better for word embeddings which is as previously seen in most of the other models (Table 8).

- **Gated Recurrent Units:** The structure of RNN comprised of 4 GRU layers each followed by dropout layers and eventually the dense layer to train 1.9M parameters. TFIDF and Word Embeddings were used to extract feature vectors and achieve an accuracy of **92.75%** and **92.14%**.
- **Long Short Term Memory:** The structure of RNN comprised of 4 LSTM layers each followed by dropout layers and eventually the dense layer to train 2.4M. TFIDF and Word Embeddings were used to extract feature vectors and achieved an accuracy of **92.71%** and **92.24%** respectively (Table 9).

5.2.4 *Random Multimodel Deep Learning*

RMDL uses CNNs, DNNs and RNNs, with varying layers, nodes, epochs to introduce normalization in the results leveraging ensemble learning [25]. It uses TFIDF for training DNN but Glove word embeddings for training CNN and RNN. We set the numbers of models used to 3 each and tuned minimum nodes, minimum number of layers and dropout to be passed as parameters to the model (Table 10).

The above tables provide insights into the accuracy models used where in DNN-x (x is just a index number used to name model). Although RMDL shouldn't be evaluated on a model by model basis, but we see that the performance of models vary greatly in terms of performance which underscores the fact that the hyper parameters

Table 10 Accuracy using RMDL

Model	Feature extraction	Accuracy
DNN-0	TFIDF	93.50
DNN-1	TFIDF	93.83
DNN-2	TFIDF	93.83
CNN-0	Glove word embeddings (50D)	73.27
CNN-1	Glove word embeddings (50D)	73.22
CNN-2	Glove word embeddings (50D)	92.89
RNN-0	Glove word embeddings (50D)	77.28
RNN-1	Glove word embeddings (50D)	91.40
RNN-2	Glove word embeddings (50D)	91.26

ultimately impact the performance of the models in a huge way. The combined accuracy achieved by RMDL was **92.75%**

5.3 Comparative Study

Analysing the best accuracy of each algorithms as shown in Table XI, the best accuracy of **93.92%** for classifying tweets in COVID-19 dataset. The major pattern we see that with most models, performance with TFIDF excels with that of Word Embedding. There can be some possible reasons for this observation.

- Word Embeddings doesn't have the ability to form relationships between new occurring words and use them for training. Though FastText have the ability to use Bag of Words to form vectors for new vocabulary [26]. Most used words in the COVID-19 dataset were *COVID-19* and *coronavirus* which prior to the pandemic weren't mainstream except medical articles and papers, this unavailability of corpus containing the above words translated to absence of their vectors and relationships in Glove Word Embeddings. The same doesn't happen for TFIDF as it uses the whole available vocabulary in train data to form the vectors.
- Overfitting is a common scenarios while using word embeddings. As word embedding is a complex form of word representation (in addition to the limited vocabulary) it is very much likely that the train data is overfitted in our case.
- Other con of using complex word representation is that they carry more hidden information which is particularly not useful for our case but we see in results indicating that word embeddings employ relationship between words to obtain better precision.

We also see a stark difference in deep learning models and conventional models being used, signalling to the fact that deep learning approaches, although computationally expensive can provide greater performance in text classification task (Table 11).

Table 11 Comparison of performance for models used

Model	TFIDF	Word Embedding
Random forest	91.30	87.05
Naive bayes	90.88	-
Gradient boost	86.86	84.90
KNN	90.32	86.54
DNN	**93.87**	92.47
CNN	89.11	**93.92**
RNN (GRU)	92.75	92.14
RNN (LSTM)	92.71	92.24
RMDL	92.75	

6 Future Work

This work could be further extended as :

- Since our current dataset contains only 10700 samples, to analyze the effectiveness and accuracy of the algorithms, this research could be extended on a larger dataset having larger set of vocabulary.
- This project is worked upon by using fake news tweets and tweets of official handles of authorities, it could further be extended using the records of more diverse social media platforms or media outlets to enable a holistic approach to the problem
- Word Embeddings used can be from a corpus which contains COVID-19 related terminology to even out any missing vocabulary scenarios.
- 2D or 3D Convolutional layers can be used to extract features more efficiently. Further the deep models can be made wider instead of deeper to enhance performance.

7 Conclusion

This paper attempts to detect and classify fake news regarding *COVID-19* circulating with rapid pace on social media platform twitter using conventional models and deep learning models with different feature extraction techniques TFIDF and GloVe word Embeddings. We found that the best performance of 93.92% was achieved by CNN along with word embeddings. Surprisingly, we observed that performance of TFIDF was better than a more complex word representation technique such as word embeddings for almost all the models used except CNN. Overall deep learning models (CNN,RNN, DNN and RMDL) performed better than conventional classification

approaches. The outcomes of the analysis will aid institutions and researchers to enhance detecting fake news regarding COVID-19 and tackle it at earliest. We were able to detect fake tweets with the best accuracy of **93.92%** using CNN along with Glove word embeddings. This

References

1. Nyhan B, Reifler J (2010) When corrections fail: the persistence of political misperceptions. Polit Behav 32(2):303–330
2. Cement J (2020) Number of social media users 2025. Statista. Accessed 2021-02-22
3. Stefanie Panke (2020) Social media and fake news. aace
4. Nasser Karimi, Jon Gambrell (2020) Hundreds die of poisoning in iran as fake news suggests methanolcure for virus. Times of Israel. Accessed 2021-03-31
5. Bernard Marr (2021) Coronavirus fake news: how Facebook, Twitter, And Instagram Are Tackling The Problem. Forbes. Accessed 2021-03-22
6. Anumeha Chaturvedi (2021) Covid fallout: WhatsApp changes limit on forwarded messages, users can send only 1 chat at a time. Economic Times India. Accessed 2021-03-22
7. Bridgman Aengus et al (2020) The causes and consequences of COVID-19 misperceptions: understanding the role of news and social media. Harvard Kennedy School Misinformation Review 1.3
8. Pulido Cristina M et al (2020) COVID-19 infodemic: more retweets for science-based information on coronavirus than for false information. Int Sociol 35.4:377–392
9. Dixon Graham, Clarke Christopher (2013) The effect of falsely balanced reporting of the autism-vaccine controversy on vaccine safety perceptions and behavioral intentions. Health Educ Res 28(2):352–359
10. WHO warns of Covid-19 fake news; clarifies never predicted deaths in India. Business Today. Accessed 2021-04-09
11. Ramos Juan (2003) Using tf-idf to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning, vol 242, No 1
12. Pennington Jeffrey, Socher Richard, Manning Christopher (2014) Glove: global vectors for word representation. EMNLP 14:1532–1543. https://doi.org/10.3115/v1/D14-1162
13. Ahmed Hadeer, Traore Issa, Saad Sherif (2018) Detecting opinion spams and fake news using text classification. Sec Priv 1(1):e9
14. Jamal Abdul Nasir (2021) Osama Subhani Khan. Iraklis Varlamis, Fake news detection: a hybrid CNN-RNN based deep learning approach. Int J Inform Manag Data Insights 1(1):100007
15. Zeng Jiangfeng, Zhang Yin, Ma Xiao (2020) Fake news detection for epidemic emergencies via deep correlations between text and images. Sustain Cities Soc. 66:102652. https://doi.org/10.1016/j.scs.2020.102652
16. Granik Mykhailo, Volodymyr Mesyura (2017) Fake news detection using naive Bayes classifier. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pp 900–903
17. Seo Yeong-Seok, Jang Yonghun, Park Chang-Hyeon (2019) Fake news analysis modeling using quote retweet. Electronics 8. https://doi.org/10.3390/electronics8121377
18. Thota, Aswini; Tilak, Priyanka; Ahluwalia, Simrat; and Lohia, Nibrat (2018) "Fake News Detection: A Deep Learning Approach," SMU Data Science Review: Vol. 1 : No. 3 , Article 10
19. Ibrain, Álvaro & Lloret, Lara. (2019). Fake news detection using Deep Learning
20. Le, Hoa T., Christophe Cerisara, and Alexandre Denis. "Do convolutional networks need to be deep for text classification?." arXiv preprint arXiv:1707.04108 (2017)
21. Patwa, Parth, **et al**. "Fighting an infodemic: Covid-19 fake news dataset." arXiv preprint arXiv:2011.03327 (2020)

22. Webb Geoffrey I (2010) Naïve Bayes. Encyclopedia of machine learning 15:713–714
23. Guo Gongde et al (2003) KNN model-based approach in classification. OTM Confederated International Conferences On the Move to Meaningful Internet Systems. Springer, Berlin, Heidelberg
24. Yu Shujuan et al (2020) Attention-based LSTM, GRU and CNN for short text classification. J Intel Fuzzy Syst Prep, 1–8
25. Kamran Kowsari, Mojtaba Heidarysafa, Donald E Brown, Kiana Jafari Meimandi, Laura E Barnes (2018) RMDL: random multimodel deep learning for classification. In Proceedings of the 2nd International Conference on Information System and Data Mining (ICISDM '18). Association for Computing Machinery, New York, NY, USA, pp 19-28
26. Bojanowski Piotr et al (2017) Enriching word vectors with subword information. Transac Assoc Comput Ling 5:135–146

Automatic Blood Vessel Segmentation in Retinal Fundus Images Using Image Enhancement and Dynamic Gray-Level Thresholding



J. Jeslin Shanthamalar and R. Geetha Ramani

Abstract Blood vessel is one of the most important retinal regions used for identification of retinal diseases like glaucoma, diabetic retinopathy and occlusion through blood vessel features. This paper presented an automated image processing techniques for blood vessel segmentation through image enhancement and dynamic gray-level thresholding. Proposed approach contains of six processes: color channel selection, image complement, image enhancement, optic disk removal, dynamic gray-level thresholding and binarization. Initially, green color channel is selected from original RGB color fundus images for clear visibility of blood vessels followed by image complement techniques to differentiate blood vessel features with other retinal features. Then, complemented image is further enhanced to improve the visibility of blood vessels including thin vessels for accurate extraction, and then, disk structuring element is applied for optic disk removal. Finally, blood vessel is segmented by applying image binarization using dynamic gray-level thresholding. Proposed approach achieved good results in terms of accuracy and specificity of 95.51 and 99.14% on DRIVE dataset and 95.67 and 98.33% on HRF dataset. Also, experimental results were compared with the state-of-the-art methods and represent our proposed method achieved high accuracy and very much helpful for the ophthalmologist for disease identification during earlier stage.

Keywords Image enhancement · Dynamic gray-level thresholding · Diabetic retinopathy · Glaucoma · Optic disk removal

1 Introduction

Automatic segmentation of retinal blood vessel represents the development of computer aided retinal diagnosis to support ophthalmologist in the field of retinal abnormality identification and also like cardiovascular diseases and biometric system. Retinal blood vessel holds very important information in fundus image analysis which

J. J. Shanthamalar (✉) · R. G. Ramani
Department of Information Science and Technology, Anna University, Chennai, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_27

is used to identify retinal pathology like exudates, cotton wool spots, hemorrhage and diseases like diabetic retinopathy, occlusion and glaucoma. Generally, retinal blood vessel contains one main artery, one main vein and its branches. These two main branches supply blood to the rest of the eye. The changes in these blood vessels and its branches feature will be used as a key attribute to identify retinal diseases during earlier stages through automatic computer aided diagnosis will support the experts to arrest further vision loss. Exact segmentation of blood vessel is very complex due to pathological images and also a time-consuming process [1]. However, reduced visibility of narrow vessels, low contrast images, camera quality, presence of other retinal distractions, poor quality images and lightening effects are an challenging task that will affect the accurate segmentation of blood vessel and efficiency of the algorithm [2, 3].

There are several challenges need to be identified before blood vessel segmentation which is dark lesions, bright lesions, optic disk and image quality [4]. Blood vessels usually have small curves, and branches and very thin vessels have lower reflectance compared to the other retinal surfaces which look more darker [5]. Hence, it is very difficult to segment the blood vessel manually by the experts and also require more time to identify retinal abnormalities. Blood vessels are mostly used for the identification of other retinal region such as optic disk, macula, fovea, left eye, right eye through blood vessel features like vessel density, thickness, distance, parabola curve calculation and ISNT rule calculation. Hence, it is considered as one of the most important key features of retinal disease identification system.

2 Related Works

Retinal blood vessel segmentation is considered as an important key feature because it reveals the subtle changes that happened in the arteries, veins and its branches which is much needed for disease identification [1]. In order to avoid manual analyses and give better segmentation accuracy, many computerized techniques are addressed by researchers in the area of blood vessel segmentation [3, 6, 7]. Unsupervised coarse-to-fine method identifies connected vessels, thin vessels and thick vessels using morphology, spatial dependency and curvature concepts which result in better evaluation through balanced accuracy for blood vessel segmentation is presented [2]. In this approach, performance was imperfect because of false positive rate, and also, the necessity to estimate the vessel diameter to improve the performance is not done.

Blood vessel segmentation from diabetic retinopathy images using tandem pulse coupled neural network and deep learning based support vector machine for features generation and classification and then extraction of blood vessels through firefly algorithm was proposed [1]. This method achieved good accuracy and needs to be improved in the sense of quantifying the curvature of the blood vessel to identify the retinal diseases. A new and efficient method of blood vessel segmentation such as neutrosophic techniques [8] and minimum spanning superpixel tree [9] was proposed. Image processing techniques like multi-scale image enhancement,

local entropy thresholding [10], morphological reconstruction [4], hybrid active contour model [11] and wavelet-based technique [12] were proposed for automatic blood vessel segmentation. Various existing techniques of blood vessel segmentation through computer aided diagnosis and its drawbacks were discussed in our literature survey. The proposed approach presented automatic blood vessel segmentation through image processing techniques with better accuracy and less computation time represents our system's competency.

Section 3 details about methods of blood vessel segmentation and also states the step-by-step process used for blood vessel segmentation. Section 4 explains the details about experimental results, performance analysis of our proposed system and also comparative study with state-of-the-art methods. Finally, Sect. 5 presents the conclusion of our proposed work and future work.

3 Methods of Blood Vessel Segmentation

This section presents an automatic method for blood vessel segmentation to obtain better accuracy for disease prediction. Proposed approach followed five stages of processes for blood vessel segmentation includes green channel selection, image complement and enhancement, optic disk removal, binarization using gray-level thresholding, blood vessel segmentation process and detailed in further subsections. Proposed system overall framework is shown in Fig. 1.

3.1 Color Channel Selection

In our approach, color channel selection is one of the most important stages in blood vessel segmentation. Because retinal fundus images often having poor quality will result in poor segmentation results during blood vessel segmentation process. In order to avoid these distractions, best color channel is chosen for further process. Most of the existing algorithm uses green color channel rather than using red and blue color channel. Proposed method, green channel is selected for further process since it gives clear view of blood vessel including thin vessels even though the image is in low quality. Green channel selection from RGB images is shown in Fig. 2.

3.2 Image Complement and Enhancement

An image complement method complements the green channel image like zero to ones and ones to zeros which results in dark areas to lighter and light areas to darker. This method is used to remove retinal image distractions like dark lesions and also

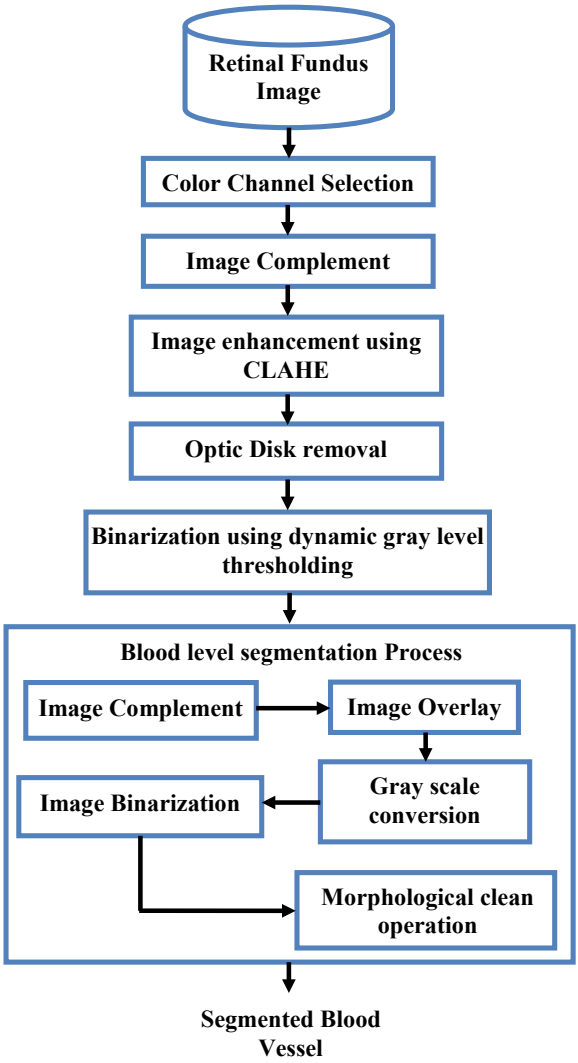


Fig. 1 Overall framework of our proposed system

used to differentiate between blood vessels and other retinal region for better segmentation process. In order to further improve the enhancement of a complemented image, contrast-limited adaptive histogram equalization (CLAHE) algorithm is used. Hence, the quality of a complemented image is further enhanced through its contrast [13]. For this method, full image is processed by small regions using 8×8 tiles with 0.001 clip limit as well as 0.4 uniform distribution parameter is used to create a flat histogram. Image complement and image enhancement process is shown in Fig. 3.

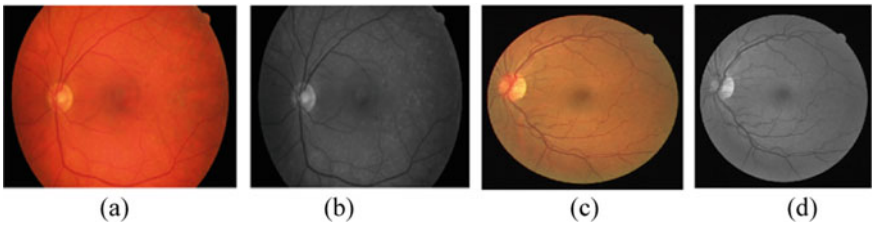


Fig. 2 Green channel selection from HRF and DRIVE dataset **a**, **c** Original image **b**, **d** Green channel image

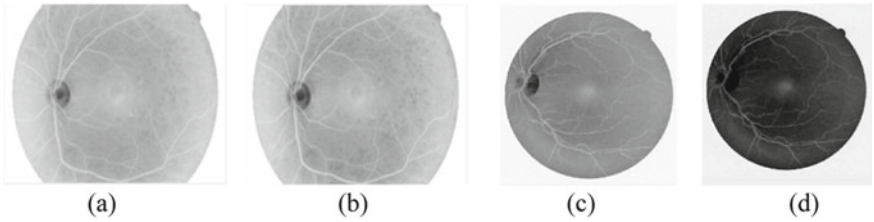


Fig. 3 **a**, **b** Image complement and image enhancement from HRF dataset **c**, **d** image complement and image enhancement from DRIVE dataset

3.3 Optic Disk Removal

During blood vessel segmentation, optic disk removal is very important because optic disk region is the starting point of optic nerve head region. Hence, morphological opening is adopted with disk structuring elements. In our work, two structuring elements such as 5 and 20 disk structuring elements are used for the removal of optic disk from DRIVE and HRF datasets. Because of different resolution images, disk structuring element range is manually calculated for both datasets. This method also removes exudates, cotton wool spots and lesions which are small in size which offers better segmentation. After that, image difference is calculated for disk removal from enhanced image and morphological opening image. Major advantage of this disk removal process is: it will not affect the blood vessels inside the optic disk region and also removes other unwanted features from the retinal image and will give clear view of blood vessels. Optic disk removal process is presented in Fig. 4.

3.4 Binarization Using Gray-Level Thresholding

After removal of optic disk region, image binarization technique is adopted for blood vessel segmentation process. For binarization process, threshold value is fixed to

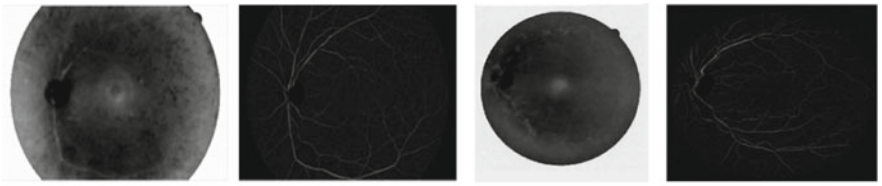


Fig. 4 Optic disk removal. **a**, **c** Morphological opening with disk size = 20 and disk size = 5 **b**, **d** image difference from HRF and DRIVE dataset

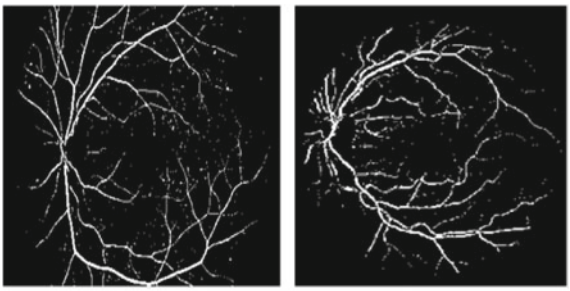


Fig. 5 Binarization using dynamic gray-level thresholding for HRF and DRIVE datasets. **a** Binarization with threshold = 0.12 **b** Binarization with threshold = 0.09

remove unwanted information from the processed image. In our approach, thresholding value calculated dynamically for each image represents the robustness of our system. For this dynamic calculation, our technique adopted global image thresholding using Otsu's method. Hence, this method will convert the intensity image into binary image using gray-level thresholding. Image binarization and gray-level thresholding process are shown in Fig. 5.

This method converts the intensity image having less than global threshold points into black pixels and above into white pixels such as above thresholds considered as blood vessels and below values are considered as background region. For global image thresholding, proposed system utilized MATLAB software inbuilt functions with its default parameter settings. After binarization method, small objects are removed from the binary image in order to remove unwanted information and also better segmentation process. For this removal, pixel count 50 is fixed on DRIVE dataset and pixel count 150 for HRF dataset.

3.5 Blood Vessel Segmentation Process

During binarization stage, unwanted pixels are removed from the binarized image, but in some images, outer layer of retinal fundus image is not eliminated due to high

intensity on that particular region. This process gives better blood vessel segmentation accuracy of our approach, and the processes are clearly presented in Fig. 6.

In order to remove those pixels, proposed system used some set of process including image complement, image overlay, gray scale conversion and morphological operation. Initially, binarized image is complemented to highlight the visibility of outer layer of retinal image followed by image overlay. Image overlay technique overlays the complemented image into original image to remove the outer layer, and its color specification is fixed as [0 0 0]. After overlaying, generated RGB image is converted into gray scale image for further process. Then, gray scale image having higher than 40 pixel densities is considered as blood vessel, and less than 40 pixel densities are removed from the gray scale image which creates the binary image. Finally, morphological clean operation was applied on binary image to remove isolated images, i.e., one valued pixels are surrounded by zero valued pixels.

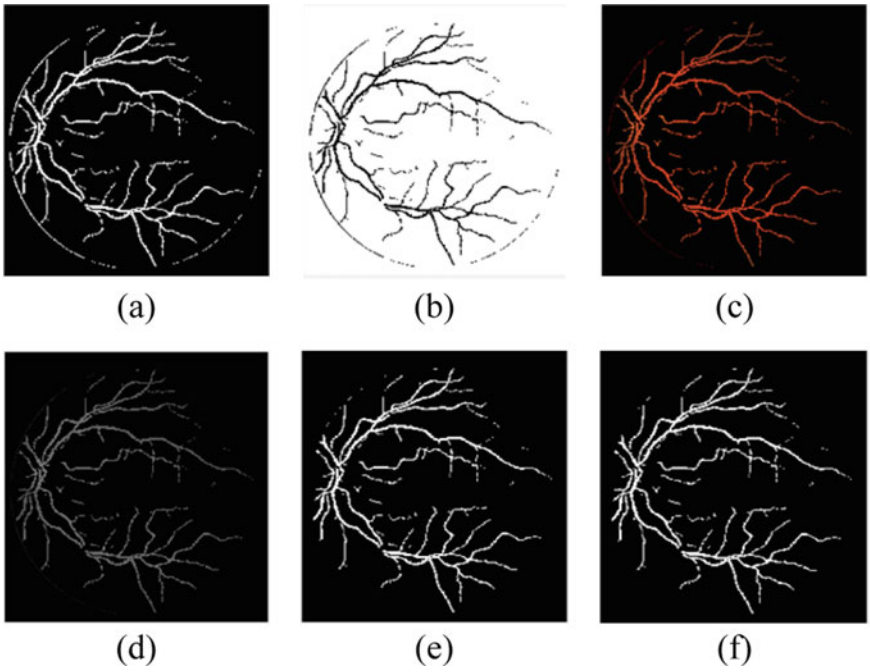


Fig. 6 Set of process used on image from DRIVE dataset for blood vessel segmentation. **a** Binarized image **b** complemented image **c** image overlay **d** gray scale image **e** binarized image having >40 pixel density **f** morphological clean operation

4 Experimental Results

This section presents the details of datasets used, blood vessel segmentation results, results of proposed approach compared with the state-of-the-art methods were listed, and evaluation metrics were calculated using ground truth provided by the experts. Proposed system was implemented in MATLAB R2017b and tested on two publically available retinal fundus image datasets having ground truth for blood vessel segmentation. To perform this process, personal computer system configured with Intel (R) Core (TM) i3, 1.70 GHz processor and 8 GB of RAM used. Furthermore, the methods presented in our approach were very simple and easy to implement without any specific need of hardware requirement which was one of the advantages of our method.

4.1 Dataset Description

Digital Retinal Images for Vessel Extraction (DRIVE) database [14] Contains 40 retinal fundus images with ground truth of blood vessel segmentation, which is marked by two experts were presented. Drive dataset divided into two sets like 20 training set and 20 testing set which is in resolution size of 565 × 584 pixels and acquired by Canon CR5 non-mydriatic 3 charge-coupled-device (CCD) cameras with field of view (FOV) at 45°. This dataset contains 33 images which do not show any signs of diabetic retinopathy, and seven images show the retinal abnormalities including diabetic retinopathy. High-resolution fundus image dataset (HRF) [13] consists of 15 healthy images, 15 glaucoma images and 15 diabetic retinopathy images of totally 45 images. Resolution of each image is in size 3504 × 2336 pixels and also contains ground truth marked by experts for blood vessel segmentation, FOV status, optic disk center point and radius of optic disk.

4.2 Blood Vessel Segmentation

For blood vessel segmentation, all the images from DRIVE and HRF datasets were used. These datasets contain totally 85 images with different resolution, FOV status and intensities. Proposed technique processed all the original images without resizing with less amount of processing time. Figures 7 and 8 show the blood vessel segmentation results from HRF and DRIVE datasets.

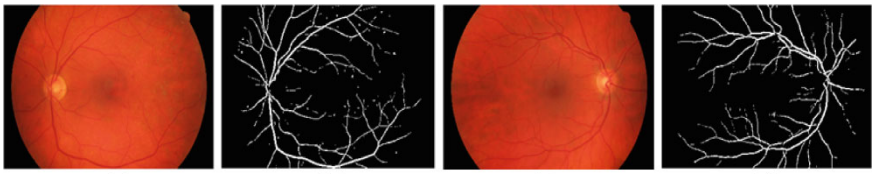


Fig. 7 Blood vessel segmentation results from HRF dataset. **a**, **c** Original retinal fundus image **b**, **d** segmented blood vessel image

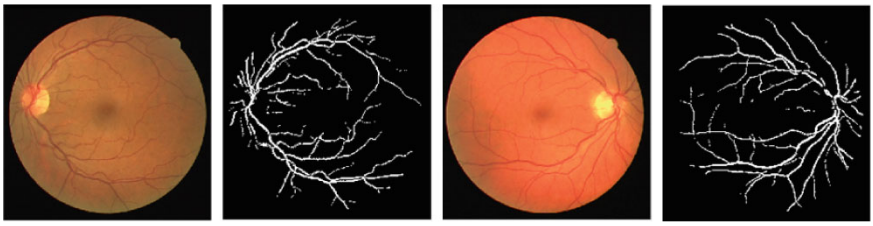


Fig. 8 Blood vessel segmentation results from DRIVE dataset. **a**, **c** Original retinal fundus image **b**, **d** segmented blood vessel image

4.3 Performance Evaluation

Proposed system performance was evaluated on HRF and DRIVE datasets with evaluation metrics of accuracy and specificity. This measurement was calculated based on predicted blood vessel segmentation versus ground truth given by experts. Results of this performance evaluation measures show the effectiveness and competence of our systems. The performances metrics of accuracy and specificity are evaluated through pixel classification and are explained below:

Accuracy is defined as the sum of correctly segmented blood vessel pixel and non-blood vessel pixel to the sum of total number of pixel and is presented in Eq. (1).

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{1}$$

Specificity is defined as sum of correctly segmented blood vessel pixel to the sum of total number of ground truth non-blood vessel pixel and is presented in Eq. (2).

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \tag{2}$$

where TP represents correctly identified blood vessel pixels, TN represents correctly identified non-blood vessel pixels, FP represents incorrectly identified non-blood vessel pixels as blood vessel pixels, and FN represents incorrectly identified blood

vessel pixels as non-blood vessel pixels. Table 1 shows the performance results of DRIVE and HRF dataset in terms of accuracy, specificity and time taken for of an image. Table 2 shows the comparative analysis of proposed system with the state-of-the-art methods. Out of all the existing techniques Samuel et al. [15] approach had highest accuracy of 96.09% on DRIVE dataset, and also, the accuracy of HRF dataset was 85.31% which is very low comparing to our approaches. Hence, it proves the low-level stability and inconsistency of the existing system. Our system overcomes all the above problems and provides better result for both the datasets represents the effectiveness of our algorithm, and it is more suitable in the field of automatic computerized segmentation of blood vessels with less time.

Table 1 Performance results of HRF and DRIVE datasets

Dataset	Total images	Accuracy (%)	Specificity (%)	Time/Image (s)
DRIVE	40	95.51	99.14	1.4
HRF	45	95.67	98.33	7.5

Table 2 Comparative analysis of proposed method with the state-of-the-art methods

No.	Method	Dataset	Accuracy (%)	Specificity (%)
1	Wang et al. [14]	DRIVE	95.41	98.17
2	Bandara et al. [16]	DRIVE	94.11	96.66
3	Sabaz et al. [17]	DRIVE	86.04	97.60
4	Kasmin et al. [18]	DRIVE	94.49	–
5	Xue et al. [19]	DRIVE	94.65	98.61
6	Ammar et al. [20]	DRIVE	94.64	97.86
7	Ali et al. [21]	DRIVE	94.46	96.88
8	Oliveira et al. [22] 1.Weighted mean using ORSF 2.Weighted mean using FCM 3.Median ranking	DRIVE	93.56 94.02 94.64	95.25 94.31 95.56
9	Ganesan et al. [23]	DRIVE	94.49	98.11
10	Memari et al. [24]	DRIVE HRF	93.40 94.94	96.93 96.69
11	Samuel et al. [15]	DRIVE HRF	96.09 85.31	97.38 85.23
12	Christodoulidis et al. [25]	HRF	94.79	95.82
13	Shukla [26]	DRIVE	94.76	98.36
14	Proposed method	DRIVE HRF	95.51 95.67	99.14 98.33

5 Conclusion and Future Work

In this paper presented automated image processing methods for blood vessel segmentation using image enhancement, gray-level thresholding and binarization. For this work, retinal images are not resized, original size of fundus images of DRIVE and HRF datasets is used for blood vessel segmentation, and achieved good accuracy shows our algorithm efficiency and suitable for different sizes of images. Image enhancement technique is adopted to increase the visibility of thin blood vessels during segmentation. For automation process, image binarization threshold level is automatically calculated from enhanced images using gray-level thresholding algorithm. After segmentation of blood vessel, the results of all the fundus images were compared with the ground truth provided by the experts and calculated performance metrics of accuracy and specificity. This method achieved good performance with less processing time that makes a handy early prediction tool of diabetic retinopathy. Our system achieved highest specificity of 99.14 and 98.33% on DRIVE and HRF dataset, and also, achieved overall accuracy of 95.59% represents the individuality of our approach and most suitable for disease diagnosis [27].

Proposed system achieved less computation time on DRIVE dataset whereas little high in HRF dataset because of its high resolution size. This algorithm was also performed on MESSIDOR dataset which contains 1200 retinal fundus images; however, the results cannot be evaluated due to lack of ground truth for blood vessel segmentation. In order to overcome this, our future work will be evaluated using high amount of datasets for diabetic retinopathy grading through automatic blood vessel segmentation. In future work, we also planned to use this segmentation work in the process of glaucoma disease identification and also retinal abnormalities like retinal artery occlusion and retinal vein occlusion.

Acknowledgements This work was supported by Centre For Research, Anna University, under the Anna Centenary Research Fellowship, Anna University, Chennai, India (Reference: CFR/ACRF/2018/AR1).

References

1. Jebaseeli TJ, Deva Durai CA, Peter JD (2019) Retinal blood vessel segmentation from diabetic retinopathy images using tandem PCNN model and deep learning based SVM. Optik 199(March 2018):163328
2. Câmara Neto L, Ramalho GLB, Rocha Neto JFS, Veras RMS, Medeiros FNS (2017) An unsupervised coarse-to-fine algorithm for blood vessel segmentation in fundus images. Expert Syst Appl 78:182–192
3. Vostatek P, Claridge E, Uusitalo H, Hauta-Kasari M, Fält P, Lensu L (2017) Performance comparison of publicly available retinal blood vessel segmentation methods. Comput Med Imaging Graph 55:2–12
4. Jyotiprava Dash NB (2017) A method for blood vessel segmentation in retinal images using morphological reconstruction. IEEE xplore

5. Wankhede PR, Khanchandani KB (2020) Feature extraction in retinal images using automated methods. Int J Sci Technol Res 9(3):5326–5333
6. Imran A, Li J, Pei Y, Yang J-J, Wang Q (2019) Comparative analysis of vessel segmentation techniques in retinal images. IEEE Access 7:114862–114887
7. Moccia S, De Momi E, El S, Mattos LS (2018) Blood vessel segmentation algorithms—review of methods, datasets and evaluation metrics. Comput Meth Prog Biomed 158:71–91
8. Sharma A, Rani S (2016) An automatic segmentation and detection of blood vessels and optic disc in retinal images. In: International conference on communication and signal processing, ICCSP 2016, pp 1674–1678
9. Mano A. Contrast enhanced superpixel based segmentation of retinal images
10. Shaik M (2016) Blood vessel segmentation in retinal images using multiscale image enhancement. Local Entropy Thresh Cluster 1(1):38–43
11. Biswal B, Pooja T, Subrahmanyam NB (2018) Robust retinal blood vessel segmentation using hybrid active contour model. IET Image Proc 12(3):389–399
12. Borah KJ, Borbhuyan AA, Rahman SS, Baruah A (2016) Wavelet based blood vessel segmentation in retinal image analysis. Int J Eng Res Technol V5(11):324–327
13. Ramani RG, Shanthamalar JJ (2020) Improved image processing techniques for optic disc segmentation in retinal fundus images. Biomed Sig Process Control 58:101832
14. Wang X, Jiang X, Ren J (2019) Blood vessel segmentation from fundus image by a cascade classification framework. Pattern Recogn 88:331–341
15. Samuel PM, Veeramalai T (2019) Multilevel and multiscale deep neural network for retinal blood vessel segmentation. Symmetry 11(7)
16. Bandara AMRR, Giragama PWGRMPB (2018) A retinal image enhancement technique for blood vessel segmentation algorithm. In: 2017 IEEE international conference on industrial and information systems, ICIIS 2017—Proceedings, vol 2018, Janua, pp 1–5
17. Sabaz F, Atila U (2017) ROI detection and vessel segmentation in retinal image. Int Arch Photogram Remote Sens Spatial Inf Sci—ISPRS Arch 42(4W6):85–89
18. Kasmin F, Abdullah A, Prabuwono AS (2017) Retinal blood vessel segmentation using ensemble of single oriented mask filters. Int J Electr Comp Eng 7(3):1414–1422
19. Xue LY, Lin JW, Cao XR, Yu L (2018) Retinal blood vessel segmentation using saliency detection model and region optimization. J Algo Comput Technol 12(1):3–12
20. Curran K, Souahlia A, Belatreche A, Benyettou A (2017) Blood vessel segmentation in retinal images using echo state networks. In: 2017 Ninth international conference on advanced computational intelligence (ICACI), pp 1–8
21. Ali A, Zaki WMDW, Hussain A (2019) Retinal blood vessel segmentation from retinal image using B-COSFIRE and adaptive thresholding. Indo J Electr Eng Comp Sci 13(3):1199–1207
22. Oliveira WS, Teixeira JV, Ren TI, Cavalcanti GDC, Sijbers J (2016) Unsupervised retinal vessel segmentation using combined filters. PLoS ONE 11(2):1–21
23. Ganesan K, Naik G, Adapa D, Raj ANJ, Alisetti SN, Zhuang Z (2020) A supervised blood vessel segmentation technique for digital fundus images using Zernike moment based features. PLoS ONE 15(3):1–23
24. Memari N, Ramli AR, Bin Saripan MI, Mashohor S, Moghbel M (2019) Retinal blood vessel segmentation by using matched filtering and fuzzy c-means clustering with integrated level set method for diabetic retinopathy assessment. J Med Biol Eng 39(5):713–731
25. Christodoulidis A, Hurtut T, Ben Tahar H, Cheriet F (2016) A multi-scale tensor voting approach for small retinal vessel segmentation in high resolution fundus images. Comput Med Imag Graphics 52:28–43
26. Shukla AK, Pandey RK, Pachori RB (2020) A fractional filter based efficient algorithm for retinal blood vessel segmentation. Biomed Sig Process Control 59(1–16):101883
27. Cheng Y, Ma M, Zhang L, Jin CJ, Ma Li, Zhou Y (2020) Retinal blood vessel segmentation based on densely connected U-Net. Math Biosci Eng 17:3088–3108

Security of Digital Images Based on SHA-3 and Multiplication-Rotation-XOR (MRX) Operations over Galois Field



Abdul Gaffar and Anand B. Joshi

Abstract In this paper, we propose a technique for securing digital images of different formats, namely binary, grayscale, index, and the true color images, based on the Secure Hash Algorithm 3 (SHA-3) and Multiplication-Rotation-XOR (MRX) operations over the Galois field. The technique performs encryption and decryption in the Cipher Block Chaining (CBC) mode of operation. SHA-3 is used for generating 256-bit message digest, to be used in constructing Initialization Vector (IV). Each plain image block is flipped horizontally and vertically, and then rows are updated via XOR and rotation (by random bits) operations. Finally, each block is pre-and-post multiplied by secret key over the Galois field. Several performance evaluation metrics are used on the standard images to empirically evaluate the efficiency and robustness of the designed approach. Moreover, a thorough comparison is also made with the recent state-of-the-art techniques.

Keywords Image encryption · CBC mode · SHA-3 · MRX operations · Galois field

1 Introduction

Recently, several methods/techniques have been designed for the security of digital images. Rehman et al. [1] in 2021 designed a method for securing gray images built on confusion-diffusion using multiple chaotic maps. The statistical results prove that the method is resistant to the noise attack. However, the images enciphered by the proposed approach do not attain the ideal entropy value, and also, the approach does not resist the differential attack.

Ibrahim and alharbi [2] in 2020 proposed a technique for the security of gray images, which is based on the Elliptic Curve Cryptography (ECC), dynamic S-boxes,

Supported by UGC-IN under the grant number [415024].

A. Gaffar (✉) · A. B. Joshi
Department of Mathematics and Astronomy, University of Lucknow, 226 007 Lucknow, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_28

and the Henon map. The empirical results show that the technique is insecure to the differential attack.

Hasanzadeh and Yaghoobi [4] in 2019 proposed a method for securing true color images based on a substitution box and a hyperchaotic system utilizing fractional keys. The encryption evaluation metrics demonstrate that the enciphered images do not attain the ideal entropy value, and moreover, the method does not resist the differential attack. Hayat and Nazam [5] in 2019 proposed an algorithm for gray image encryption based on the ECC. The scores of the differential analysis show that the algorithm does not attain the required diffusion property, essential for resisting the differential attack. Rehman et al. [6] in 2019 designed a technique for the encryption of gray images based on two-fold operations using chaos. The results of encryption evaluation metrics verify that the technique fails to attain the diffusion property, and also, the cipher images fail to attain the ideal entropy value.

It is worth to note that the mentioned algorithms/methods are either vulnerable to the differential attack, or fail to turn (or encrypt) a plain image into a noisy-like image. Moreover, the above methods are designed merely for securing/encrypting true color or grayscale *square* images. None of the methods consider index or binary images. The proposed method, which is based on SHA-3 and MRX operations over the Galois field, is an attempt to secure the binary, grayscale, index, and the true color images, simultaneously. Also, the proposed method considers a *non-square*-sized image.

The rest of the paper is organized as follows: Sect. 2 introduces preliminaries, which consists of SHA-3 and the Galois field. Section 3 describes the proposed technique. Section 4 contains implementation and experimental results. Section 5 provides statistical and security analyses. Section 6 gives comparison results, while Section 7 summarizes the paper.

2 Preliminaries

2.1 SHA-3

A hash function is a mathematical function that accepts arbitrary size input, but producing only fixed size output also known as (a.k.a.) message digest. SHA-3 [8] is a recent member of the SHA family and can produce message digests of size 224 bits, 256 bits, 384 bits, or 512 bits. According to the message digest size, it is denoted as SHA-3-224, SHA-3-256, SHA-3-384, or SHA-3-512, respectively. We utilize SHA-3-256 version for constructing IV to be used in the CBC mode of operation (see Fig. 1).

2.2 Galois Field

In the proposed technique, the Galois field [9] is utilized for finding the multiplication of two 4×4 block matrices, or to calculate the multiplicative inverse of a block matrix. The utilized Galois field is given by (1):

$$GF(2^8) = \frac{GF(2)}{< (x^8 + x^6 + x^5 + x + 1) >} \tag{1}$$

where the irreducible polynomial $x^8 + x^6 + x^5 + x + 1 = f(x)$ is of degree 8 over $GF(2)$.

3 Description of the Proposed Technique

3.1 Preprocessing on Secret Key and SHA-3-256

Input: Secret key $K = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}\}$ (16-byte) and SHA-3-256.

1. *Constructing the matrix of secret key*:
   Reshape K, a 16-byte key array into a 4×4 matrix (column-wise), given by (2):

$$K_m = \begin{pmatrix} q_1 & q_5 & q_9 & q_{13} \\ q_2 & q_6 & q_{10} & q_{14} \\ q_3 & q_7 & q_{11} & q_{15} \\ q_4 & q_8 & q_{12} & q_{16} \end{pmatrix} \tag{2}$$

2. *Construction of IV*:
   (a) Let P be a plain image of size $R \times C$. Then apply SHA-3-256 on the plain image P, which gives a unique message digest of size 256-bit, i.e., 32 bytes. Let the message digest $H = \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}, h_{12}, h_{13}, h_{14}, h_{15}, h_{16}, h_{17}, h_{18}, h_{19}, h_{20}, h_{21}, h_{22}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}, h_{28}, h_{29}, h_{30}, h_{31}, h_{32}\}$.
   (b) Now, append every two consecutive hash bytes h_i's of H as: $H = \{h_1h_2, h_3h_4, h_5h_6, h_7h_8, h_9h_{10}, h_{11}h_{12}, h_{13}h_{14}, h_{15}h_{16}, h_{17}h_{18}, h_{19}h_{20}, h_{21}h_{22}, h_{23}h_{24}, h_{25}h_{26}, h_{27}h_{28}, h_{29}h_{30}, h_{31}h_{32}\}$.
   (c) Arrange H (column-wise) as a 4×4 matrix:

$$H_m = \begin{pmatrix} h_1h_2 & h_9h_{10} & h_{17}h_{18} & h_{25}h_{26} \\ h_3h_4 & h_{11}h_{12} & h_{19}h_{20} & h_{27}h_{28} \\ h_5h_6 & h_{13}h_{14} & h_{21}h_{22} & h_{29}h_{30} \\ h_7h_8 & h_{15}h_{16} & h_{23}h_{24} & h_{31}h_{32} \end{pmatrix}$$

(d) Compute IV to be used in (12) and (13) as: IV $= H_m \pmod{K_m}$, i.e.,

$$IV = \begin{pmatrix} h_1 h_2 \pmod{q_1} & h_9 h_{10} \pmod{q_5} & h_{17} h_{18} \pmod{q_9} & h_{25} h_{26} \pmod{q_{13}} \\ h_3 h_4 \pmod{q_2} & h_{11} h_{12} \pmod{q_6} & h_{19} h_{20} \pmod{q_{10}} & h_{27} h_{28} \pmod{q_{14}} \\ h_5 h_6 \pmod{q_3} & h_{13} h_{14} \pmod{q_7} & h_{21} h_{22} \pmod{q_{11}} & h_{29} h_{30} \pmod{q_{15}} \\ h_7 h_8 \pmod{q_4} & h_{15} h_{16} \pmod{q_8} & h_{23} h_{24} \pmod{q_{12}} & h_{31} h_{32} \pmod{q_{16}} \end{pmatrix} \quad (3)$$

where "mod" is the modulo operation and q_is are the key bytes of the secret key K.

Output: Secret key matrix K_m and the initialization vector IV.

3.2 Block Substitution

Input: Plain image block L of size 4×4.

1. Perform horizontal and vertical flipping on L as:
   (a) *Horizontal flip*: If $L = [b_{ij}]_{4\times4}$, $i, j = 1, 2, 3, 4$ and F_h denotes horizontal flip, then $L_h = F_h(L)$, where action of F_h on L is defined by (4):

$$F_h([b_{ij}]) = [b_{i(5-j)}]_{4\times4} \quad (4)$$

   Note that, inverse of F_h is itself.
   (b) *Vertical flip*: If $L_h = [b_{rs}]_{4\times4}$, $r, s = 1, 2, 3, 4$ and F_v denotes vertical flip, then $L_v = F_v(L_h)$, where action of F_v on L_h is defined by (5):

$$F_v(L_h) = [b_{(5-r)s}]_{4\times4} \quad (5)$$

   Note that, inverse of F_v is itself.

   Note that, steps 1a and 1b together can be written as: $L_v = F_v \circ F_h(L)$, where symbol $\circ$ denotes a composite operation.
2. Perform the following operations on the flipped block L_v:
   (a) If $L_v = [c_{ij}]_{4\times4}$, then rows of block L_v can be denoted as: $R_1 = [c_{11}, c_{12}, c_{13}, c_{14}] = [c_{1j}]$, $R_2 = [c_{2j}]$, $R_3 = [c_{3j}]$, and $R_4 = [c_{4j}]$. In row notation, $L_v = [R_1, R_2, R_3, R_4]^t$, t denotes the transpose.
   (b) Now, update the rows R_i's of block L_v as:

$$\left.\begin{aligned} S_1 &= R_4 \oplus [(R_1 \oplus R_2) \lll (\bmod q_1)] \\ S_2 &= S_1 \oplus [(R_2 \oplus R_3) \lll (\bmod q_2)] \\ S_3 &= S_2 \oplus [(R_3 \oplus R_4) \lll (\bmod q_3)] \\ S_4 &= S_3 \oplus [(R_4 \oplus S_1) \lll (\bmod q_4)] \end{aligned}\right\} \quad (6)$$

where $\oplus$ denotes XOR (exclusive-OR) operation between the pixels of two rows, while $R_i \lll (\text{mod } q_i)$ denotes the left rotation of pixels of R_i by mod (q_i) bits, where $i = 1, ..., 4$.
So, the updated block $L_{upd} = Upd(L_v) = [S_1, S_2, S_3, S_4]^t$, where the operation Upd is defined by (6).
Note that, $L_v = (Upd)^{-1}(L_{upd})$, where the inverse operation $(Upd)^{-1}$ on L_{upd} is defined by (7):

$$\left.\begin{aligned} R_4 &= [(S_4 \oplus S_3) \ggg (\text{mod } q_4)] \oplus S_1 \\ R_3 &= [(S_3 \oplus S_2) \ggg (\text{mod } q_3)] \oplus R_4 \\ R_2 &= [(S_2 \oplus S_1) \ggg (\text{mod } q_2)] \oplus R_3 \\ R_1 &= [(S_1 \oplus R_4) \ggg (\text{mod } q_1)] \oplus R_2 \end{aligned}\right\} \tag{7}$$

where $\ggg$ denotes the right rotation.
Thus, we obtained $L_v = [R_1, R_2, R_3, R_4]^t$.

3. Now, pre-and-post multiply L_{upd} by the secret key matrix K_m as (8) shows:

$$Mult(L_{upd}) = (K_m * L_{upd} * K_m)_{2^8} \tag{8}$$

where $*$ denotes matrix multiplication and is evaluated over $GF(2^8)$ [see (1)].
We denote (8) as: $L_{Sub} = Mult(L_{upd})$, where the action of $Mult$ on L_{upd} is defined by (8).
Note that, $L_{upd} = (Mult)^{-1}(L_{Sub})$, where the inverse operation $(Mult)^{-1}$ on L_{Sub} is given by (9):

$$(Mult)^{-1}(L_{Sub}) = ({K_m}^{-1} * L_{Sub} * {K_m}^{-1})_{2^8} \tag{9}$$

where ${K_m}^{-1}$ denotes inverse of K_m over the Galois field $GF(2^8)$.
4. Note that, (4), (5), (6), and (8) can be compositely written as:

$$L_{Sub} = Mult \circ Upd \circ F_v \circ F_h(L)$$

Let $SUB = Mult \circ Upd \circ F_v \circ F_h$, then

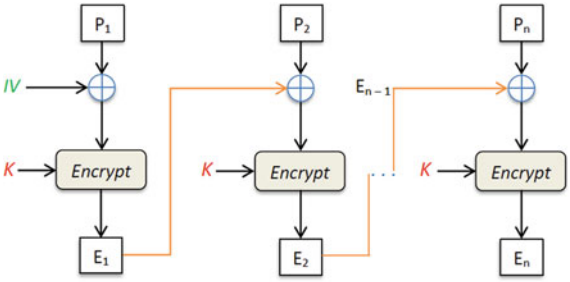
$$L_{Sub} = SUB(L) \tag{10}$$

The inverse of (10) is given by:

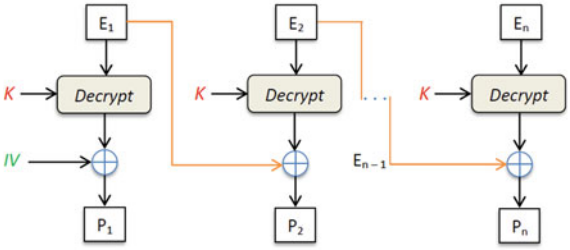
$$L = (SUB)^{-1}(L) \tag{11}$$

where $(SUB)^{-1}$ is given as:

$$(SUB)^{-1} = {F_h}^{-1} \circ {F_v}^{-1} \circ (Upd)^{-1} \circ (Mult)^{-1}$$



(a) Encryption



(b) Decryption

Fig. 1 CBC mode of operation: **a** Encryption and **b** Decryption.

where $F_h{}^{-1}$, $F_v{}^{-1}$, Upd^{-1}, and $(Mult)^{-1}$ are given by (4), (5), (7), and (9), respectively.

Output: Substituted block L_{Sub}.

It is noticeable that encryption (*Encrypt*) and decryption (*Decrypt*) in the CBC mode of operation (see Fig. 1) means the block substitution (see Sect. 3.2), given by (10) and (11), respectively.

3.3 Encryption Procedure

Input: Original image P of size $R \times C$ and the secret key K.

1. Decompose the original image P into 4×4 blocks (non-overlapping) P_j's, for $j = 1, 2, ..., RC/16$.
2. Note that, if the image size $R \times C$ is indivisible by 4×4, the block size, then pad $(4 - r)$ rows and $(4 - c)$ columns with zeros (or nonzero, if the image is a null image), where r and c denote remainders of $R/4$ and $C/4$, respectively.
3. Perform encryption in the CBC mode of operation (see Fig. 1a) as:

   (a) For the first block P_1, i.e., for $j = 1$:

$$\left.\begin{array}{l} L = P_1 \oplus IV \\ E_1 = SUB(L) \end{array}\right\} \tag{12}$$

(b) For rest of the blocks P_j's, $j \in \{2, ..., RC/16\}$:

$$\left.\begin{array}{l} \text{for } j = 2 : RC/16 \\ \quad L_j = E_{j-1} \oplus P_j \\ \quad E_j = SUB(L_j) \\ \text{end for} \end{array}\right\}$$

where IV is given by (3), second line of (12) denotes block substitution (SUB) given by (10), and rest of the symbols have their usual meanings.

4. Concatenate all the encrypted blocks E_j's, for j =1, 2,..., $RC/16$, to form an encrypted image, say, E, of size $R \times C$.

Output: Encrypted image E of size $R \times C$.

3.4 Decryption Process

Input: Encrypted image E of size $R \times C$ and the secret key K.

1. Decompose the encrypted image E into 4×4 blocks (non-overlapping) E_j's, for $j = 1, 2, ..., RC/16$.
2. Perform decryption in the CBC mode of operation (see Fig. 1b) as:

(a) For the first block E_1, i.e., for $j = 1$:

$$\left.\begin{array}{l} L = (SUB)^{-1}(E_1) \\ P_1 = L \oplus IV \end{array}\right\} \tag{13}$$

(b) For rest of the blocks E_j's, $j \in \{2, 3, ..., RC/16\}$:

$$\left.\begin{array}{l} \text{for } j = 2 : RC/16 \\ \quad L_j = (SUB)^{-1}(E_j) \\ \quad P_j = L_j \oplus E_{j-1} \\ \text{end for} \end{array}\right\}$$

where first line of (13) denotes inverse of the substitution process $(\text{SUB})^{-1}$, given by (11), and rest of the symbols have their usual meanings.

(c) Concatenate all the decrypted blocks P_j's, for $j = 1$, 2,..., $RC/16$, to form a decrypted (original) image, say, P, of size $R \times C$.

Output: Original image P of size $R \times C$.

4 Implementation and Experimental Results

The designed approach is implemented in the software *MATLAB* (version R2019b). To evaluate the encryption quality of the proposed method, a number of test images of different formats are considered from the standard databases. In particular, grayscale (Figs. 3a and 3b of sizes 256 ×256 and 512 × 512 pixels, respectively) and true color (Fig. 3c of size 512 × 512) images are chosen from the image database called as USC-SIPI,[1] index image (Fig. 3d of size 256 × 256) is taken from the *MATLAB* IPT database,[2] while binary image (Fig. 3e of size 257 × 383) is taken from the *MPEG7* database.[3] The results of the encryption are shown by Fig. 3k–o.

5 Statistical and Security Analyses

5.1 Key Space Analysis

The key space should be large enough in order to resist many of the statistical attacks such as brute force, chosen/known plain-text. Since, in the proposed technique, SHA-3-256 generates a 32-byte output and the secret key is of 16-byte, so the size of our key space is 2^{256} ($\approx 10^{77.06}$), which is sufficient for resisting the most of the known attacks.

5.2 Key Sensitivity Analysis

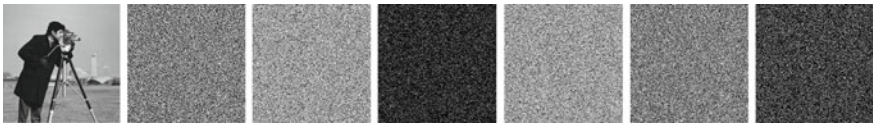
It is a core building block of any secure cryptographic method. Without the strong key sensitivity concept, none of the cryptographic algorithms is believed to be secure. Moreover, it is responsible for creating the property of confusion [10]. Here, we analyze the robustness of our designed encryption and decryption algorithms via altering merely single bit in the key K. The results of the key sensitivity analysis are shown in Fig. 2, wherein Fig. 2a is the original image (P) and Fig. 2b is the encryption (Enc) of P using K. Figure 2c is the encryption of P using modified key λ, where λ is constructed from K via changing merely LSB of the first byte of K. Figure 2e is the difference (diff.) between the cipher images. Figures 2i, j are the decryption (Dec) of the encrypted image E and E_1 using λ and K, respectively. Figure 2k is the difference between the deciphered images.

On observing the difference images of the encryption and decryption, we conclude that the proposed approach possesses the required property of confusion.

[1] Available at, http://sipi.usc.edu/database/.

[2] Available at, https://in.mathworks.com/matlabcentral/fileexchange/22089-iptimages.

[3] Available at, http://www.imageprocessingplace.com/root_files_V3/image_databases.htm.



(a) Original image (P) (b) $E = Enc(P, K)$ (c) $E_1 = Enc(P, \lambda)$ (d) Diff. $|E_1 - E|$ (e) $D = Dec(E, \lambda)$ (f) $D_1 = Dec(E_1, K)$ (g) Diff. $|D_1 - D|$

Fig. 2 Results of the key sensitivity analysis

5.3 Histogram Analysis

An image histogram is a very important tool for analyzing the distributions of pixels of a plain/cipher image. As secure encryption method turns an intelligible/original image into an unintelligible (random-like) image, so the pixels of an encrypted image should be uniformly distributed. The histogram results of the original and the encrypted images are shown in Fig. 3, whence we notice that the histograms of the encrypted images (Fig. 3k–o) are uniform, unlike to that of original images (Fig. 3a–e), which have non-uniform histograms, and thus, the images encrypted by the proposed method are random-like images.

5.4 Information Entropy Analysis

Entropy test is used to calculate the randomness of an image, say, I, and can be computed via (14):

$$\mathrm{Ent}(I) = -\sum_{r=1}^{2^N} \Pr(e_r) \log_2(\Pr(e_r)) \tag{14}$$

where e_r represents the rth potential value in I and $\Pr(e_r)$ represents the probability of e_r.

$\mathrm{Ent}(I)$ attains its ideal value, which is 8, when the pixels of the image I are uniformly distributed. The values of entropy for the original and encrypted images are provided in Table 1.

5.5 NPCR and UACI Tests

Number of Pixel Change Rate (NPCR) and the Unified Averaged Changed Intensity (UACI) [11] are the two commonly utilized tests for analyzing the resistance of the differential attack [12] or judging Shannon's diffusion property [10]. The scores of

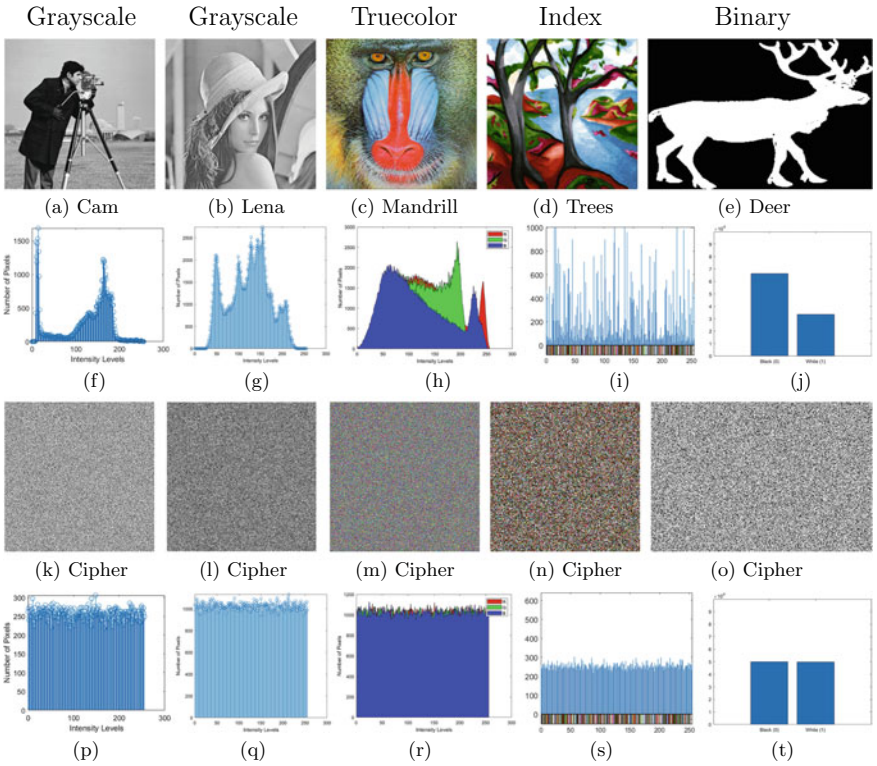


Fig. 3 Histogram results: **a–e** are the original images; **f–j** are the histograms of the original images; **k–o** are the cipher images; and **p–t** are the histograms of the cipher images

these metrics between the encrypted images E_1 and E_2 can be calculated via (15) and (16), respectively:

$$\text{NPCR} = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{\beta(i,j)}{RC} \times 100\% \tag{15}$$

$$\text{UACI} = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{|E_1(i,j) - E_2(i,j)|}{F_p \cdot RC} \times 100\% \tag{16}$$

where $\beta(i,j) = 0$ if $E_1(i,j) = E_2(i,j)$, else $\beta(i,j) = 1$; $E_1(i,j)$ and $E_2(i,j)$ are the pixels of the encrypted images prior-and-after alteration of only 1-bit in a pixel of the original image; and $F_p = 2^N - 1$, the maximum pixel value of an image. The calculated NPCR and UACI scores are shown in Table 1. If the theoretical NPCR score is less than that of reported/calculated score, which is 99.5527 at 0.01 significance

Table 1 Results of comparison based on the entropy values, NPCR, and UACI test scores

Method	Image	Size (in pixels)	Entropy		NPCR	UACI
			Original	Encrypted		
Proposed	Cameraman	256 × 256	7.0097	7.9978	99.64	33.36
	Lena	512 × 512	7.4451	7.9993	99.60	33.53
	Mandrill	512 × 512	7.7624	7.9998	99.61	33.50
	Trees	256 × 256	7.2697	7.9974	99.64	33.62
	Deer	257 × 383	0.9204	7.9984	99.60	33.45
Rehman and Shafique [1]	Lena	512 × 512	7.4451	7.9971	99.37	33.97
Hasanzadeh and Yaghoobi [4]	Lena	512 × 512	7.4451	7.8836	99.30	32.43
Hayat and Azam [5]	Lena	512 × 512	7.4451	7.9992	98.98	31.01
Rehman et al. [6]	Lena	512 × 512	7.4451	7.9972	99.5483	33.3055
Rehman et al. [7]	Lena	512 × 512	7.4451	7.9971	99.5548	33.4234
Ibrahim and Alharbi [2]	Cameraman	256 × 256	7.0097	7.9973	99.6086	33.4409
Ismail et al. [13]	Cameraman	256 × 256	7.0097	7.9972	76.1681	24.2534

level and 99.5693% at 0.05 level, then the encryption method is said to *pass* the NPCR test.

If the calculated UACI score lies in (33.2225, 33.7016)% for 0.01 significance level and (33.2824, 33.6447)% for 0.05 level, then the method is said to *pass* the UACI test. The suggested technique passes both the tests, and thus, ensures the property of diffusion, and also, outperforms the methods listed in Table 1, which are vulnerable to the differential attack.

6 Comparison with the Existing Methods

The designed encryption method is compared with the recent state-of-the-art schemes based on the frequently available encryption evaluation metrics, namely entropy value, NPCR, and the UACI test scores, for empirically assessing the proposed technique. The comparison results are shown in Table 1, whence we conclude that the proposed technique outperforms the compared techniques.

7 Conclusion

In this paper, an attempt has been made to improve the security of the digital images of different formats, namely binary, grayscale, index, and the true color images, based on SHA-3-256 and MRX operations over the Galois field. Several statistical and security evaluation metrics are utilized in order to experimentally assess the robustness and efficiency of the designed scheme. Moreover, the designed approach is thoroughly compared with the recent state-of-the-art methods, and the results of comparison prove that our approach is better than the compared approaches.

References

1. Rehman A, Shafique A (2021) Dynamic substitution and confusion-diffusion-based noise-resistive image encryption using multiple chaotic maps. IEEE Access 9:52277–52291. https://doi.org/10.1109/ACCESS.2021.3069591
2. Ibrahim S, Alharbi A (2020) Efficient image encryption scheme using Henon map, dynamic S-boxes, and elliptic curve cryptography. IEEE Access 8:194289–194302. https://doi.org/10.1109/ACCESS.2020.3032403
3. Mondal B, Singh S, Kumar P (2019) A secure image encryption scheme based on cellular automata and chaotic skew tent map. J Inf Secur Appl 45:117–130. https://doi.org/10.1016/j.jisa.2019.01.010
4. Hasanzadeh E, Yaghoobi M (2019) A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys. Multimedia Tools Appl 1–19. https://doi.org/10.1007/s11042-019-08342-1
5. Hayat U, Azam NA (2019) A novel image encryption scheme based on an elliptic curve. Signal Process 155:391–402. https://doi.org/10.1016/j.sigpro.2018.10.011
6. Rehman A, Xiao D, Kulsoom A, Hashmi MA, Abbas SA (2019) Block mode image encryption technique using two-fold operations based on chaos, MD5 and DNA rules. Multimedia Tools Appl 78(7):9355–9382. https://doi.org/10.1007/s11042-018-6516-1
7. Rehman A, Liao X, Hahsmi MA, Haider R (2018) An efficient mixed inter-intra pixels substitution at 2 bits-level for image encryption technique using DNA and chaos. Optik 153:117–134. https://doi.org/10.1016/j.ijleo.2017.09.099
8. FIPS PUB 202: SHA-3 standard: Permutation-based hash and extendable Output functions, http://dx.doi.org/10.6028/NIST.FIPS.202. Accessed 01 July 2021
9. Galois Field. Encyclopedia of Mathematics. EMS Press, http://encyclopediaofmath.org/index.php?title=Galois_field&oldid=34238. Accessed 01 July 2021
10. Shannon CE (1949) Communication theory of secrecy systems. Bell Syst Tech J 28:656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x
11. Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. J Selected Areas Telecommun (JSAT), 31–38
12. Biham E, Shamir A (1993) Differential cryptanalysis of the data encryption standard. Springer, Berlin, Heidelberg
13. Ismail SM, Said LA, Radwan AG, Madian AH, Abu YMF (2020) A novel image encryption system merging fractional-order edge detection and generalized chaotic maps. Signal Process 167. https://doi.org/10.1016/j.sigpro.2019

Multimodal Depression Severity Detection Using Deep Neural Networks and Depression Assessment Scale



H. V. Madhu Sudhan and S. Saravana Kumar

Abstract Depression is a common mental disorder worldwide which not only impacts the sufferer, but also the people surrounding them. It is a mental illness with persistent depressed mood or losing interest in activities which may impact the daily life actions. If its long lasting with moderate or severe intensity and untreated, depression will lead to a serious health condition and even may lead to suicide. It is necessary to detect the disorder at an early stage and reduce the effects of disorder before it reaches for suicide. This paper presents a novel depression severity detection technique with multimodal data by using deep neural networks and Hamilton Depression Rating Scale. Individual modalities for video, speech, and text are detected using deep neural networks (DNN), and all the modalities are fused with corresponding weights to calculate the total points in Hamilton Depression Rating Scale (HDRS). Based on the points calculated, severity of the depression will be classified as no depression, mild, moderate, and severe depression. Performance is evaluated at individual modalities and results showed an accuracy of 66% through video modality, 81% for speech modality, and 82% for text modality to calculate the points for Hamilton Depression Rating Scale (HDRS) and to classify severity.

Keywords Depression detection · Deep neural networks · Multimodal emotion recognition · Hamilton Depression Rating Scale

1 Introduction

Depression is a common mental health disorder that affects worldwide. According to World Health Organization (WHO) [1], there are 264 million people affected worldwide for depression and around 800,000 people die to suicide every year. Suicide is mainly seen in 15–29 years old, and it is the second leading cause of death. Depressed people find it more difficult to concentrate on work, have problems in social communication and find it difficult to carry out daily activities, and are more

H. V. M. Sudhan · S. S. Kumar (✉)
Department of Computer Science and Engineering, CMR University (CMRU), Bangalore, Karnataka 560043, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_29

inclined to sadness, loneliness, hopelessness, anxiety, and disinterested [2]. Various factors like social, psychological, and biological with complex interactions result in depression. Depression normally comes when people undergo adverse life events like loss of job, partner, and psychological trauma. In low- and middle-income countries, between 76 and 85% receive no treatment for their disorder [3]. Some of the barriers for effective treatment are lack of resources and trained healthcare professionals and also the social stain which is related to depression. Other barriers include inaccurate assessment and mis-diagnosis of people of all income levels across countries and prescribing antidepressants to those who actually do not have the disorder and vice versa.

Detection of depression plays an important role as the current methods rely on clinical review of the patient through physically or virtually. Clinical review takes more time and is expensive and not reachable to people of all income levels. Affective sensing methods can assist physicians in the early stages of depression and in the subsequent monitoring. Through affective sensing techniques, people can assess themselves about severity of the depression initially before approaching the healthcare professionals by saving time, cost, and chances of mis-diagnosis. When a person is affected by depression, person expresses the intricate signs that can be captured by studying all the modalities. For example, when a person is asked with a question related to Hamilton Depression Rating Scale (HDRS), person may have less eye contact and displays sad emotion majority of the time during the interview that can be captured by video, low and depressed tone can be captured through speech and context of words used can be analyzed through the lexicon text. In some of the cases, people with severe depression may not show any facial emotions, but this can be captured by other modalities like speech and text by understanding the tone and the words used during communication. This paper discusses fusing of three modalities and calculating the severity of depression using Hamilton Depression Rating Scale (HDRS).

2 Related Work

Automatic depression detection has gained more attention recently in the affective sensing community. Lot of research has been done to understand the depression using individual modalities and multimodal approach by fusing the various modalities. Review of various works done using different modalities of video, speech, and text is discussed in this section.

2.1 Facial Emotion Analysis from Video

Various methods have been proposed to detect depression from videos and images. Active appearance model (AAM) [4, 5] was used to understand the facial features

which are further used to compute parameters like action units associated with depression by computing the mean duration, ratio of onset to total duration and ratio of offset to onset phase [6]. Kleinsmith and Bianchi-Berthouze [7] discuss to classify face into a set of emotions like happiness, sadness, and anger using facial expression recognition system and other studies like [8] focuses on facial expression recognition through individual muscle movements. Facial Action Coding System (FACS) is proposed in Bartlett et al. [9] to perform facial emotion recognition with the help of action units (AU). Bayesian networks, hidden Markov model (HMM), and neural networks were proposed in Ekman and Friesen [10] for facial expression recognition. Alizadeh and Fazel [11] used convolutional neural networks (CNN) for facial expression recognition using grayscale images. Combination of raw pixel data and histogram of oriented gradients (HOG) features were used to train a CNN model.

Real-time vision system which does the task of face detection, gender classification, and emotion recognition is proposed in Arriaga et al. [12] using CNN architecture and guided back-propagation technique. It showcased that regularization methods and visualization of previously hidden features are necessary to reduce the gap between slow performances and real-time architectures. Facial feature extraction for depression detection in students is proposed in Venkataraman and Parameswaran [13] which captures the facial features from video and extracts the facial features in each frame. Facial features are analyzed to detect the signs of depression. OpenFace [14], an interactive open-source facial expression analysis, is used in Jannat et al. [15] to extract features for face landmark regions, head pose, and eye gaze estimations and converts to a facial action unit. Features extracted from OpenFace and bag-of-visual words are used to train individual model per feature all having a single layer of 200 BLSTM hidden units, followed by a maxpooling and then learn a regressor to detect the depression. The current paper uses a CNN model with Xception [12, 16] architecture to identify the emotion from video which will be used to detect depression.

2.2 Emotion Analysis from Speech

Emotion analysis from speech plays a significant role in identifying signs of depression. Iqbal and Barua [14] propose an emotion recognition system using gradient boosting, K-nearest neighbors (KNN), and support vector machine (SVM) to classify emotion and also to identify differences based on gender. Another study proposed in Jannat et al. [15] achieved an accuracy of 66.41% on audio data and achieved 90% by mixing both audio and video. Three shared emotion recognition models for speech and song are proposed in Zhang et al. [16], namely a single task model, a single task hierarchical model, and multitask hierarchical model. de Pinto et al. [17] proposed an emotion classification method using deep neural networks (CNN). F1-score evaluation metric accuracy on the test set was 0.91, and "angry" emotion had an accuracy score of 0.95. Comparison of various speech emotion recognition techniques is studied in Venkataramanan and Rajjamohan [18]. Feature significance

such as mel-frequency cepstral coefficients (MFCCs), log-mel spectrogram, energy, and pitch were equated by using techniques such as convolutional neural networks (CNNs), long short-term memory (LSTM), hidden Markov models (HMMs), and deep neural networks (DNNs). Two-dimensional four-layer Convolutional Neural Network achieved an accuracy of 68% and results were impacted by choice of audio factors.

Attention-based fully convolutional network is proposed for speech emotion recognition [19] along with transfer learning to improve the accuracy, because of limited data. Proposed model outperformed with a weighted accuracy of 70.4% and an unweighted accuracy of 63.9%, respectively. Performance of two categories of models for speech emotion recognition is studied in Sahu [20]. Six traditional machine learning algorithms were trained using the extracted features in first method while in the second method, long short-term memory (LSTM)-based classifier and a feed-forward neural network are trained over the features. Study concludes that traditional machine learning techniques are able to achieve performance comparable with deep learning techniques. The current paper proposes emotion identification from speech using MFCC features and CNN.

2.3 Emotion Analysis from Text

Emotion analysis from text has been researched for quite a sometime in natural language processing community. Recently, deep learning techniques have been used for text classification. Personality traits and meta features such as age and gender could have positive impact on model performance while detecting depression from social media text [21]. Social media text is used to classify depression in Resnik et al. [22], and also other studies have been done to detect for various mental disorders. Recurrent neural network (RNN) with attention is used to detect social media posts resembling crisis [23]. Orabi et al. [24] showcased that using CNN produces better result when compared with RNN while detecting depression. Singh and Wang [25] aim to predict depression in tweets using RNN, gated recurrent unit (GRU) and CNN. Study examines the effect of character-based vs word-based models and pretrained embeddings vs learned embeddings. Best performing models are word-based GRU with 98% accuracy and word-based CNN with 97% accuracy. Hamilton [26] and Williams [27] use transformer encoders for detecting the emotions using bidirectional encoder representations from transformers (BERT). In the current research, BERT based emotion detector is used to predict the emotions in the text to classify for depression.

2.4 Hamilton Depression Rating Scale (HDRS)

To administer clinical depression, Ham-D is used as a depression assessment scale [26, 27]. $HDRS_{17}$ consists of 17 items relating to symptoms of the depression over the past week. Later $HDRS_{21}$ was introduced with four additional items for sub-categorizing the depression. Limitations of HDRS include atypical symptoms (e.g., hypersomnia, hyperphagia) are not assessed using the scale. For $HDRS_{17}$, a score of 0–7 is accepted as normal and a score >20 is usually required for entry into the clinical trial.

3 Proposed Methodology

This section presents the architecture and methods for depression severity detection system for all the modalities. In Sects. 3.1–3.3, individual modality is discussed in detail about the dataset, feature extraction, methodology, evaluation, and performance. In Sect. 3.4, fusion of all the modalities with Hamilton Depression Rating Scale (HDRS) is discussed. Figure 1 represents the technical architecture for depression severity detection system.

3.1 Facial Emotion Recognition Model

Facial emotion recognition is achieved with the help of a real-time facial classification model [12]. Model architecture is divided into two parts as shown in Fig. 2. First part of the model eliminates fully connected layer completely, and the second part combines the deleted fully connected layer and combines the depth-wise separable convolutions and residual modules. Both parts are trained using adaptive moment estimation (ADAM) optimizer [28].

Global average pooling is used to remove fully connected layers and it is achieved by having the same number of feature maps in the last convolutional layer as same number of classes and applying a softmax activation function to each feature map.

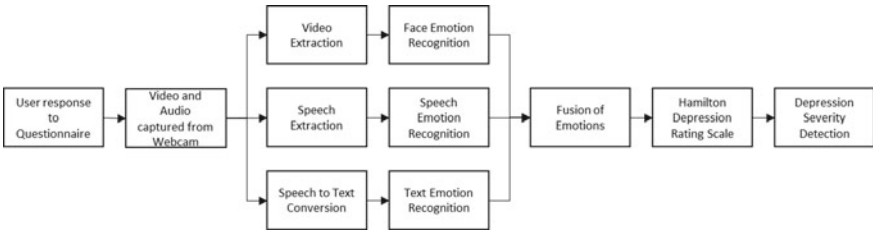


Fig. 1 Technical architecture for depression severity detection system

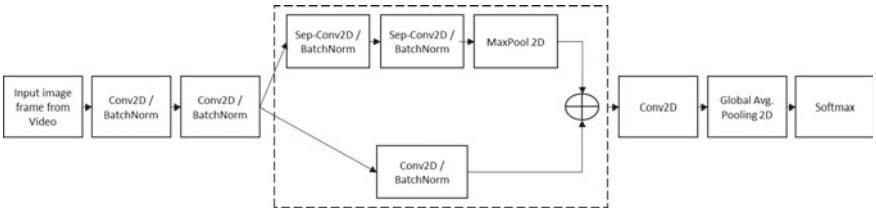


Fig. 2 Model architecture for facial emotion recognition

Model architecture has nine convolutional layers, rectified linear units (ReLUs), batch normalization and global average pooling. Dataset used to train the model was taken from FER-2013 dataset [29]. Dataset has 35,887 grayscale images, and it has classes for "angry," "disgust," "fear," "happy," "sad," "surprise," and "neutral." First model achieved an accuracy of 66% on the dataset. Second part of the architecture is based on Xception [30], and it combines the use of depth-wise separable convolutions [31] and residual modules [32]. Learned features become the difference of original feature map and the desired features and are obtained by residual modules which modify the desired mapping between two subsequent layers.

Final combined architecture consists of a fully convolutional network with four residual depth-wise separable convolutions, and each convolution is followed by a batch normalization and rectified linear unit (ReLU) activation function. Last layer has global average pooling and softmax activation function for the prediction. Final architecture gave an accuracy of 66% for the emotion classification task on FER-2013 dataset. Limitation includes the misclassification between sad, fear, angry, and disgust, but this limitation is not that impactful on current study as all the above emotions are signs of depression, and we could use collectively for calculating the HDRS score.

3.2 Speech Emotion Recognition Model

Speech emotion recognition model is derived from [17] which proposes an architecture using deep neural networks and mel-frequency cepstral coefficients (MFCC). MFCC is the only feature that is used to train the model using CNN and dense layers. In speech recognition, MFCC is considered as one of the most recognized sound formalization techniques [33]. The amplitude spectrum of sound wave in a compact vectorial form is represented using MFCC. Forty features have been extracted and used with MFCC for training the model.

Figure 3 explains the model architecture for speech emotion recognition using CNN. Input to the network is a vector of 40 features for each speech file. On the architecture, one-dimensional CNN with rectified linear unit activation function, 20% dropout and maxpooling is used. Pooling is used for the model to focus on principal characteristics of data. A dropout and flatten is added to make it compatible for



Fig. 3 Model architecture for speech emotion recognition

other layers. Finally, dense layer with softmax activation function is added for the prediction. Dataset used was Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [34]. Speech dataset has angry, fearful, surprise, neutral, calm, happy, sad, and disgust expressions in the data, and it has 7356 files made by 12 female and 12 male professional actors. Figure 4 shows the waveplots for speech data for all the emotion types.

Waveplots are used to plot a signals amplitude envelope over time. It is helpful to choose the right feature extraction method which in this case is mel-frequency cepstral coefficients (MFCC) by looking at the overall shape of the emotion.

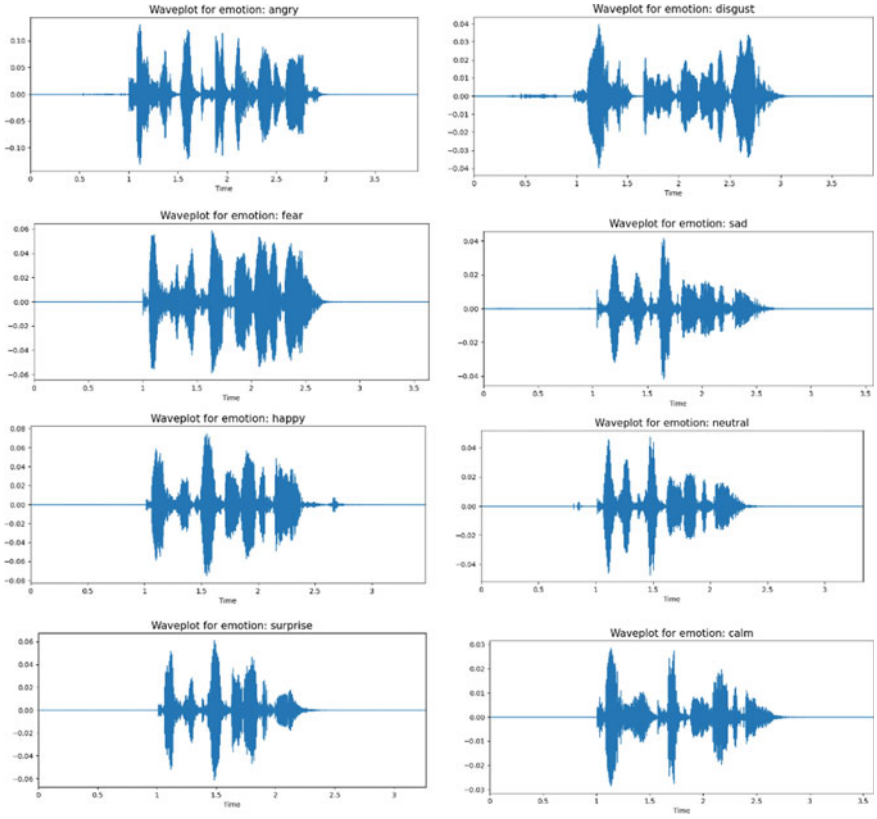


Fig. 4 Waveplots for speech data

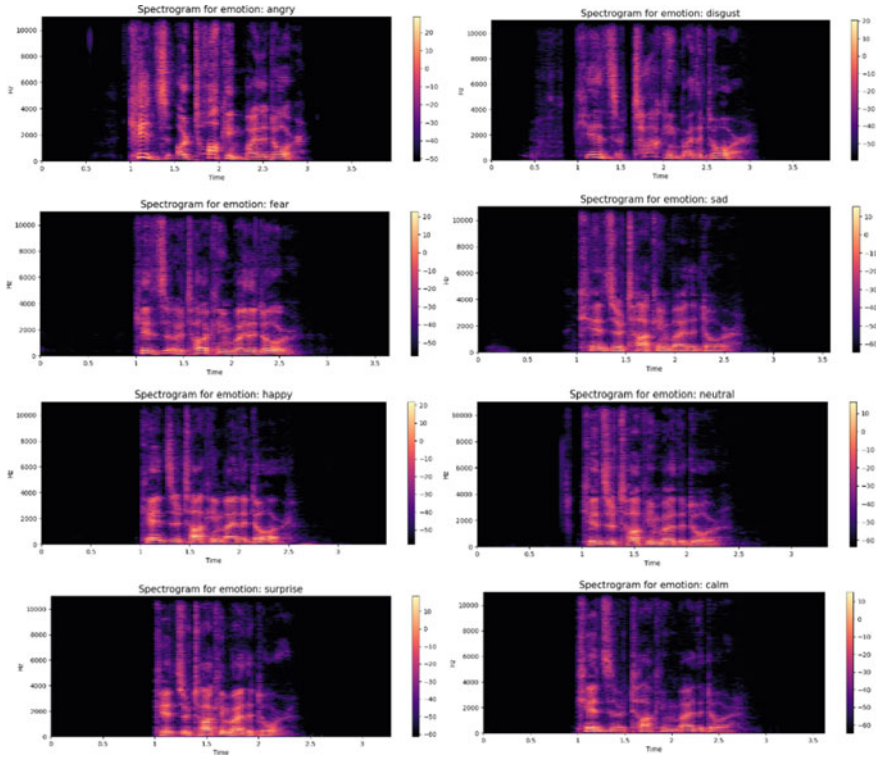


Fig. 5 Spectrograms for speech data

Spectrograms for speech data are shown in Fig. 5. Spectrogram of a time signal depicts two-dimensional graph with frequency in the vertical axis and time in the horizontal axis. In spectrogram, darkest purple represents low energy and lightest red points with high energy. During training, files have been split into train and test datasets with 77:33, respectively. Training set has 3315 mel-frequency cepstral coefficients (MFCC) vector of 40 features. Model is trained using sparse categorical cross-entropy loss function for 50 epochs along with rmsprop optimizer.

Figure 6 shows the model accuracy during training with 1000 epochs. Overall F1-score is 81.14%, and the best score is obtained for "anger" and "neutral" classes with 87% and 86% F1-scores. From the confusion matrix, it can be inferred that there is overlapping between happy and calm, fear and sad. Output prediction from the speech emotion recognition is used to calculate the HDRS points to predict the depression.

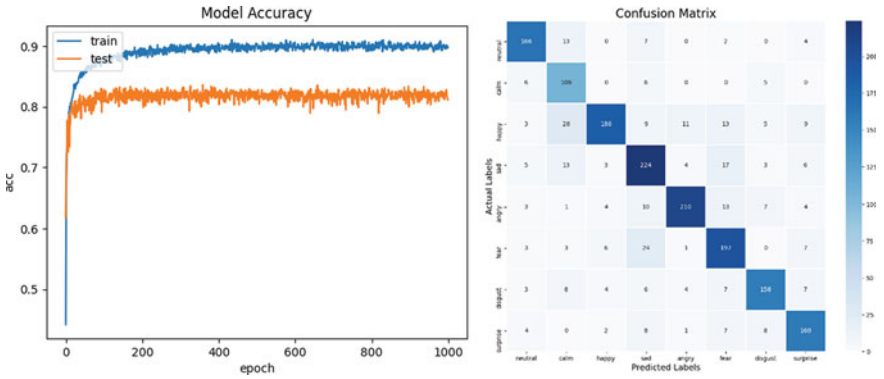


Fig. 6 Model accuracy and confusion matrix

3.3 Text Emotion Recognition Model

Text emotion recognition model is based on bidirectional encoder representations from transformers (BERT) and derived from [35]. Architecture consists of two stages, namely BERT fine-tuned training and bidirectional LSTM (Bi-LSTM) classification, as shown in Fig. 7. Datasets used are International Survey on Emotion Antecedents and Reactions (ISEAR) [36], DailyDialog [37], and Emotion-Stimulus [38]. All the datasets are combined and have the classes for joy, sadness, anger, fear, and neutral. Data preprocessing is performed on the input text, and natural language processing (NLP) techniques like stop words removal, lemmatization, and stemming are applied on the dataset. BERT fine-tuning stage uses self-attention and transformers to modeling language using a bidirectional pre-training approach. Bert-base-uncased model is used with 12-layer transformer blocks with each block having 768 hidden layers and 12 head self-attention layers that produce around 110 million parameters. Each transformer receives list of token embeddings and produces a feature vector of same length at the output.

Twelve transformer layer containing vector transformations is used as aggregated sequence transformation for classification. For bi-LSTM classification, input layer, mask layer, bidirectional LSTM layer, and dense layer are attached to the BERT model. Input layer receives the output from previous stage, bidirectional layer has 100 neurons, and a dense layer is containing five layers to predict the five emotion classes using softmax activation function. Overall accuracy is 81.79%, and classification

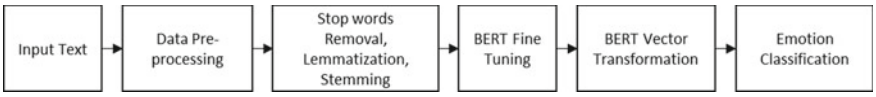


Fig. 7 BERT text emotion recognition architecture

	precision	recall	f1-score	support
joy	0.85	0.84	0.84	707
sadness	0.81	0.80	0.80	676
fear	0.85	0.84	0.84	679
anger	0.79	0.79	0.79	693
neutral	0.79	0.82	0.80	638
accuracy			0.82	3393
macro avg	0.82	0.82	0.82	3393
weighted avg	0.82	0.82	0.82	3393

Fig. 8 Classification report for BERT model

report is shown in Fig. 8. Emotions for "joy" and "fear" have the highest F1-score with 84% for both. Emotions from text will be used for calculating depression severity.

3.4 Depression Severity Detection with HDRS

HDRS is used to evaluate the severity of depression. This paper proposes a novel approach to calculate depression severity by using emotion detected from all the modalities and HDRS. Emotion detected from all the modalities above is used to calculate the final score for all the 17 items in $HDRS_{17}$.

Sample question during structured interview of HDRS is shown in Fig. 9. For each item, one cue which best characterizes the patient has to be selected. Appropriate answers have to be recorded in positions 0 through 4. Sample rule to find point category for the question "Are you feeling bad about yourself or that you are a failure, or let yourself or your family down?" is shown in Table 1. Emotions are represented with characters as sad—S, anger—A, and neutral—N. Minimum emotion frequency count is performed for each question for all the modalities; e.g., if the frequency count of sad emotion is 2, anger is 4 and neutral is 0, then it is represented as S2-A4-N0.

1 **DEPRESSED MOOD** *(sadness, hopeless, helpless, worthless)*

0 |__| Absent.

1 |__| These feeling states indicated only on questioning.

2 |__| These feeling states spontaneously reported verbally.

3 |__| Communicates feeling states non-verbally, i.e. through facial expression, posture, voice and tendency to weep.

4 |__| Patient reports virtually only these feeling states in his/her spontaneous verbal and non-verbal communication.

Fig. 9 Sample question 1 of structured interview for HDRS

Table 1 Rules for calculating score in HDRS for sample question

Point category	Facial emotion recognition	Speech emotion recognition	Text emotion recognition	Selected point category
0	S0-A0-N2	S0-A0-N1	S0-A0-N1	2
1	S1-A0-N0	S2-A0-N0	S2-A0-N0	
2	S2-A1-N0	S2-A2-N0	S2-A1-N0	
3	S3-A2-N0	S3-A3-N0	S3-A3-N0	
4	S4-A3-N0	S3-A4-N0	S3-A4-N0	

Table 2 Severity classification for depression using HDRS score

Total points	Severity of depression
0–7	No depression
8–17	Mild
18–24	Moderate
>25	Severe

Algorithm does the minimum frequency count match from bottom-up for the defined rules, and the rule which matches all the criteria for all the emotions will be picked up and corresponding point category will be selected for a particular question. If there are multiple matches, single rule will be selected while going from bottom to top in the rules so that priority will be given from higher points to smaller points. In this way, it is possible to identify high severity and then go to moderate, mild, and no severity for depression detection. Final score is calculated by summing up the individual point categories for each question. Severity detection is classified as in Table 2.

4 Results

$HDRS_{17}$ has 17 items which are used to prepare 17 questions that will be used in the interview through a webcam and mic for depression counseling. User will sit in front of the webcam with microphone enabled and respond to 17 questions. After each question, recording will be interrupted and the video, speech, and converted text are stored as files on the drive. Later these files are used to calculate the point categories and finally to calculate the overall score for identifying the depression severity.

Screenshot of live depression severity detection system is shown in Fig. 10. Emotions are recognized from all the modalities for individual question, and total score is calculated using HDRS for classifying the severity. Sample final report is shown in Fig. 11 which classifies the depression severity for $HDRS_{17}$.

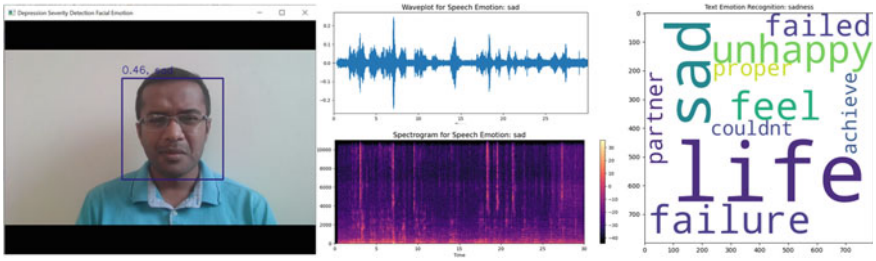


Fig. 10 Emotion recognition from face, speech, and text

Automatic Depression Severity Detection Report

Patient Name: Madhu Sudhan H V

Item No	Points Obtained
1	0
2	0
3	0
4	1
5	0
6	1
7	0
8	0
9	1
10	2
11	0
12	1
13	0
14	0
15	1
16	1
17	2
Total Score	**10**
Depression Severity	**MILD**

**Please note that this tool and report is for self-evaluation purposes only. Please consult your physician before taking any actions.

Fig. 11 Depression severity detection report

5 Conclusion

This paper proposes a novel depression severity prediction system using deep neural networks and HDRS. Architecture of individual modalities for face, speech, and text is defined and analyzed with different datasets and techniques. Facial emotion recognition gave an accuracy of 66%, speech emotion recognition gave accuracy up to 81.14%, and text emotion recognition gave accuracy of 81.79%. All the emotions were combined and rules developed to calculate the point categories and finally calculate the overall score for HDRS which in turn is used to classify the severity. The

proposed system is fully automated, and the end result is a report that can be consumed by the patient for self-evaluation. Limitations include non-availability of enough data to validate the severity of depression, and evaluation would be subjective in nature as the points allocated in $HDRS_{17}$ may vary from therapist to therapist. Other limitation is fake emotions which are not considered in the current study. Future scope would involve using hand pose movements, gesture, eye gaze movements, mood swings to identify the depression severity. Future scope would also involve performing a comparative analysis for depression detection using electroencephalography (EEG) signals from brain.

References

1. WHO Homepage. https://www.who.int/news-room/fact-sheets/detail/depression. Last accessed 2 May 2021
2. Katoon W, Sullivan MD (1990) Depression and chronic medical illness. J Clin Psychiatry 51(Suppl 6):3–11
3. Waang (2007). Use of mental health services for anxiety, mood, and substance disorders in 17 countries in the WHO world mental health surveys. The Lancet 370(9590):841–850
4. Edwards G, Taylor C, Cootes T (1998) Interpreting face images using active appearance models. In Proceedings of the IEEE international conference on automatic face and gesture recognition FG'98, Apr. 1998. IEEE, Nara, Japan, pp 300–305
5. Saraagih J, Goecke R (2009) Learning AAM fitting through simulation. Pattern Recogn 42(11):2628–2636
6. Joshi J, Dhall A, Goecke R, Cohn JF (2013) Relative body parts movement for automatic depression analysis. In: 2013 Humaine association conference on affective computing and intelligent interaction, pp 492–497.https://doi.org/10.1109/ACII.2013.87
7. Kleinsmith A, Bianchi-Berthouze N (2012) Affective body expression perception and recognition: a survey. IEEE Trans Affect Comput 99, 1
8. Littlewort G, Bartlett M, Fasel I, Susskind J, Movellan J (2006) Dynamics of facial expression extracted automatically from video. Image Vis Comput 24(6)
9. Bartlett MS, Littlewort G, Frank MG, Lainscsek C, Fasel I, Movellan JR (2006) Automatic recognition of facial actions in spontaneous expressions. J Multimedia
10. Ekman P, Friesen W (1978) Facial action coding system: a technique for the measurement of facial movement. Consulting Psychologists Press
11. Alizadeh S, Fazel A (2017) Convolutional neural networks for facial expression recognition. arXiv preprint arXiv:1704.06756
12. Arriaga O, Valdenegro-Toro M, Plöger P (2019) Real-time convolutional neural networks for emotion and gender classification. ArXiv, abs/1710.07557
13. Venkataraman D, Parameswaran NS (2018) Extraction of facial features for depression detection among students. Int J Pure Appl Math 118
14. Iqbal A, Barua K (2019) A real-time emotion recognition from speech using gradient boosting. In: International conference on electrical, computer and communication engineering (ECCE). IEEE, pp 1–5
15. Jannat R, Tynes I, Lime LL, Adorno J, Canavan S (2018) Ubiquitous emotion recognition using audio and video data. In: Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers. ACM, pp 956–959
16. Zhang B, Essl G, Provost EM (2015) Recognizing emotion from singing and speaking using shared models. In: 2015 International conference on affective computing and intelligent interaction (ACII) (2015). IEEE, pp 139–145

17. de Pinto MG, Polignano M, Lops P, Semeraro G (2020) Emotions understanding model from spoken language using deep neural networks and mel-frequency cepstral coefficients. In: 2020 IEEE conference on evolving and adaptive intelligent systems (EAIS), pp 1–5.https://doi.org/10.1109/EAIS48028.2020.9122698
18. Venkataramanan K, Rajjamohan H (2019) Emotion recognition from speech. ArXiv, abs/1912.10458
19. Zhang Y, Du J, Wang Z, Zhang J, Tu Y (2018) Attention based fully convolutional network for speech emotion recognition. In: 2018 Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC), pp 1771–1775.https://doi.org/10.23919/APSIPA.2018.8659587
20. Sahu G (2019) Multimodal speech emotion recognition and ambiguity resolution. ArXiv, abs/1904.06022
21. Preot D, Eichstaedt J, Park G, Sap M, Smith L, Tobolsky V, Schwartz HA, Ungar L (2015) The role of personality, age and gender in tweeting about mental illnesses. In: Proceedings of the workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality, pp 21–30
22. Resnik P, Armstrong W, Claudino L, Nguyen T (2015) The University of Maryland CLPsych 2015 shared task system. In: CLPsych 2015 shared task system, pp 54–60
23. Kshirsagar R, Morris R, Bowman S (2017) Detecting and explaining crisis. In: Proceedings of the fourth workshop on computational linguistics and clinical psychology—from linguistic signal to clinical reality, Vancouver. Association for Computational Linguistics, pp 66–73
24. Orabi AH, Buddhitha P, Orabi MH, Inkpen D (2018) Deep learning for depression detection of Twitter users. In: Fifth workshop on computational linguistics and clinical psychology, pp 88–97
25. Singh D, Wang A. Detecting depression through tweets. Stanford University, CA, pp 1–9
26. Hamilton M (1967) Development of a rating scale for primary depressive illness. Br J Soc Clin Psychol 6(4):278–296
27. Williams JB (1988) A structured interview guide for the Hamilton depression rating scale. Arch Gen Psychiatry 45(8):742–747
28. Kingma D, Adam JB (2014) A method for stochastic optimization. arXiv preprint arXiv:1412.6980
29. Zahara L, Musa P, Wibowo EP, Karim I, Musa SB (2020) The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi. In: 2020 Fifth international conference on informatics and computing (ICIC), pp 1–9. https://doi.org/10.1109/ICIC50835.2020.9288560
30. Chollet F (2016) Xception: deep learning with depthwise separable convolutions. CoRR, abs/1610.02357
31. Howard AG (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861
32. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
33. Muda L, Begam M, Elamvazuthi I (2010) Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. arXiv preprint arXiv:1003.4083
34. Livingstone SR, Russo FA (2018) The Ryerson audio-visual database of emotional speech and song (ravdess): a dynamic, multimodal set of facial and vocal expressions in North American English. PloS one 13(5):e0196391
35. Aadoma AF, Henry N, Chen W, Niyongabo RA (2020) Recognizing emotions from texts using a Bert-based approach. In: 17th International computer conference on wavelet active media technology and information processing (ICCWAMTIP), pp 62–66
36. ISEAR Homepage. https://www.unige.ch/cisa/research/materials-and-online-research/research-material/. Last accessed 2 May 2021

37. Li Y, Su H, Shen X, Li W, Cao Z, Niu S (2017) DailyDialog: a manually labelled multi-turn dialogue dataset. IJCNLP 2017
38. Ghazi D, Inkpen D, Szpakowiczz S (2015) Detecting emotion stimuli in emotion-bearing sentences. In: Proceedings of the 16th international conference on intelligent text processing and computational linguistics (CICLing 2015), Cairo, Egypt

Machine Learning Model for Automated Sleep Scoring Based on Single-Channel EEG Signal Data



Santosh Satapathy and Hari Kishan Kondaveeti

Abstract Sleep is an important aspect of human body which alternatively decides the physical and mental conditions. It has been seen that several sleep-related disorders are affected with irrespective of different age group subjects. The most important part of diagnosis any types of sleep disorders is the classification of sleep stages. But the manual inspection of sleep-related behavior is quite difficult and more time-consuming process. Sometimes it leads toward several other neurological disorders. Therefore in this research work, we propose an automated sleep staging approach for the classification of the sleep stages. Mainly the proposed experiments followed three basic steps such as preprocessing feature extraction, feature selection, and classification. The entire experiment is conducted upon the ISRUC-Sleep dataset. The proposed model reported an overall accuracy of 97.73% using machine learning classification model.

Keywords Electroencephalography · Automatic sleep stage analysis · Time–frequency features · Machine learning

1 Introduction

Sleep is one of the important parts of human life which is directly responsible for maintaining the physical and mental fitnesses of the human body. Generally, the human spends their one-third of their lives as sleeping [1, 2]. It has been found that sleep disorders not only affect the physical ability during the day but their effects were also seen during the long-term manners like reducing the ability of learning, attention, and memory. In general, sleep-related disorders are diagnosed by analyzing the sleep pattern structures including its changes in characteristics over the individual

S. Satapathy (✉)
Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India
e-mail: santosh.satapathy@pec.edu

H. K. Kondaveeti
School of Computer Science Engineering, VIT-AP University, Vijayawada, Andhra Pradesh, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_30

sleep stages, overall sleep time, and its temporal organization of the sleep cycles throughout an average night. Generally, the normal sleep cycles for a human lasting for 90–110 min, which repeats 4–5 times in a normal sleep cycle and it typically begins with light sleep gradually shifting into deeper sleep [3]. Mainly, there are two sleep scoring guidelines followed by the clinicians during the sleep staging. The Rechtschaffen and Kales (R&K) segmented into six sleep stages which includes REM, N4, N3, N2, and N1, Wake stage [4, 5]. Recently, the clinicians followed new sleeping rules called as the American Academy of Sleep Medicine (AASM) [6]. This most important evaluation method of sleep staging is PSG testing, which contains different physiological signals such as electroencephalogram (EEG), electrooculogram (EOG), electromyogram (EMG), and electrocardiogram (ECG) [7]. The manual inspection of sleep behavior requires more human experts, and variations occurred in sleep staging performance with subjective to clinician's expertise. The other shortcoming of the PSG test is, more expensive and also creates more uncomfortable situations during the fixing of the more electrodes in the body for acquiring the physiological signals during sleep [6, 8]. Therefore, it is more important to develop a sleep staging system that could be a great help during sleep staging. The EEG signals are considered as one of the most effective toward monitoring the sleep behavior in classifying the sleep stages either in manual scoring by human experts or automated classification systems [9]. The EEG signals are collected during sleep considering single channel or multiple channels. The EEG signals have several advantages. The signals can be obtained using wearable technologies, which are comfortable for subjects. Moreover, the data collection process can be done in the patient's home or healthcare facilities [10].

1.1 Related Work

There are several contributions proposed by the different researchers on sleep staging. The common steps followed by the different authors were data acquisition, data preprocessing, feature extraction, feature selection, and classification. From the literatures, it has been found that most of the contributions use single and multiple channels. Similarly, most of the research works used linear and nonlinear features. Additionally few of studies adopted the feature selection step for reducing the complexity of the feature vector. For the classification of the sleep stages, mostly the researchers used traditional machine learning classification models. Here we have discussed some of the contributions proposed by the researchers and identify contained existing challenges.

Obayya et al. [11] achieved overall classification accuracy reported was 85% using fuzzy c-means algorithm.

In Güneş et al. [12], the author obtained the welch spectral transform and decision trees techniques and achieved 83%.

In Aboalayon et al. [13], the authors extracted features forwarded into SVM classifier and the model resulted as 90%.

In Hassan and Subasi [14], the authors have obtained bootstrap techniques and the reported accuracy is 92.43%.

In Diykh et al. [15], the extracted linear properties and achieved 95.93% using SVM classifier.

In Gunnarsdottir et al. [16], the model was performed with the healthy controlled subjects sleep recordings and achieved 80%.

In Sriraam et al. [17], author proposed sleep scoring based on feedforward neural network. The accuracy performance of the model reached to 92.9%

Memar and Faradji [18] have used Kruskal–Wallis test and RF for selecting features and classification of the sleep stages, respectively. The whole experiment was performed through cross-wise and subject-wise validation. The performance achieved 95.31% and 86.64% for with fivefold cross-validation and subject-wise validation, respectively.

Da Silveira et al. [19] obtained RF techniques stages and resulted in accuracy is 90%.

Wutzl B et al. [20] extracted linear and nonlinear features and fed into the RF classifiers. The classification model achieved as 85.95%.

Zhu et al. [21] used graph features and SVM. The performance of sleep staging reported as 87.5%

Braun et al. [22] used a different combination of features and the system achieved an accuracy of 97.1%.

This proposed work aims to design a framework for sleep staging using different gender-specific subjects having different symptoms of sleep problems. Our proposed sleep EEG study performed well even using only a single EEG channel, an interesting observation of our work. We described the experimental details in Sect. 2. The proposed methodology is discussed in Sect. 3. Section 4 briefly discusses the results of the proposed model. Sect. 5 gives the concluding remarks of the research work.

2 Experimental Sleep Data

The required data retrieved from the subgroup-I data of ISRUC-sleep database. The entire dataset was directly monitored by which was prepared by the set of experts at Hospital of Coimbra University [23, 24]. In this study, the brain behavior was monitored through C3-A2 channel for computing sleep stage classification. The retrieved sleep epochs information from the individual subjects is described in Table 1.

Table 1 Distribution of sleep epochs

Subject/Subgroup	W	N_1	N_2	N_3	REM
Subject-1 subgroup-1/One session	165	63	173	231	118
Subject-2 subgroup-1/One session	231	72	226	147	74
Subject-3 subgroup-1/One session	119	142	194	126	169
Subject-4 subgroup-1/One session	19	44	327	214	146
Subject-5 subgroup-1/One session	245	104	199	164	38
Subject-9 subgroup-1/One session	72	143	315	136	84
Subject-16 subgroup-I/One session	128	125	280	120	97
Subject-23 subgroup-I/One session	212	99	270	65	104

3 Methodology

In this study, we propose an automated machine learning-based sleep staging system is proposed. The complete layout of the proposed model is described in Fig. 1. The extracted features from the input signals are described in Table 2.

3.1 Feature Selection

The final selected feature is presented in Table 3. The features were selected through online streaming feature selection techniques [25, 26].

4 Experimental Results and Discussion

The proposed experiments are executed and coded using MATLAB software. The analysis of the effectiveness of the proposed methodology, we obtained performance metrics such as accuracy (Acc) [27], recall (Re) [28], specificity (Spc) [29], precision (Pre) [30], F1-score (F1sc) [31], and Kappa score (Ksc) [32, 33]. Most of the studies considered the C3-A2 channel for recording the sleep behavior, so in this study we considered this channel as input channel [34–51]. The final decision was taken based on the sleep staging score as per proper consultation with sleep experts [52]. The reported performance of each individual subjects is presented in Table 4.

The details with subject to analysis the performance in between proposed system and existing contributions are presented in Table 5. The presented study shows that it is possible to achieve high classification accuracies through three different sleep data subgroups and considering different patients with different medical conditions. The authors validate the effectiveness and usefulness of the proposed method by presenting a detailed comparative analysis with similar research proposals available

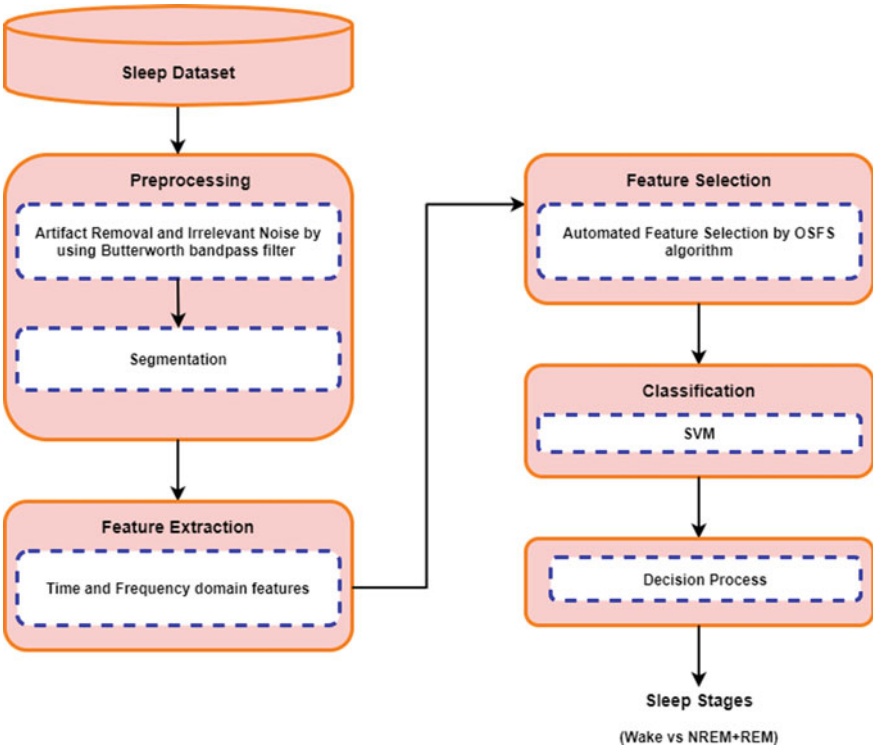


Fig. 1 Block diagram of the proposed method

Table 2 Extracted features

Feature type	Extracted feature	Number
Time domain	Mean	1
	Max	2
	Min	3
	Standard deviation	4
	Median	5
	Variance	6
	Zero-crossing rate	7
	75 percentile	8
	Skewness	9
	Kurtosis	10
	Hjorth parameters	11–13
Frequency domain	Relative spectral power (δ–β) band	14–18
	Power ratios	18–24
	Power band	25–28

Table 3 Final feature selection list

Subject	Selected features	Total
Subject1/Subgroup-I	Fe_1_1, Fe_2_1, Fe_3_1, Fe_4_1, Fe_5_1, Fe_7_1, Fe_9_1, Fe_10_1, Fe_11_1, Fe_13_1, Fe_14_1, Fe_15_1, Fe_22_1, Fe_25_1, Fe_27_1	15
Subject2/Subgroup-I	Fe_1_2, Fe_5_2, Fe_6_2, Fe_7_2, Fe_8_2, Fe_9_2, Fe_10_2, Fe_11_2, Fe_12_2, Fe_13_2, Fe_14_2, Fe_16_2, Fe_18_2, Fe_20_2, Fe_27_2	15
Subject3/Subgroup-I	Fe_1_3, Fe_5_3, Fe_6_3, Fe_7_3, Fe_8_3, Fe_10_3, Fe_12_3, Fe_13_3, Fe_17_3, Fe_18_3	10
Subject4/Subgroup-I	Fe_1_4, Fe_2_4, Fe_3_4, Fe_4_4, Fe_10_4, Fe_12_4, Fe_14_4, Fe_15_4, Fe_16_4, Fe_17_4, Fe_18_4, Fe_22_4, Fe_24_4, Fe_28_4	14
Subject5/Subgroup-I	Fe_1_5, Fe_5_5, Fe_6_5, Fe_7_5, Fe_8_5, Fe_9_5, Fe_10_5, Fe_12_5, Fe_13_5, Fe_14_5, Fe_15_5, Fe_17_5, Fe_18_5, Fe_23_5, Fe_28_5	15
Subject9/Subgroup-I	Fe_1_9, Fe_5_9, Fe_6_9, Fe_7_9, Fe_8_9, Fe_9_9, Fe_10_9, Fe_12_9, Fe_13_9, Fe_14_9, Fe_18_9, Fe_28_9	12
Subject16/Subgroup-I	Fe_1_{16}, Fe_2_{16}, Fe_3_{16}, Fe_4_{16}, Fe_5_{16}, Fe_7_{16}, Fe_9_{16}, Fe_10_{16}, Fe_11_{16}, Fe_13_{16}, Fe_14_{16}, Fe_15_{16}, Fe_22_{16}, Fe_25_{16}, Fe_27_{16}	15
Subject23/Subgroup-I	Fe_1_{23}, Fe_2_{23}, Fe_3_{23}, Fe_4_{23}, Fe_5_{23}, Fe_7_{23}, Fe_11_{23}, Fe_12_{23}, Fe_15_{23}, Fe_16_{23}, Fe_17_{23}, Fe_20_{23}, Fe_24_{23}, Fe_26_{23}	14

Table 4 Performance of the proposed sleep EEG study using SVM classifiers for ISRUC-sleep subgroup-I

C3-A2/SVM	Pre (%)	Re (%)	Spc (%)	F1sc (%)	Acc (%)	KSc
Subject1	92.33	96.75	71.51	94.49	91.2	0.72
Subject2	94.71	96.72	87.87	97.71	94	0.85
Subject3	94.42	96.67	69.74	95.53	92.4	0.70
Subject4	98.63	99.04	47.36	98.83	97.73	0.5
Subject5	96.17	94.65	92.24	95.40	93.86	0.86
Subject6	96.18	96.60	63.88	96.39	93.46	0.61
Subject16	95.11	99.49	81.60	97.25	95.60	0.86
Subject23	93.09	95.17	82.08	94.12	91.46	0.78

in the literature. The authors have analyzed the classification results of our proposed work with other eight similar studies that include the same dataset.

The classification accuracy results of the existing contributed work for two stages are fallen in the ranges of 75.29% to 95%, respectively. The proposed research work has obtained better accuracy results for two-stage classification. We achieved an overall accuracy of 97.73% with subgroup-I subjects through the SVM classifier.

Table 5 Result analysis in between proposed study with state-of-the-art contributions

Author	Classifier	Accuracy (%)
Ref. [53]	SVM	95
Ref. [54]	Bayesian classifier	83
Ref. [55]	SVM	93.97
Ref. [56]		86.75
Ref. [57]		81.74
Ref. [58]	Random Forest	75.29
Ref. [59]	SSAE	82.3
Ref. [60]	SVM	83.33
Proposed	SVM	97.73

5 Conclusion and Future Directions

In this study, we proposed an artificial intelligence-based sleep staging system, where we obtained the sleep recordings of the subjects who were affected with different sleep-related disorders. For accurate recordings from the brain, we selected the C3-A2 channel as input channel of the EEG signals. The main contribution of this study is how best to classify the sleep patterns using machine learning classification model. The proposed methodology extracted time and frequency domain features from the processed signals to identify the irregularities of the sleep behavior. Finally, the best suitable features are forwarded into the SVM classification model for classifying the sleep stages. The model was evaluated on the ISRUC-Sleep data. The performance of the proposed model well-performed with subject to the existing contributions. We will further extend this proposed methodology by considering more combinations of the channels and signals. Additionally, we will also focus on including more number of subjects during sleep staging.

References

1. Panossian LA, Avidan AY (2009) Review of sleep disorders. Med Clin N Am 93:407–425. https://doi.org/10.1016/j.mcna.2008.09.001
2. Smaldone A, Honig JC, Byrne MW (2007) Sleepless in America: inadequate sleep and relationships to health and well-being of our nation's children. Pediatrics 119:29–37
3. Prabhu SS, Sinha N (2020) Sleep EEG analysis utilizing inter-channel covariance matrices. Biocybern Biomed Eng 40(1):527–545
4. Rechtschaffen A, Kales A (1968) A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects. In: Brain information service, pp 1–350
5. Boostani R, Karimzadeh F, Nami M (2017) A comparative review on sleep stage classification methods in patients and healthy individuals. Comput Meth Programs Biomed 140:77–91
6. Iber C, Ancoli-Israel S, Chesson AL, Quan SF (2007) The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specification. American Academy of Sleep Medicine

7. Abeyratne UR, Swarnkar V, Rathnayake SI, Hukins C (2007) Sleep-stage and event dependency of brain asynchrony as manifested through surface EEG. In: Proceedings of the 29th IEEE annual international conference of the engineering in medicine and biology society, pp 709–712
8. Rechtschaffen A, Kales A (1968) A manual of standardized terminology, techniques and scoring systems for sleep stages of human subjects. U.G.P. Office, Public Health Service, Washington, DC, USA
9. Bianchi MT (2017) Sleep devices: wearables and nearables, informational and interventional, consumer and clinical. Metabolism
10. Cogan D, Birjandtalab J, Nourani M, Harvey J, Nagaraddi V (2017) Multi-biosignal analysis for epileptic seizure monitoring. Int J Neural Syst. https://doi.org/10.1142/S0129065716500313
11. Obayya M, Abou-Chadi F (2014) Automatic classification of sleep stages using EEG records based on Fuzzy c-means (FCM) algorithm. In: Radio science conference (NRSC), pp 265–272
12. Güneş S, Polat K, Yosunkaya Ş (2010) Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. Expert Syst Appl 37:7922–7928
13. Aboalayon K, Ocbagabir HT, Faezipour M (2014) Efficient sleep stage classification based on EEG signals. In: Systems, applications and technology conference (LISAT), pp 1–6
14. Hassan AR, Subasi A (2017) A decision support system for automated identification of sleep stages from single-channel EEG signals. Knowl-Based Syst 128:115–124
15. Diykh M, Li Y, Wen P (2016) EEG sleep stages classification based on time do-main features and structural graph similarity. IEEE Trans Neural Syst Rehabil Eng 24(11):1159–1168
16. Gunnarsdottir KM, Gamaldo CE, Salas RME, Ewen JB, Allen RP, Sarma SV (2018) A novel sleep stage scoring system: combining expert-based rules with a decision tree classifier. In: 40th Annual international conference of the IEEE engineering in medicine and biology society (EMBC)
17. Sriraam N, Padma Shri TK, Maheshwari U (2018) Recognition of wake-sleep stage 1 multichannel EEG patterns using spectral entropy features for drowsiness detection. Australas Phys Eng Sci Med 39(3):797–806
18. Memar P, Faradji F (2018) A novel multi-class EEG-based sleep stage classification system. IEEE Trans Neural Syst Rehabil Eng 26(1):84–95
19. Da Silveira TLT, Kozakevicius AJ, Rodrigues CR (2016) Single-channel EEG sleep stage classification based on a streamlined set of statistical features in wavelet domain. Med Biol Eng Compu 55(2):343–352
20. Wutzl B, Leibnitz K, Rattay F, Kronbichler M, Murata M (2019) Genetic algorithms for feature selection when classifying severe chronic disorders of consciousness. PLOS ONE 14(7)
21. Zhu G, Li Y, Wen PP (2014) Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal. IEEE J Biomed Health Inform 18(6):1813–1821
22. Braun ET, Kozakevicius ADJ, Da Silveira TLT, Rodrigues CR, Baratto G (2018) Sleep stages classification using spectral based statistical moments as features. Revista de Informática Teórica e Aplicada 25(1)
23. Scholkopf B, Smola A (2002) Learning with kernels. MIT Press
24. Khalighi S, Sousa T, Santos JM, Nunes U (2016) ISRUC-Sleep: A comprehensive public dataset for sleep researchers. Comput Meth Progr Biomed 124:180–192
25. Eskandari S, Javidi MM (2016) Online streaming feature selection using rough sets. Int J Approx Reasoning 69:35–57
26. İlhan HO, Bilgin G (2017) Sleep stage classification via ensemble and conventional machine learning methods using single channel EEG signals. Int J Intell Syst Appl Eng 5(4):174–184
27. Sanders TH, McCurry M, Clements MA (2014) Sleep stage classification with cross frequency coupling. In: 36th Annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp 4579–4582
28. Bajaj V, Pachori R (2013) Automatic classification of sleep stages based on the time-frequency image of EEG signals. Comput Meth Programs Biomed 112(3):320–328

29. Hsu Y-L, Yang Y-T, Wang J-S, Hsu C-Y (2013) Automatic sleep stage recurrent neural classifier using energy features of EEG signals. Neurocomputing 104:105–114
30. Zibrandtsen I, Kidmose P, Otto M, Ibsen J, Kjaer TW (2016) Case comparison of sleep features from ear-EEG and scalp-EEG. Sleep Sci. 9(2):69–72
31. Berry RB, Brooks R, Gamaldo CE, Hardsim SM, Lloyd RM, Marcus CL, Vaughn BV (2014) The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications. American Academy of Sleep Medicine
32. Sim J, Wright CC (2005) The kappa statistic in reliability studies: use, interpretation, and sample size requirements. Phys Therapy 85(3):257–268
33. Liang S-F, Kuo C-E, Kuo YH, Cheng Y-S (2012) A rule-based automatic sleep staging method. J. Neurosci Meth 205(1):169–176
34. Liang S-F, Kuo C-E, Hu Y-H, Pan Y-H, Wang Y-H (2012) Automatic stage scoring of single-channel sleep EEG by using multiscale entropy and autoregressive models. 61
35. Kim J (2014) A comparative study on classification methods of sleep stages by using EEG. J Korea Multimed Soc 17(2):113–123
36. Peker M (2016) A new approach for automatic sleep scoring: combining Taguchi based complex-valued neural network and complex wavelet transform. Comput Meth Programs Biomed 129:203–216
37. Subasi A, Kiymik MK, Akin M, Erogul O (2005) Automatic recognition of vigilance state by using a wavelet-based artificial neural network. Neural Comput Appl 14(1):45–55
38. Tagluk ME, Sezgin N, Akin M (2010) Estimation of sleep stages by an artificial neural network employing EEG, EMG and EOG. J Med Syst 34(4):717–725
39. Hassan AR, Bhuiyan MIH (2017) An automated method for sleep staging from EEG signals using normal inverse Gaussian parameters and adaptive boosting. Neurocomputing 219:76–87
40. Hassan AR, Bhuiyan MIH (2017) Automated identification of sleep states from EEG signals by means of ensemble empirical mode decomposition and random under sampling boosting. Comput Meth Programs Biomed 140:201–210
41. Diykh M, Li Y (2016) Complex networks approach for EEG signal sleep stages classification. Expert Syst Appl 63:241–248
42. Diykh M, Li Y, Wen P (2016) EEG sleep stages classification based on time domain features and structural graph similarity. IEEE Trans Neural Syst Rehabil Eng 24(11):1159–1168
43. Mahvash Mohammadi S, Kouchaki S, Ghavami M, Sanei S (2016) Improving time–frequency domain sleep EEG classification via singular spectrum analysis. J Neurosci Methods 273, 96–106
44. Şen B, Peker M, Çavuşoğlu A, Çelebi FV (2014) A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms. J Med Syst 38(3):18
45. Burioka N et al (2005) Approximate entropy in the electroencephalogram during wake and sleep. Clin EEG Neurosci 36(1):21–24
46. Obayya M, Abou-Chadi FEZ (2014) Automatic classification of sleep stages using EEG records based on Fuzzy c-means (FCM) algorithm. In: 2014 31st National radio science conference (NRSC), pp 265–272
47. Fraiwan L, Lweesy K, Khasawneh N, Fraiwan M, Wenz H, Dickhaus H (2018) Classification of sleep stages using multi-wavelet time frequency entropy and LDA. Methods Inf Med 49(3), 230–237
48. Herrera LJ et al (2013) Combination of heterogeneous EEG feature extraction methods and stacked sequential learning for sleep stage classification. Int J Neural Syst 23(03):1350012
49. Radha M, Garcia-Molina G, Poel M, Tononi G (2014) Comparison of feature and classifier algorithms for online automatic sleep staging based on a single EEG signal. In: 36th Annual international conference of the IEEE engineering in medicine and biology society, pp 1876–1880
50. Jo HG, Park JY, Lee CK, An SK, Yoo SK (2010) Genetic fuzzy classifier for sleep stage identification. Comput Biol Med 40(7):629–634

51. Herrera LJ, Mora AM, Fernandes CM (2011) Symbolic representation of the EEG for sleep stage classification. In: 11th International conference on intelligent systems design and applications, pp 253–258
52. Vanbelle SA (2016) New Interpretation of the weighted Kappa coefficients. Psychometrika 81:399–410
53. Khalighi S, Sousa T, Oliveira D, Pires G, Nunes U (2011) Efficient feature selection for sleep staging based on maximal overlap discrete wavelet transform and SVM. In: Annual international conference of the IEEE engineering in medicine and biology society
54. Simões H, Pires G, Nunes U, Silva V. Feature extraction and selection for automatic sleep staging using EEG. In: Proceedings of the 7th international conference on informatics in control, automation and robotics, 3, pp 128–133
55. Khalighi S, Sousa T, Santos JM, Nunes U (2016) ISRUC-sleep: a comprehensive public dataset for sleep researchers. Comput Methods Programs Biomed 124:180–192
56. Sousa T, Cruz A, Khalighi S, Pires G, Nunes U (2015) A two-step automatic sleep stage classification method with dubious range detection. Comput Biol Med 59:42–53
57. Khalighi S, Sousa T, Pires G, Nunes U (2013) Automatic sleep staging: a computer assisted approach for optimal combination of features and polysomnographic channels. Expert Syst Appl 40(17):7046–7059
58. Tzimourta KD, Tsilimbaris AK, Tzioukalia AT, Tzallas MG, Tsipouras LG (2018) EEG-based automatic sleep stage classification. Biomed J Sci Tech Res 7(4)
59. Najdi S, Gharbali AA, Fonseca JM (2017) Feature transformation based on stacked sparse autoencoders for sleep stage classification. In: Technological innovation for smart systems, pp 191–200
60. Kalbkhani H, Ghasemzadeh P, Shayesteh MG (2018) Sleep stages classification from EEG signal based on Stockwell transform. In: IET signal processing 2018

Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders, and Attacks



Alekya Sai Laxmi Kowta, P. K. Harida, Shruti Varsha Venkatraman, Siddharth Das, and V. Priya

Abstract IoT applications are only growing day after day, and the more one uses them, the more the question of their safety arises. The concept of the Internet of Things is to connect various devices to the Internet. The more computing power these devices have, the more security flaws can arise. When your device becomes smart—when it is able to connect to the Internet, people can be vulnerable to an attack by an intruder. We aim to find out various vulnerabilities that IoT applications can contain with the help of pen testing and other kinds of IoT-based tests along with the analysis of the common vulnerabilities and exposures (known as CVE) using Nmap. Our objective is to list out the various security threats that each device can have. We give you a brief overview of these different threats and vulnerabilities by performing some attacks on them. We try to incorporate some fuzzy logic to come up with solutions and efficient methods.

Keywords IoT · Pen testing · Fuzzy testing · Common vulnerabilities and exposures · Denial of service (DoS) · Cyber security

1 Introduction

Wireless sensor networks and radio frequency identification technologies have taken up a spike, and due to this, IoT applications have seen a lot of development in the recent years. IoT aims to transform the way we live. We are living in the smart era right now—smart cities, smart transportation, and infrastructure are some terms that are used in relevance to IoT. Due to the environment that IoT provides us with, the risk of cyber threats is very concerning. Most cyber attacks focus on the infrastructure of the system and try to gain all the system vulnerabilities to start attacks. Cyber attacks are not really new to IoT, but the fact that IoT is a huge part of our lives makes it more of a reason to be aware of how to be protected from these threats,

A. S. L. Kowta · P. K. Harida (✉) · S. V. Venkatraman · S. Das · V. Priya
School of Information Technology and Engineering, Vellore of Institute of Technology, Vellore, Tamil Nadu, India
e-mail: harida.pk2018@vitstudent.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_31

however minor they seem to be. Some of the different threats that are IoT related are device attacks, application service attacks, Web interface attacks, etc. Security is as essential in IoT as it is in information communication system (ICT). The heterogeneity of home-based IoT devices contributes to all these threats and vulnerabilities because even if the core functionalities are the very same, some special features which are based on device type can vary. Because of the vastness of IoT, there is a lot of research going on in this field. Cyber security in IoT is such a huge thing right now. Because of the development of IoT architecture, smart middleware is now going to be capable of creating dynamic maps of the actual world within the virtual realm. It does the mapping with high spatial and temporal precision and also combines the characteristics of various sensor networks together. IoT when connected with other networks through the cloud has different types of attacks. Major attacks include encryption attacks and man-in-the-middle attacks. Lack of authentication and weak authentication are usually the major reason why IoT devices are attacked. Pairing of devices over the cloud establishes a secure and trusted channel between a client and their devices. The DNS protocol is a lightweight protocol that almost all Internet services rely on. But, this leaks very private information based on the client configurations. HSTS can also be attacked through man-in-the-middle attacks. HTTP is known for giving a reliable mode of transportation, but it does not provide any kind of confidentiality or integrity. IoT devices rely on UPnP protocol, and this protocol uses HTTP, so it has the same flaws as HTTP. To address these problems, TLS/SSL sessions are used. Lightweight and symmetric answers for help gadgets compelled to resources. Lightweight key administration frameworks empower the foundation of trust connections and distribute the encryption materials utilizing least correspondence and handling resources, steady with the resources obliged nature of numerous IoT gadgets. Keeping data as nearby as conceivable utilizing decentralized registering and key administration. Anticipation of local protection and individual data protection that people may wish to keep hidden by noticing IoT-related concerns. In the light of need power, bandwidth, transfer speed, and microchip, security countermeasures like public key encryption calculation and recurrence jumping correspondence cannot be applied. Security dangers like maltreatment of resources, Dos attacks, malware, Trojans, viruses also pose great threat with increasing technology. With the frequent changes made in the architecture of menu, smart devices which vary a lot with the standard IoT architecture cause deploying the security solutions as a major challenging task. As most of the IoT devices are connected to Internet whose main functions are related to the cloud, there has been an increasing risk as cloud infrastructure itself is under great threat which could even risks data in terms of their bandwidth, data security, and transferring of data between the cloud to the devices as the network topology is not secured with some security measures as such.

2 Related Work

Till date, the significant difficulties looked in IoT are the security issue [1] which are because of the guideline and the usefulness, while the mechanical issue is because of the heterogeneousness of the gadgets. They proposed a model that utilizes public key, and the MIM assaults are forestalled utilizing timestamp in validating the messages. Because of the conceivably truly reasonable nature of a portion of these information, protection and security viewpoints must be considered by plan and naturally. To fuse different security properties for locally established IoT gadgets, the creator assesses security dependent on the IoT gadgets, versatile application, cloud endpoints, and the correspondence channel. Various assaults have been accounted for like vulnerability services, weak authentication, and default setups. TLS/SSL tends to uncertain conventions that are feasible for man-in-the-middle assaults. Internet of Things (IoT) is the most recent pattern, and like all innovation, it is open for misuse. The most well-known assault which is accustomed to cut down an entire organization, without discovering an escape clause in the security—DoS [2], can be utilized to pull down any IoT network too. In this paper, we propose a honeypot model for moderating DoS assaults dispatched on IoT gadgets. Broad pen testing tools for IoT applications were led in this work to find the weaknesses. Notwithstanding the examination of the weakest security defects characterized by the Open Web Application Security Project (OWASP) [3], this work additionally tried a bunch of man-in-the-center assaults misusing discovered weaknesses. This paper presents the potential weaknesses [4] in the advanced IoT network alongside their countermeasures to keep them from the viewpoint of programming, equipment, and organization foundation. The outline could assist the peruse with acquiring knowledge into the IoT network dangers and security. The significant difficulties confronted are that the conglomeration of the security identified with each layer from the IoT convention stack by thinking about various weaknesses at each layer. Fluffing testing is utilized to test infusion assaults and can be joined with the testing components.

In an examination identified with the IP cameras, they attempt a broader examination of IP cameras' [5] weaknesses and exhibit their impact on clients' security and protection using the Kali Linux entrance testing stage and its apparatuses. For this reason, the paper plays out an involved test on an IP camera with the name ("Wise Onvif YY HD") to investigate the security components of this gadget. The aftereffects of this paper show that IP cameras have a few security needs and shortcomings which these blemishes affect clients. Code-based testing identifies the weaknesses by taking a gander at the code. The benefits of this work is: Observing perspectives distinguishes new weaknesses. As a future work, incorporation of observing procedures can naturally dispatch the testing cycle. Among these chances, making some cooperative energy between the "edge processing + IoT" [6] stage and the arising blockchain and AI innovations might actually produce numerous helpful effects. Security evaluation completed on the IoT organization would feature the shortcomings present in the organization. In view of the shortcomings, a fitting countermeasure ought to be applied to make it secure. This paper [7] presents IoT

testing philosophy that should be joined during or post-execution of IoT innovation. The authors' conversation includes the recognizable proof [8] of various weaknesses inalienable inside IoT application and administration spaces. They executed two distinctive assault situations (tests) on shrewd metering correspondence foundation arrangement. Their work explains the different intricacies clarified in the examination, regarding figuring the monetary effect [9] of IoT digital danger, likewise lead to the end that effect must be evaluated with new danger measurements, and another valuation strategy explicit for the new danger measurements, joined with new administrative structure and normalization IoT information bases with new danger vectors as characterized as International IoT Asset Classification (IIoTAC) and Key IoT Cyber Risk Factors (KIoTCRF) [10]. Before, just mobiles and PCs were associated with the Web; however, in the new period with the appearance of new advances, different things like surveillance cameras, microwaves, vehicles, and mechanical supplies are presently associated with Web [11]. This organization of things is known as the Web of Things. As of late, a large number of surveillance cameras were penetrated to dispatch a DDOS assault that caused Twitter blackout. IoT arrangements are not simply programming yet a whole environment of equipment, programming, cloud, Web, and portable interfaces. This biological system is not extremely developed, and there are as yet significant concerns prowling around IoT appropriation principally because of safety dangers. IoT Top Security Concerns: gadget cloning, delicate information openness, forswearing of administration, unapproved gadget access and control, altering information. This examination work achieves the need to alleviate IoT security challenges gadget cloning and delicate information openness. Administrators of shrewd environments face mounting strain to improve their perceivability into their IoT framework with numerous weak gadgets. This paper built up [12] a constant observing answer for IoT gadgets utilizing SDN-based flow-level telemetry joined with AI. The authors distinguished traffic streams that can all things considered portray the organization conduct of IoT gadgets and their states, for example, booting, client connection, or inactive. They showed how they could balance the compromise between cost and execution of our plan and exhibited how administrators can utilize it to recognize IoT social changes (both authentic and malevolent).

There exists various IoT network safety, explicitly, the cutting edge of the current position and likely future bearings, the significant countermeasures against IoT attacks, and the applications in the industrial level. Moreover, the authors have presented and talked about a potential four-layered [13] IoT network protection foundation and a scientific classification of attacks on IoT network safety. The special arrangement actualized is painstakingly picked because of equipment limitations of preparing and memory on IoT gadgets just as the limited cost of information transfer charged by ISP. Usage is done to build up gadget association with cloud parts for verifying gadgets [14] to forestall gadget clone attacks. Post- fruitful verification information is scrambled to forestall touchy information openness. The danger present in IoT which is gadget dangers, application administration assaults, Web interface assault, etc. decreases the framework exhibitions and dependability. A forswearing-of-administration assault (DoS assault) [15] endeavors to deliver a

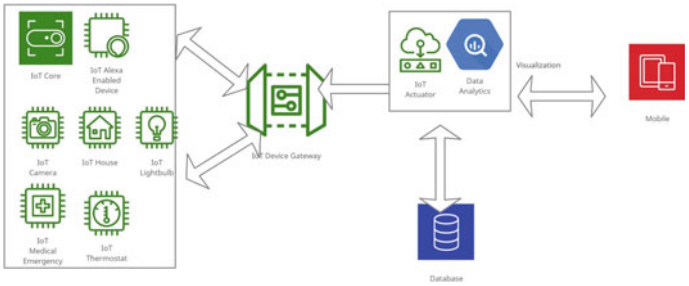


Fig. 1 General architecture of IoT

machine or organization asset inaccessible to its clients by upsetting administrations of a host associated with the Web. Essentially, it makes it unavailable for the clients. On account of a disseminated refusal of administration assault (DDoS), approaching traffic flooding an objective begins from numerous sources, making it hard to stop the digital hostile by essentially obstructing a solitary source. DoS and DDoS assaults can contrarily influence a wide reach IoT applications, causing genuine interruptions for utility administrations and assembling offices. Perpetual Forswearing of Administration (PDoS): Lasting refusal of administration assaults (PDoS), otherwise called plashing, is an assault that harms the gadget so gravely that it requires substitution or reinstallation of equipment. BrickerBot, coded to misuse hard-coded passwords in IoT gadgets and cause lasting refusal of administration, is one such illustration of malware that could be utilized to cripple basic gear on a processing plant floor, in a wastewater treatment plant, or in an electrical substation (Fig. 1).

Today, IoT devices are technologically advanced with a lot of algorithms, models, and trained datasets. Because we are progressing higher with deep learning methods, a lot of these devices are well-trained. However, these models are vulnerable to adversarial attacks, which pose a huge risk to the normal application of deep learning technology.

3 Adversarial Attacks on IoT Devices

This paper studies the strength of current adversarial attacks on wireless IoT devices with a small amount of stealth and passive field measurements along with unsupervised learning techniques. For wireless IoT cameras, for example, a very important yet simple vulnerability could result from signal leakage and monitoring of wireless measurements made by external attackers and then planning intrusion to avoid detection. Researchers have found that 60% of intrusion attacks consider the presence of a camera nearby while choosing targets. Pen testing is typically performed by a testing proficient to identify security dangers implied in a framework. Infiltration testing [16] is broadly utilized for testing both organization and programming; however, in some

place, it neglects to make IoT safer. The attacker will use localization techniques to locate wireless transmitters belonging to third parties, a type of attack sometimes referred to as adversarial localization. This paper investigates the practical intricacies of adversarial localization in a realistic scenario. Given the direct implications on IoT security systems, we are very interested in passive attacks against wireless cameras; e.g., in order to establish the camera's location, a hacker, whether a human or a robot, traveling inside the perimeter of the area where IoT devices are installed, stealthily collects the received signal strength of the Wi-Fi camera and then uses the RSS data. Unlike traditional attacks, the adversary does not need to communicate directly with the target; only being required to measure RSS, the attacker can be easily made to exploit the existing RSS-based localization algorithms by walking by the IoT device and applying existing COTS Wi-Fi receivers. Even though advanced localization algorithms (including time, fingerprints, AoA) have improved efficiency and are more accurate, they still require active communication as opposed to passive communication with the target or bulky infrastructure with a lot of antennas/gear. Our proposed approach also helps the attacker identify and distinguish high-quality and noisy measurements accurately through unsupervised learning techniques. There is also a need to study security measures against such attacks that add noise to RSS signals both in space (spatial) and in time (temporal) domains. We find that these defenses are moderately effective against the basic attack, but these have less impact when applying advanced machine learning techniques. However, attackers with sophisticated skillset and expertise may deploy countermeasures in the form of stationary receivers, thus ensuring more robust signals. Furthermore, current defenses are lacking in combating these localization attacks. IoT devices, like video cameras deployed in households, pose serious safety risks if exposed to vulnerability. The threat increases as adversaries leverage more advanced algorithms, or more advanced hardware. We should develop effective defenses against these attacks to maximize the security of wireless IoT devices [17].

4 Methodology

Vulnerability Detection Using CVE

Normal weaknesses and openness dataset (CVE) is an enormous accessibility of public wellsprings of weaknesses for both equipment and programming information. We examine the derived weakness and add in for certain proposed answers to further fortify the security of those IoT gadgets and their sorts. The strong motivation behind this characterization is to test against every one of the weaknesses in IoT gadgets with their exhibition and dependability of those CVE information for additional security. As a countermeasure for the recorded weaknesses, we would alleviate them against the digital assaults and produce the rundown of measures to diminish the opportunity of digital assault in such gadgets as everyday there are a ton of safety penetrates found in such arising innovation by abusing the weakness.

These weaknesses when found in the IoT gadget will be hard to be identified as they perform capacities like that of the genuine framework because of which the outcome against the weaknesses in the customer side additionally increments. For example, the espresso machine [18] when associated with the Web can take the casualty's bank card subtleties. For the classification of the vulnerabilities in IoT devices from CVE analysis, we have considered various IoT home devices, health care devices, network devices, VR devices. Vulcan has its own CVE database with .csv extension through which the remote attackers can cause a denial of service attacks (DOS) through these DNS queries. To perform this vulnerability detection, we have used Nmap for scanning the IP address and the ports. Nmap commands have been integrated with NSE scripts wherein Nmap Vulcan and vulners have been used together to perform the vulnerability check on IP of the devices (Table 1).

After analysis through the CVE [19], various IoT devices are measured against the vulnerabilities range. Nmap basically helps in deterring the service information of the port accessed.

Table 1 Comparison analysis of various IoT devices

Vendor name	Vulnerabilities description	CVSS score
Astra Telecom	DoS attack	7.5
Google Map Travel Project	XSS, CSRF	6.2
Apple iPhone	Overflow memory corruption	4.3
OnePlus smart mobiles	Gain privileges, overflow	10
Google home	Obtain information	3.3
Samsung TV	DoS, overflow	7.8
Smart home	Disclosure of information	10
macOS	Overflow memory corruption	9.3
Hikvision-IP cameras	Execute code, overflow	7.5
iWatch	DoS, overflow, memory corruption, gain privilege	7.2
Nespresso Prodigo coffee machine	File system modification	4.8
D-link fiber network	Information disclosure	7.5
Ring DoorBot	Information disclosure	2.1
Samsung smart things	Overflow	6.9
Wink Android	Information disclosure	5.0
Apple Watch	DoS, overflow	7.5
Fitbit	Information disclosure	3.3
Arduino	Modification of file system	3.3
Raspberry Pi	Execute code	10

Defense against adversarial attacks on IoT devices

A better defense in terms of security is to reconfigure Wi-Fi hardware or install signal reflectors to prevent wireless signals from reaching elsewhere outside the wall/room. Although such an approach appears sound in principle, it has the disadvantage of being very expensive, limited in the camera's ability to communicate with the AP, and is almost never feasible to deploy. Our defenses, instead, can focus on different measures that can be taken to generate disturbances to RSS signals both in the time and frequency domains, by modifying the Wi-Fi transmit power or by adding additional "cameras."

Control measures and effects against adversaries

While current defenses demonstrate some promise in regards to localization attacks, they are not as effective when the attacker uses advanced machine learning methods, such as pruning. An attacker may even deploy countermeasures in additional stationary receivers to reduce noise to the temporal domain. They use a stationary Wi-Fi sniffer for monitoring the RSS within the attack area (referred to as an anchor), and an attacker can theoretically remove the randomization contribution to the RSS based on the change that the anchor sees. Naturally, this is only applicable if the anchor and receiver share the same channel conditions. As RSS performance depends heavily on the choice of anchor, a good anchor will essentially build a comprehensive RSS trace by capturing the vast majority of the victim's traffic for deriving power. It is recommended to use anchors with the lowest loss rate. They compare the localization error using some basic and advanced attacks, with and without anchors, and it is evident that when power randomization is used, anchors greatly increase localization accuracy. The accuracy improvement becomes negligible when using data pruning, but the pruning overhead downgrades substantially.

Cloud security threats

Interfaces and APIs that are insecure: In cloud computing, APIs enable communication between applications and servers. A weak API set makes it more likely for security threats to occur.

Vulnerabilities of cloud-based services: Malicious attackers who violate cloud contracts often use brute force attacks, Trojans, SQL injection, botnets, phishing, e-mail spoofing, etc. Neither the service providers nor the hackers can generate or block attacks on their networks, so the owners of those networks will not be able to detect attacks on their networks.

Intentional attackers: They work on cloud platforms with users' data and resources at their disposal and manipulate that data in some way. IoT depends on cloud computing because it stores data and resources, so it is critical to the Internet of Things. Both technologies, however, are at risk of being hacked. It is a major problem. As IoT and cloud computing technologies advance rapidly, factual security is more necessary than ever. Keeping customers' trust and integrity is crucial.

Bluetooth Hacking

Bluetooth hacking is a procedure which stays new to numerous individuals even with the headway in the innovation. We can explore three inbuilt tools in Kali Linux that will help us get details of the devices. Bluetooth is very commonly used. It uses 2.4 GHz frequency bands, just like Wi-Fi. The properties of these both are different though. But, both of them are hopping frequencies. So, when you have two devices using Bluetooth, they both have an algorithm that they understand and that shift the frequency many times per second. This means you cannot listen to it on one frequency because it keeps changing and hopping to different frequencies. Another property of Bluetooth is that rather than generating keys every single time like Wi-Fi does, (this makes it easier for people to join and leave a Wi-Fi network), it negotiates the key only once and stores the key and will refer to it every time it sees the device. This means that the sniffer has to be there when the Bluetooth device gets connected for the first time; otherwise, they will not really get any valuable information. It would be impossible to jump into the conversation like you can with Wi-Fi. Despite all of this, you can track Bluetooth devices near you and read and write some characteristics. So, reconnaissance is useful because you get to list out some vulnerabilities. With Bluetooth, it is very important to gather information first so that we can circle on to some vulnerability that we can use.

- hciconfig Hciconfig hci0 up Hcitool scan
- Hci tool is used to configure, scan, enquire. This is helpful to gather a lot of information.

Hcitool scan: It uses the Bluetooth interface to find out Bluetooth devices around and gives their MAC address so that we can perform additional scans.

Hcitool <mac> name to get the name SDP tool:

Queries on Bluetooth devices. The permissions are available. Sdptool browse <mac>

Btscanner

A GUI device for finding Bluetooth devices. It looks similar to the Kismet interface. Inquiry scan—finds devices with Bluetooth and allows us to connect. It also gives a list of things that each device is vulnerable to. Bluetooth adapters can be used to search for more devices.

Many Bluetooth devices do not have randomized MAC addresses. This can be used. One can track where the person is from time to time. Opt out of Bluetooth tracking, switch off your Bluetooth when not used.

Bettercap

Successor to Bettercap. Specializes in reconnaissance and man-in-the-middle attacks. This is a Wi-Fi hacking tool, but it also has a lot of Bluetooth hacking techniques. Because they both are very similar, you can use the modules interchangeably. We can dig into nearby Bluetooth devices and get info like the model of the phone and what battery percent the phone is at and also can write something into it.

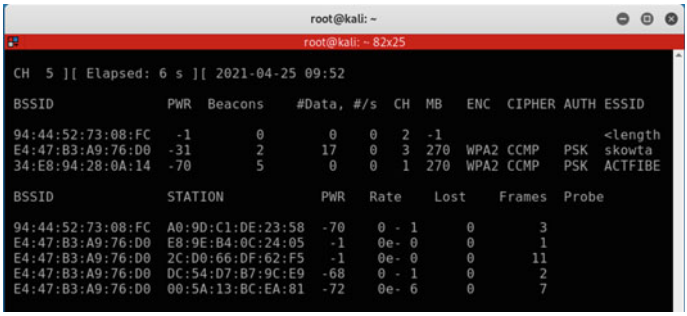


Fig. 2 Monitor mode

- Apt update
- Apt install bettercap Bettercap Commands:
- ble.recon on.

Shows the devices and the MAC addresses. ble.recon on (turn your Bluetooth on). Starts looking for Bluetooth devices in range. Stop the scan with ble.recon off. Show results with: ble.show Ble.enum <mac>. If you find a flaw in the Bluetooth technology, you can use that to connect and exploit the device.

Wi-Fi Hacking

Process of Packet Sniffing: captures packets that are in range and displays detailed information about the networks around us.

Tool used: Airodump-ng. Enable monitor mode on your wireless interface (Fig. 2).

Airodump-ng wlan0

Wi-Fi Bands

Frequency range that can be used. They determine the channels that can be used. Clients need to support band used by the router so as to communicate with it.

Different Wi-Fi bands are:

- a—5Ghz
- b, g—2.4Ghz
- n—both 5 and 2.4 Ghz ac—lower than 6Ghz

Airodump-ng wlan0 gives 2.4 Ghz frequency bands. Showing various bands also depends on what wireless adapters one is using. Airodump-ng–band a wlan0.

Airodump-ng–band abg wlan0

Targeted Packet Sniffing

Airodump-ng–bssid <bssid> –channel <channel> —write test wlan0

See Fig. 3.

Second section consists of the clients or the devices that are connected to that network. The stations are the clients that are connected to that particular Wi-Fi. The

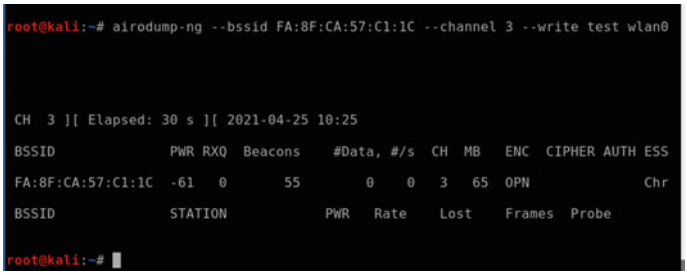


Fig. 3 Packet sniffing

power, rate, and lost packets can also be seen over here. Now, these captured files can also be analyzed in wire shark or other tools. Cap file contains everything that was sent to and from the network. Anything that they have to do will have to be sent through the router. The packets will be encrypted because the Wi-Fi router has WPA2 encryption. Although from this cap file, we have absolutely no idea as to what the sent packets were, we can still get to know the device manufacturers of the clients present. We can see that one of the clients is an Apple device. You can also see a Xiaomi device and a Realme device.

Deauthentication Attack

Disconnecting any client from the network. This kind of attack works on any encrypted network—including WEP, WPA, and WPA2. You do not need to know the network key to perform this attack.

airreplay-ng–deauth #deauthpackets–a networkmac–c targetmac interface

We pretend to be the client and send a message saying we want to disconnect from the router. We then pretend to be the router and allow this to happen. This way you get disconnected.

WEP Cracking: wired equivalent privacy old encryption algorithm used: RC4 IV + key = key stream. So, the random initialization vector IV is very small. It is only 24bits. Then, IV will repeat in busy networks, and hence, it is vulnerable to statistical attacks.

To crack WEP:

- Capture a lot of packets using airodump-ng.
- Analyze the packets and crack the key using aircrack-ng.

Airodump-ng–bssid <> –channel <> –write basicwep wlan0

#Data column has a different number of packets that contain an IV packet, so the higher the number of packets, the higher IVs you can have, and you can crack WEP. Now, run aircrack-ng against this capture file. You will now get the ASCII password and the key. You can connect using the key.

Illegitimate Authentication Attack

If the data frame moves slowly, you will have to wait for hours to connect to the network which is why you can perform a fake authentication attack so that data frames are sent out. We need to make the access point to generate more frames. So, we try to associate with it. Associating is not connecting. We are merely just saying that we want to communicate, and we exist to the AP. Auth column changes to open if your network is WEP. The number of clients associated with the network also changes because of the fake authentication.

ARP Request Replay Attack

- Wait for an ARP packet.
- Capture the ARP packet and replay it. This results in producing another packet with a new IV.
- We repeat this till we have enough IVs to crack the key.

First, fake authentication attack:

Aireplay-ng–fakeauth 0–a <target> –h <wlan0 uspec> wlan0 Then: Aireplay-ng–arpreplay–b <target> –h <wlan0uspec> wlan0.

After you have enough packets, c aircrack-ng arpreplay-01.cap. You will manage to get the key of WEP through this.

WPA/WPA2 Cracking

Both of these can be cracked using the same methods. These are much more secure, and each packet is encrypted using a temporary key.

WPA—TKIP; WPA2—CCMP

WPS: If this feature is enabled in your device, one can connect to the access point without even trying to get the password. WPS is a feature that can be used with WPA and WPA2. Authentication is done using an 8-digit pin. All the possible pins can be tried in a relatively small time. However, this only works if the router is configured not to use push button authentication (PBC). So, it should have WPS feature enabled, and it should be misconfigured to not use PBC.

Hacking WPA and WPA2 without a word list:

wash—interface wlan0

This shows you the list of interfaces that have WPS enabled. First, do the fake authentication attack. Set the association time to 30 s.

Aireplay-ng–fakeauth 30–a <target> –h <wlan0 uspec> wlan0

But before executing this, you have to run Raever. This is the program that will brute force various pins and will tell you the right pin.

raever–bssid <target> –channel <channel number> –interface wlan0–vvv–no associate.

This will give you the pin. Usually, a lot of routers have default 12,345,678 as their pin. Capturing the handshake: Now, the weaknesses in WEP were all fixed and taken into consideration when making up WPA and WPA2. Packets now contain no useful data. The only ones that contain something useful are the handshake packets—these are the packets that are sent by the client when it connects to the network. One

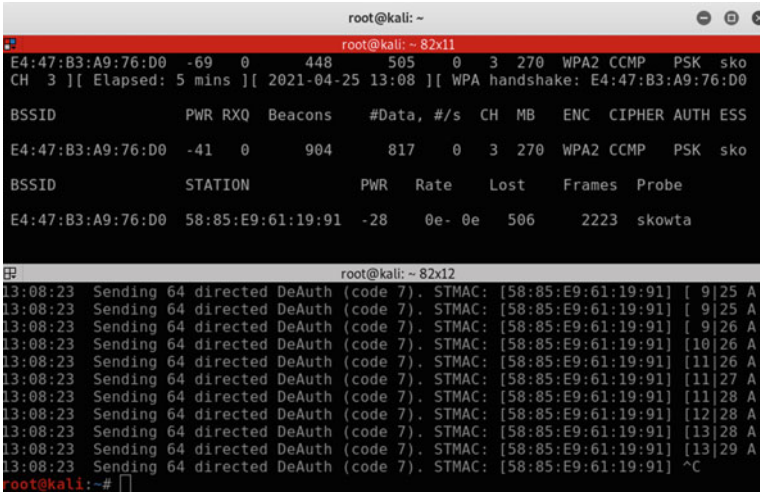


Fig. 4 ARP attack

must now wait for a new client to connect to the network to capture the handshake. Alternatively, we can make a deauthentication attack. So that, a client disconnects for a short period of time and connects back. This way, the handshake can be captured (Fig. 4).

Aireplay-ng–deauth 4–a <mac of target> –c <client disconnecting> wlan0

We have now captured the handshake, and it is in the handshake01.cap. The handshake does not contain the data that recover the key. It will only help us check if a key is valid or not. So now, we create a wordlist that will work along with the handshake and will tell if the password is recovered or not.

Aircrack-ng handshake01.cap–w test.txt (name of word list)

This gives you the password (Table 2).

5 Conclusion and Future Work

In this paper, we have discussed various vulnerabilities that are found in various IoT devices irrespective of the domain, and we have analyzed that most of the tested devices pose major threat to different vulnerabilities depending on their usage. We have analyzed and showed how Wi-Fi attacks can take place. Bluetooth hacking was also an IoT device that we chose to scan. From our experiments, we have realized that most of the attacks are only due to small misconfigurations or the default settings that people do not tend to change. During inference, we provide insight into a defense mechanism systematically increasing the uncertainty of the adversary. The results help us discover new ways to defend the IoT networks through deep learning. IoT security has a huge scope. Combining ML with IoT for its security would be very

Table 2 IoT security attacks and vulnerabilities and techniques used

Attack type	Vulnerabilities and issues	Techniques used
Physical attack	Inserting an additional storage device to spread malicious code	From within the company
Encryption attack	Takes control of the system	Unencryption of IoT devices
Denial of service	Collects signal strength from camera using RSS data	Applying machine learning techniques
Botnets	Auto-scan and get in through passwords	Detection techniques or traffic monitoring
Man-in-the-middle	Session hijacking	Through DNS queries
Ransomware	E-mail phishing	Scanning for IP address
Eavesdropping attack	Inflow/outflow of the data packets through access points	Deauthentication and fake authentication
Brute force attack	Due to random guesses	Two factor authentication
Software device failure	Buffer overflows	Verification techniques
Node tampering attack	Manual replacement	Various tamper proof techniques can be used
Malicious code injection	Due to lack of software integrity checks	Input validation could be done
Unauthorized access	vulnerabilities found in software and hardware	Generation of session key on a timely basis
Malicious node insertion	Weak encryption schemes	Encryption scheme can be used

useful in the near future. We can also test IoT devices based on their performances through model-based testing methods which could be implemented along with fuzzy techniques—these techniques learn based on the history of detected vulnerabilities to enhance the security of IoT devices, therefore protecting them from various cyber security attacks.

References

1. Balachandran V(2020) On the need for automatic knowledge management in modern collaboration tools to improve software maintenance. In: 2020 IEEE International conference on software maintenance and evolution (ICSME). IEEE, pp 722
2. Cinque P (2019) Microservices monitoring with event logs and black box execution tracing. In: IEEE transactions on services computing
3. Alsmadi I, Mira F (2020) IoT security threats analysis based on components, layers and devices. Am J Inno Entrepreneur 1(1):1–10
4. Chatterjee PL (2020) Software-related slack chats with disentangled conversations. In: Proceedings of the 17th international conference on mining software repositories, pp 588–592
5. Cusack B, Tian Z (2017) Evaluating IP surveillance camera vulnerabilities
6. "Slack api," Slack technologies
7. Sirt L (2020) Logz.io deepens observability platform with new release of infrastructure monitoring, sees significant customer adoption. Globe News Wire, 13 April 2020. Available:

https://www.globenewswire.com/news-release/2020/04/13/2015079/0/en/Logz-io-Deepens-Observability-Platform-with-New-Release-of-Infrastructure-Monitoring-Sees-Significant-Customer-Adoption.html
8. Liew Z (2018) Understanding and using REST APIs. Smashing Magazine, 17 Jan 2018
9. Munonye K, Martinek P (2018) Performance analysis of the Microsoft .net-and Java-based implementation of REST web services. In: 2018 IEEE 16th internationalsymposium on intelligent systems and informatics (SISY). IEEE
10. Radanliev P, De Roure D, Cannady S, Montalvo RM, Nicolescu R, Huth M (2019) Economic impact of IoT cyberrisk-analysing past and presentto predict the future developments in IoT risk analysis and IoT cyber insurance
11. Stray B (2020) Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and slack. J Syst Softw 170:110717
12. Ortu M, Murgia A, Destefanis G, Tourani P, Tonelli R, Marchesi M, Adams B (2016) The emotional side of software developers in JIRA. In: 2016 IEEE/ACM 13th working conference on mining software repositories (MSR), pp 480–483
13. Sharma S (2016) Mastering microservices with java., Packt Publishing Ltd
14. Soni A, Ranga V (2019) API features individualizing of web services: REST and SOAP. In: International journal of innovative technology and exploring engineering (IJITEE), pp 2278–3075
15. Tawalbeh LA, Muheidat F, Tawalbeh M, Quwaider M (2020) IoT Pri- vacy and security: challenges and solutions. Appl Sci 10(12):4102
16. https://www.i-scoop.eu/internet-of-things-guide/
17. Li Z, Xiao Z, Zhu Y, Pattarachanyakul I, Zhao BY, Zheng H (2018) Adversarial localization against wireless cameras. In: Proceedings of the 19th international workshop on mobile computing systems and applications
18. Arcuri A, Campos J, Fraser G (2016) Unit test generation during software development: evosuite plugins for Maven, IntelliJ and Jenkins. In: IEEE International conference on software testing, verification and validation (ICST), pp 401–408
19. George B, Williams L (2003) An initial investigation of test driven development in industry. In: Proceedings of the 2003 ACM symposium on applied computing, pp 1135–1139

Asymmetric Cryptosystem for Triple Color Images Using 6D Hyperchaotic System and Equal Modulus Decomposition in Fractional Transform Domain

Dhanesh Kumar, Anand B. Joshi, and Sonali Singh

Abstract A novel scheme for security of triple color images using a 6D hyperchaotic system and equal modulus decomposition (EMD) in a two-dimensional fractional discrete cosine transform domain (2D-FrDCT) is presented in this paper. To improve the reliability of the asymmetric cryptosystem, a 6D chaotic map is utilized to generate three public keys as well as three secret keys. The public keys are used to encode digital images into the 2D complex matrix. After 2D-FrDCT in the encoded complex matrix, the EMD is executed to obtain the powerful trapdoor function. Finally, the secret keys are used to scramble the private key and a XOR operation is operated to get encrypted images. The private key is used during decryption. The results of experiment and the security analysis are discussed to check the capability and validity of the presented scheme.

Keywords Color image · 2D Fractional discrete cosine transform · Equal modulus decomposition · 6D Hyperchaotic system.

1 Introduction

In this era, the fast expansion of the Internet and computer technology creates the issue of the security of digital data (image, audio, video, etc.). Nowadays, for dealing with such issues, most of the researcher uses technology which is based on cryptography or steganography or combination of both.

In cryptographic techniques, some usual image encryption schemes use chaotic maps [1, 2], whose role is very crucial in the protection of digital data due to their sensitive nature to initial and control parameters and pseudo-randomness. These features make chaotic maps a first choice for developing cryptosystems as compared to other techniques. Most of the schemes [4–7] are symmetric encryption algorithm. Asymmetric encryption schemes have attracted much attention in the past few years. Many authors [8–12] have been done much work in asymmetric cryptosystem.

D. Kumar (✉) · A. B. Joshi · S. Singh
Department of Mathematics and Astronomy, University of Lucknow, Lucknow, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_32

Chen et al. [8] proposed an public-key approach for the security of digital image using chaotic Ushiki map and EMD in fractional Fourier transform domain. In this method, the public-key cryptosystem is obtained by applying EMD as a trapdoor function. Yadav et al. [10] proposed an asymmetric cryptosystem for digital image using fractional Hartley transform. For the design of public-key cryptosystem, this technique used amplitude and phase-truncation approach. Also in [12], Huang et al. also presented public-key algorithm based on the amplitude and phase-truncation technique.

Yao et al. [13] presented a public-key technique for digital image using singular value decomposition (SVD). In this technique, the red and green components of the digital color image are treated as real and imaginary parts of the complex function and then this output is again considered as real and imaginary parts with the blue component of the digital color image.

This manuscript proposes a new asymmetric scheme for digital triple color image security using the 6D chaotic map and EMD in the 2D-FrDCT domain. For encryption of three color images, we are using the indexed format of the color images. A 6D chaotic map is used for generating public and secret keys.

The outline of the manuscript is given as. Section 2 is about 2D-FrDCT, 6D hyperchaotic system, and EMD. Section 3 presents the encryption and decryption scheme. The results based on the experiment are discussed in Sect. 4. Section 5 describes the security analysis. The comparison of the presented cryptosystem with other existing cryptosystems is given in Sect. 6. At last, Sect. 7 concludes the presented cryptosystem.

2 Preliminary Knowledge

This section provides the preliminary knowledge of tools that are used in the encryption process. The tools are 2D-FrDCT, 6D hyperchaotic system, and equal modulus decomposition.

2.1 2D-fractional Discrete Cosine Transform

The general form of the discrete cosine transform (DCT) is the fractional discrete cosine transform. Ahmed et al. [14] presented the DCT which is a most useful function in image analysis. The 2D-DCT of the 2D-array $S_{s,t}$ of size $S \times T$ is defined by Eq. 1,

$$S'_{u,v} = \beta_u \beta_v \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} S_{s,t} \cos \frac{\pi(2s+1)u}{2S} \cos \frac{\pi(2t+1)v}{2T}, \tag{1}$$

where $0 \leq u \leq S-1$, $0 \leq v \leq T-1$, $0 \leq s \leq S-1$, and $0 \leq t \leq T-1$,

$$\beta_u = \begin{cases} \frac{1}{\sqrt{S}} & \text{if } u = 0 \\ \frac{2}{\sqrt{S}} & \text{otherwise} \end{cases}, \beta_v = \begin{cases} \frac{1}{\sqrt{T}} & \text{if } v = 0 \\ \frac{2}{\sqrt{T}} & \text{otherwise.} \end{cases}$$

where the size of the matrix $S'_{u,v}$ of Eq. 1 and the matrix $S_{s,t}$ is same. In matrix form, it is defined by Eq. 2,

$$S'_{u,v} = C_c S_{s,t} \tag{2}$$

where C_c is given by Eq. 3,

$$C_c = \left\| \frac{1}{\sqrt{S}} \alpha_u \cos\left(2\pi \frac{(2s+1)u}{4S}\right) \right\|, \tag{3}$$

where $\|.\|$ represents $S \times S$ matrix, $0 \leq s, u \leq S-1$ and $\alpha_0 = 1, \alpha_u = \sqrt{2}$ for $u \geq 1$.

The eigenvalue substitution and the eigen-decomposition are used to derived the FrDCT of Eq. 3, which is defined by Eq. 4,

$$C_c = A(\text{Diag})A^* = \sum_s A_s e^{i\phi_s}, \tag{4}$$

where A represents the unitary matrix, A^* represents the conjugate transpose of A, and (Diag) is the diagonal matrix.

The FrDCT matrix C_a is given by Eq. 5,

$$C_a = A(\text{Diag})^a A^*, \tag{5}$$

where a is an order of FrDCT.

For an image $I_{s,t}$, 2D-FrDCT of orders a and b is defined in Eq. 6,

$$I'_{u,v} = C_a I_{s,t} C_b^T. \tag{6}$$

The 2D-IFrDCT is defined by Eq. 7,

$$I_{s,t} = C_{-a} I'_{u,v} C_{-b}^T. \tag{7}$$

2.2 The 6D Hyperchaotic System

A hyperchaotic map is defined as chaotic properties with minimum two positive Lyapunov exponents. Grassi et al. [16] presented a 6D chaotic map using two same 3D-Lorenz chaotic maps, which is defined in Eq. 8,

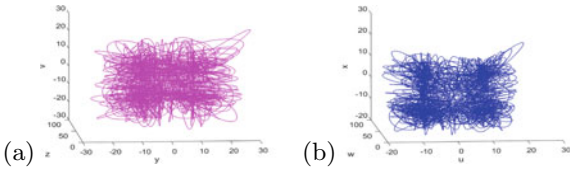


Fig. 1 6D hyperchaotic attractors of the system 8: **a** in the (y, z, v) space and **b** in the (u, w, x) space

$$\begin{cases} \dot{u} = a_1(v-u) \\ \dot{v} = a_2 u - v - uw + r_1(x-y) \\ \dot{w} = uv - a_3 w \\ \dot{x} = a_1(y-x) \\ \dot{y} = a_2 x - y - xz + r_2(u-v) \\ \dot{z} = xy - a_3 z \end{cases} \tag{8}$$

where a_1, a_2, a_3, r_1, r_2 are the parameters of the 6D chaotic map. When $a_1 = 10$, $a_2 = 28$, $a_3 = 8/3$, and $r_1 = r_2 = 0.05$ system 8 give the four-wings as displayed in Fig. 1. In the presented scheme, a_1, a_2, a_3, r_1, r_2, u, v, w, x, y, and z are considered to be the secrete keys, to generate the keystream.

2.3 *Equal Modulus Decomposition*

In a complex number system, a complex number is represented as a coordinate in 2D system. The EMD defined in Cai et al. [17] is a kind of asymmetric cryptosystem, which is used to separate the complex number into two complex numbers with equal modulus. If one complex number in the 2D system is taken as a vector $Z(f, g)$, then it is decomposed into two parts with equal modulus as displayed in Fig. 2.

Let $Z(f, g)$ in Fig. 2 is an image, then $C_1(f, g)$ and $C_2(f, g)$ are expressed by Eqs. 9 and 10,

$$C_1(f,g) = \frac{A(f,g)/2}{\cos[\alpha(f,g) - \beta(f,g)]}.\exp\{i\beta(f,g)\}, \tag{9}$$

$$C_2(f,g) = \frac{A(f,g)/2}{\cos[\alpha(f,g) - \beta(f,g)]}.\exp\{i[2\alpha(f,g) - \beta(f,g)]\}, \tag{10}$$

where the functions $\alpha(f, g)$, $\beta(f, g)$, and $\phi(f, g)$ are given below,

$$\alpha(f,g) = \text{angle}\{Z(f,g)\}, \quad \beta(f,g) = 2\pi \times \text{rand}(f,g),$$

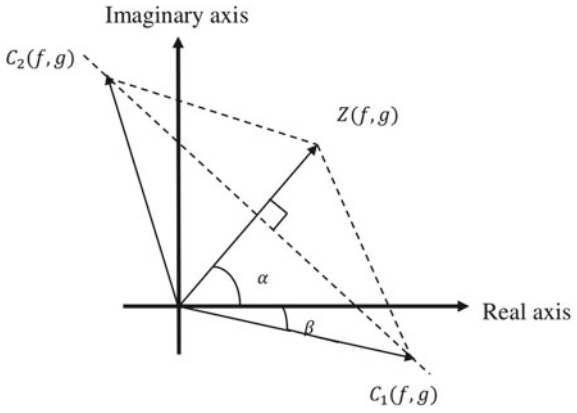


Fig. 2 Pictorial view of the complex number in EMD

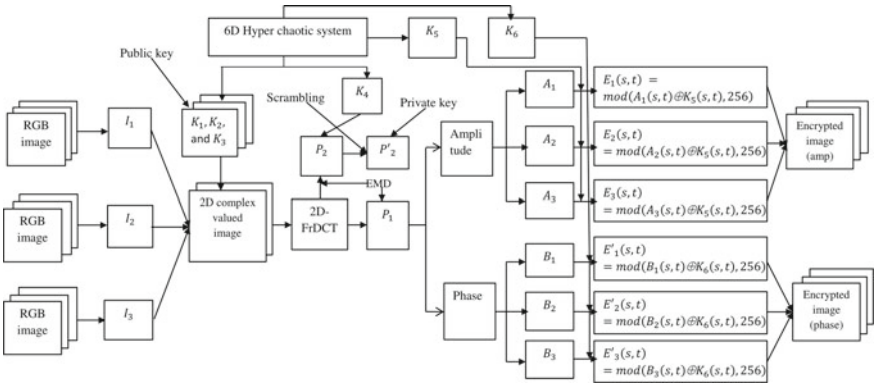


Fig. 3 Pictorial representation of the proposed asymmetric cryptosystem

where the function rand(f, g) generates a random matrix whose element is distributed normally in the interval [0, 1] and angle finds the argument of the complex number. The modulus of $Z(f, g)$ is $A(f, g)$, defined as $A(f, g) = |Z(f, g)|$.

3 Triple Color Image Encryption and Decryption Process

Figure 3 displays the pictorial view of our asymmetric cryptosystem. In the first step of algorithm, the indexed formats of the color images are used and then use 2D-FrDCT for encryption and decryption.

3.1 Key Generation

The required keys are generated in four steps.

Step 1: Iterate system 8 to $(S \times S + p)$-times. Exclude initial p values to avoid bad effects and get random sequences $u = \{u_1, u_2, u_3, ..., u_{S\times S}\}$, $v = \{v_1, v_2, v_3, ..., v_{S\times S}\}$, $w = \{w_1, w_2, w_3, ..., w_{S\times S}\}$, $x = \{x_1, x_2, x_3, ..., x_{S\times S}\}$, $y = \{y_1, y_2, y_3, ..., y_{S\times S}\}$ and $z = \{z_1, z_2, z_3, ..., z_{S\times S}\}$, respectively each of size $\max\{1 \times S^2\}$.

Step 2: Now, Convert the sequences u, v, w, x, y, and z into integers as:

$$\begin{cases} U = \text{floor}(u \times 10^{15}) \text{mod } S \times S, \\ V = \text{floor}(v \times 10^{15}) \text{mod } S \times S, \\ W = \text{floor}(w \times 10^{15}) \text{mod } S \times S, \end{cases} \quad \begin{cases} X = \text{floor}(x \times 10^{15}) \text{mod } S \times S, \\ Y = \text{floor}(y \times 10^{15}) \text{mod } S \times S, \\ Z = \text{floor}(z \times 10^{15}) \text{mod } S \times S, \end{cases} \tag{11}$$

where floor(o) returns o to the nearest integers less than or equal to p and mod defines modulo function.

Step 3: Sort the sequences given in Eq. 11 and get six sorted sequences U', V', W', X', Y', and Z'. Find the value of positions of U', V', W', X', Y', Z' in U, V, W, X, Y, Z and write down the change position value, i.e., $O' = \{O'(s) : s = 1, 2, 3, ..., S \times S\}$, $P' = \{P'(s) : s = 1, 2, 3, ..., S \times S\}$, $Q' = \{Q'(s) : s = 1, 2, 3, ..., S \times S\}$, $R' = \{R'(s) : s = 1, 2, 3, ..., S \times S\}$, $S' = \{S'(s) : s = 1, 2, 3, ..., S \times S\}$, $T' = \{T'(s) : s = 1, 2, 3, ..., S \times S\}$, where $U(O'(s)) = U'(s)$, $V(P'(s)) = V'(s)$, $W(Q'(s)) = W'(s)$, $X(R'(s)) = X'(s)$, $Y(S'(s)) = Y'(s)$, and $Z(T'(s)) = Z'(s)$.

Step 4: Now, the sequences O', P', Q', R', S', and T' are transform into matrices M_1, M_2, M_3, M_4, M_5, and M_6, respectively, each of size $S \times S$ and generate keys K_1, K_2, K_3, K_4, K_5, and K_6 as:

$$\begin{cases} K_1 = (M_1) \text{mod } 256, \\ K_2 = (M_2) \text{mod } 256, \\ K_3 = (M_3) \text{mod } 256, \end{cases} \quad \begin{cases} K_4 = (M_4) \text{mod } 256, \\ K_5 = (M_5) \text{mod } 256, \\ K_6 = (M_6) \text{mod } 256. \end{cases} \tag{12}$$

3.2 Algorithm for Triple Color Image Encryption

The pictorial view of the presented cryptosystem is displayed in Fig. 3. The details of the procedure are given in Step 1 to Step 8.

Step 1: The three color images CI_1, CI_2, and CI_3 of size $S \times S$ are converted into indexed images I_1, I_2, I_3 and get corresponding colormaps cmap1, cmap2, and cmap3.

Step 2: In this step, the indexed images I_1, I_2, and I_3 are combined with public keys K_1, K_2, K_3 and we get combined images C_1 and C_2, each of size $(\frac{3S}{2} \times 2S)$.

$$\begin{cases} I_3' &= I_3(1:S/2, 1:S) \\ I_3'' &= I_3((S/2)+1:S, 1:S) \\ K_3' &= K_3(1:S/2, 1:S) \\ K_3'' &= K_3((S/2)+1:S, 1:S) \end{cases} \tag{13}$$

$$\begin{cases} C_1 &= [I_1,\ K_1;\ I_3',\ K_3'] \\ C_2 &= [I_2,\ K_2;\ I_3'',\ K_3''] \end{cases} \tag{14}$$

Step 3: Now, we normalize the image C_2 in between [0, 1] and form a complex valued function C_m given by Eq. 15,

$$C_m = C_1 \cdot \exp\left(i \cdot \pi \cdot \frac{C_2}{255}\right) \tag{15}$$

where $\exp(x)$ is the exponential function of x and i is complex number satisfy $i^2 = -1$.

Step 4: The complex matrix C_m is encrypted by using 2D-FrDCT, we get C', which is given by Eq. 16,

$$C' = C_a C_m C_b^T, \tag{16}$$

where a and b are secret keys.

Step 5: The output of the cosine map is separated into two independent parts P_1 and P_2 by using EMD. Now, the pixel position scrambling operation is performed according to K_4 in P_2 to obtain P_2', which is a private key.

Step 6: The complex mask P_1 is separated into amplitude part $P_{1\text{amp}}$ and phase part $P_{1\text{phase}}$. three complex matrices A_1, A_2, and A_3, each of size $S \times S$.

Step 7: Now, the amplitude part $P_{1\text{amp}}$ and phase part $P_{1\text{phase}}$ are decomposed into three matrices A_1, A_2, A_3 and B_1, B_2, B_3, respectively, each of size $S \times S$.

Step 8: Finally, using the secret keys K_5 and K_6, the pixels substitution using XOR operation is employed in A_1, A_2, A_3 and B_1, B_2, B_3, respectively, to get the encrypted image components.

$$\begin{cases} E_1(s,t) = \text{mod}(A_1(s,t) \oplus K_5(s,t), 256) \\ E_2(s,t) = \text{mod}(A_2(s,t) \oplus K_5(s,t), 256) \\ E_3(s,t) = \text{mod}(A_3(s,t) \oplus K_5(s,t), 256) \end{cases} \tag{17}$$

$$\begin{cases} E_1'(s,t) = \text{mod}(B_1(s,t) \oplus K_6(s,t), 256) \\ E_2'(s,t) = \text{mod}(B_2(s,t) \oplus K_6(s,t), 256) \\ E_3'(s,t) = \text{mod}(B_3(s,t) \oplus K_6(s,t), 256) \end{cases} \tag{18}$$

where $1 \leq s, t \leq S$, and $\oplus$ represent the bitwise XOR operator. Thus, the ciphered color images E and E' are obtained after from E_1, E_2, E_3 and E'_1, E'_2, E'_3 treating them the three components of these ciphered images.

3.3 Algorithm for Decryption

During the decryption, the private as well as secret keys are needed to retrieve the encrypted images. The presented approach is an asymmetric approach, the encrypted images cannot be retrieved without the private and public keys. Steps 1 to 6 explain the decryption scheme in details.

Step 1: Receiver obtains the encrypted images and decompose it into E_1, E_2, E_3 and E'_1, E'_2, E'_3 components. Receiver calculate A_1, A_2, A_3 and B_1, B_2, B_3 using Eqs. 19 and 20,

$$\begin{cases} A_1(s,t) = \text{mod}(E_1(s,t) \oplus K_5(s,t), 256) \\ A_2(s,t) = \text{mod}(E_2(s,t) \oplus K_5(s,t), 256) \\ A_3(s,t) = \text{mod}(E_3(s,t) \oplus K_5(s,t), 256) \end{cases} \tag{19}$$

$$\begin{cases} B_1(s,t) = \text{mod}(E'_1(s,t) \oplus K_6(s,t), 256) \\ B_2(s,t) = \text{mod}(E'_2(s,t) \oplus K_6(s,t), 256) \\ B_3(s,t) = \text{mod}(E'_3(s,t) \oplus K_6(s,t), 256) \end{cases} \tag{20}$$

Step 2: Now, combined A_1, A_2, A_3 and B_1, B_2, B_3 according to Eqs. 13 and 14, we obtained two components, say L_1 and L_2 each of size $(\frac{3S}{2} \times 2S)$. After that, receiver forms a complex matrix L using L_1 and L_2, which is given as:

$$L = L_1 \cdot \exp(i \cdot L_2) \tag{21}$$

Step 3: In this step, receiver calculates the correct private key P_2 from P'_2 using inverse pixels scrambling operation in P'_2 according to key K_2. After calculating P_2, the inverse of EMD is applied using L and P_2 to obtain one complex matrix L_c, which is given by Eq. 22,

$$L_c = L + P_2 \tag{22}$$

Step 4: Now, receiver applied 2D-IFrDCT (using Eq. 7) and obtained transformed complex matrix L_t, given in Eq. 23,

$$L_t = C_{-a} L_c C^T_{-b}. \tag{23}$$

Step 5: Separates the amplitude and phase part of L_t and got L_{amp}, L_{phase}, respectively, using Eqs. 24 and 25,



Fig. 4 Results of images based on the presented cryptosystem: **a** real Lena, **b** real Sailboat, **c** real House, **d** ciphered image (amplitude), **e** ciphered image (phase), **f** deciphered Lena, **g** deciphered Sailboat, and **h** deciphered House

$$L_{\text{amp}} = \text{amp}(L_t), \tag{24}$$

$$L_{\text{phase}} = \frac{\text{phase}(L_t) \cdot 255}{\pi}. \tag{25}$$

Step 6: Finally, receiver separates the L_{amp} and L_{phase} into three components I_1, I_2, and I_3 each of size $S \times S$ and add color maps cmap1, cmap2, and cmap3, respectively, to get decrypted images.

4 Experimental Results

For the experimental results, we have taken standard color images of Lena, Sailboat, and House. For the encryption of these color images, the initial and control parameters of 6D hyperchaotic system are taken randomly as: $u_0 = 2.5432$, $v_0 = 3.6745$, $w_0 = 1.2356$, $x_0 = 1.6758$, $y_0 = y = 4.7854$, $z_0 = z = 2.3576$, $a_1 = 10.0001$, $a_2 = 28.0042$, $a_3 = 2.4248$, $r_1 = 0.0367$, and $r_2 = 0.0400$. The orders of 2D-FrDCT are $a = 2.232$, and $b = 1.534$. Figure 4 shows encryption and decryption results for method validation.

5 The Security Analysis

This section presents the security test of the defined cryptosystem.

5.1 Analysis Based on Keyspace and Key Sensitivity

In this subsection, we have given keyspace and key sensitivity analysis.

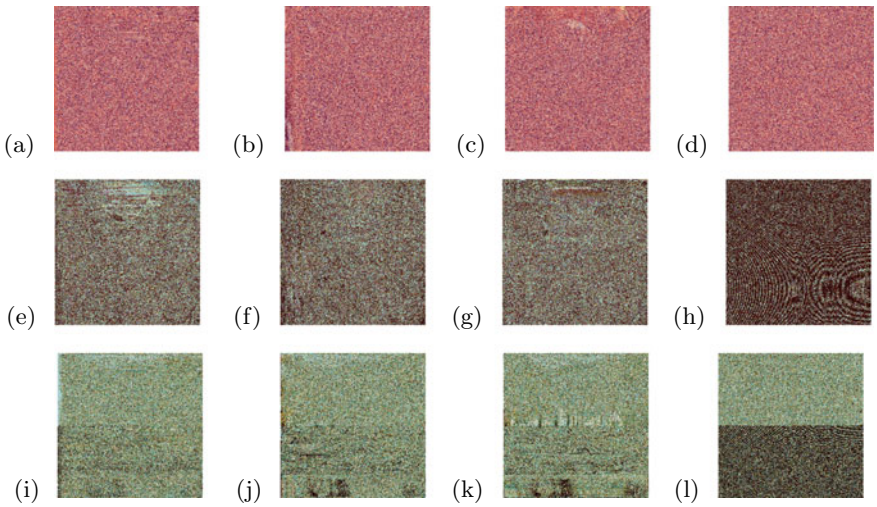


Fig. 5 Figure displayed from first column to fourth column; deciphered images of Lena, Sailboat, and House with incorrect keys: **a**, **e**, **i** private key, **b**, **f**, **j** $a = 2.231$, **c**, **g**, **k** $b = 1.533$, and **d**, **h**, **l** initial value of 6D hyperchaotic system

5.1.1 Keyspace Analysis

A sufficient large keyspace of a cryptosystem is essential for resisting the brute-force attack efficiently. The designed cryptosystem is very sensitive to secret keys of 6D hyperchaotic system: the initial values $u_0, v_0, w_0, x_0, y_0, z_0$ and control parameters a_1, a_2, a_3, r_1, r_2. Also, the fractional order parameters a, b of 2D-FrDCT are highly sensitive. If the precision is about 10^{-15}, the total keyspace of the presented cryptosystem is about $10^{195} \approx 2^{648}$, which is sufficient to resist the exhaustive search-based attacks.

5.1.2 Sensitivity of Secret Key

The private key and fractional orders are very sensitive. Also, the 6D hyperchaotic system is very sensitive to its parameters (initial and control). The experimental results of key sensitivity are depicted in Fig. 5. The results show that our presented cryptosystem is reliable against sensitivity of the secret keys.

5.2 Differential Attack Analysis

The differential attack (DA) is first analyzed by Biham and Shamir [18]. Differential attack plays a crucial role in cryptology. In this type of attack, the attacker replaces any one of the pixel in real image and then guess some relevant connection between

Table 1 Experimental values of the NPCR and UACI performed on the standard images

	Experimental values	
Standard image	NPCR (%)	UACI (%)
Lena	99.6022	33.3442
Sailboat	99.6523	33.4631
House	99.6242	33.4152

real and corresponding ciphered image. The algorithm resists DA if minor change in real image creates sufficient change in ciphered image. The DA is calculated via number of pixel change rate (NPCR) and unified averaged changed intensity (UACI) (given by Wu et al. [19]). The experimental results validate this fact. NPCR and UACI are defined in Eqs. 26 and 27,

$$\mathrm{NPCR}(E, E') = \frac{\sum_{s=1}^{S}\sum_{t=1}^{S} B(s,t)}{S \times S} \times 100\%, \tag{26}$$

$$\mathrm{UACI}(E, E') = \frac{1}{S \times S}\left[\frac{\sum_{s=1}^{S}\sum_{t=1}^{S} |E(s,t) - E'(s,t)|}{255}\right] \times 100\%, \tag{27}$$

where E and E' are ciphered images with respect to real images and the image obtained after change any one of the pixel in the real image and $B(u, v)$ is given in Eq. 28,

$$B(s,t) = \begin{cases} 0 & \text{if } E(s,t) = E'(s,t), \\ 1 & \text{if } E(s,t) \neq E'(s,t). \end{cases} \tag{28}$$

From Table 1, the proposed scheme has NPCR close to standard value 100 and UACI close to standard value 34.

5.3 *Statistical Analysis*

This subsection discusses some analyses like entropy, error, histogram, and correlation coefficient analysis.

5.3.1 Entropy Analysis

The randomness among the image pixels is measured by entropy. The entropy of the image z is defined by Eq. 29,

Table 2 Experimental values of entropy for the real image, ciphered image (amplitude), and ciphered image (phase)

Image	Entropy		
	Real image	Ciphered image (amplitude)	Ciphered image (phase)
Lena	7.2544	7.9975	7.9971
Sailboat	7.3968		
House	7.4005		

$$\text{Entropy} = -\sum_{u=1}^{S} P(z_i)\log_2 P(z_i), \tag{29}$$

where P is the probability of the symbol z_i.

Table 2 shows the experimental values of entropy for the images of Fig. 4.

5.3.2 Error Analysis

This subsection discusses the error between real and ciphered image using mathematical formulas of mean square error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index metric (SSIM) analysis.

The mathematical expression for computing MSE and PSNR for the real image (I) and ciphered image (E) is given in Eqs. 30 and 31,

$$\text{MSE}(I, E) = \frac{1}{S \times S}\sum_{s=1}^{S}\sum_{t=1}^{S}[I(s,t) - E(s,t)]^2, \tag{30}$$

$$\text{PSNR}(I, E) = 10\log_{10}\frac{(255)^2}{\text{MSE}(I, E)} \quad \text{dB}, \tag{31}$$

where s and t are the numbers of pixels in the frame.

SSIM is a formula for measuring the similarity. The lower value of SSIM shows high dissimilarity. The SSIM between I and E is calculated by Eq. 32,

$$\text{SSIM}(I, E) = \frac{(2\mu_I\mu_E + J_1)(2\sigma_{IE} + J_2)}{(\mu_I^2 + \mu_E^2 + J_1)(\sigma_I^2 + \sigma_E^2 + J_2)}, \tag{32}$$

where μ represents mean, σ represents the standard deviation, $J_1 = (k_1 H)^2$, $J_2 = (k_2 H)^2$, $k_1 = 0.01$, $k_2 = 0.03$, and $H = 2^{\text{total number of bits}} - 1$.

The calculated values of MSE, PSNR, and SSIM are given by Table 3.

Table 3 MSE, PSNR, and SSIM values for the images of Sect. 4

Image	MSE	PSNR	SSIM
Lena and encrypted (amplitude)	$8.9368e+03$	8.7152	0.0061
Sailboat and encrypted (amplitude)	$1.0211e+04$	8.1710	0.0082
House and encrypted (amplitude)	$9.3204e+03$	8.4728	0.0089
Lena and encrypted (phase)	$8.9309e+03$	8.7174	0.0066
Sailboat and encrypted (phase)	$1.0198e+04$	8.1755	0.0087
House and encrypted (phase)	$9.3009e+03$	8.4820	0.0094

Table 4 Experimental results of PSNR and SSIM

Image	PSNR	SSIM
Real Lena and decrypted Lena	38.9327	0.9547
Real Sailboat and decrypted Sailboat	38.8752	0.9376
Real House and decrypted House	38.9276	0.9423

We computed the PSNR and SSIM values between the real and deciphered images of Fig. 4. In this case, the computed values of PSNR and SSIM are displayed in Table 4.

5.3.3 2D-Histogram Analysis

This is a plot between the intensity level as X-axis and the number of pixels as Y-axis. The presented scheme is created such that it produces the ciphered images having pixels that are uniformly distributed. Figure 6 displays the histogram of the images of Sect. 4.

5.3.4 Image Pixel Correlation Analysis

Between the pixels of image, the correlation coefficient (CC) is computed by Eq. 33,

$$\mathrm{CC}(k,l) = \frac{\mathrm{cov}(k,l)}{\sqrt{D(k)}\sqrt{D(l)}}, \tag{33}$$

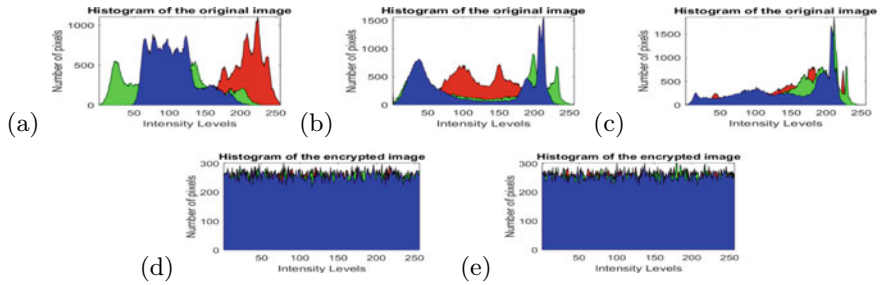


Fig. 6 Histogram results: **a** Lena, **b** Sailboat, **c** House, **d** encrypted image (amplitude), and **e** encrypted image (phase).

Table 5 CC among the pixels in H, V, and D directions for real image

Direction	H			V			D		
Image	R	G	B	R	G	B	R	G	B
Lena	0.9528	0.9360	0.9181	0.9761	0.9669	0.9484	0.9285	0.9111	0.8892
Sailboat	0.9191	0.9268	0.9308	0.9181	0.9210	0.9339	0.8859	0.8847	0.9011
House	0.8770	0.8432	0.9190	0.8831	0.8669	0.9202	0.8106	0.7673	0.8654

Table 6 CC among the pixels in H, V, and D directions for ciphered image

Direction	H			V			D		
Image	R	G	B	R	G	B	R	G	B
Encrypted image (amplitude)	0.0083	−0.0040	−0.0008	0.0018	−0.0085	−0.0014	−0.0004	0.0048	−0.0057
Encrypted image (phase)	0.0053	−0.0088	−0.0069	−0.0040	0.0006	−0.0004	0.0073	−0.0030	−0.0027

where $\text{cov}(k,l) = \sum_{u=1}^{s}\sum_{v=1}^{t}(k_{u,v} - \overline{k})(l_{u,v} - \overline{l})$, $D(k) = [\sum_{u=1}^{s}\sum_{v=1}^{t}(k_{u,v} - \overline{k})]^2$, $D(t) = [\sum_{u=1}^{s}\sum_{v=1}^{t}(l_{u,v} - \overline{l})]^2$, $k_{u,v}$ and $l_{u,v}$ are the pixels at $(u, v)^{\text{th}}$ position, $\overline{k}$ and $\overline{l}$ are mean of k and l, respectively.

Tables 5 and 6 show the calculated CC values for different original and ciphered images in three directions: H, V, and D, where H means horizontal, V means vertical, and D means diagonal.

Figure 7 shows the graph of the correlations in the adjacent pixels of real and ciphered images in the three directions. Figure 7 shows that our cryptosystem is resistant to correlation attacks.

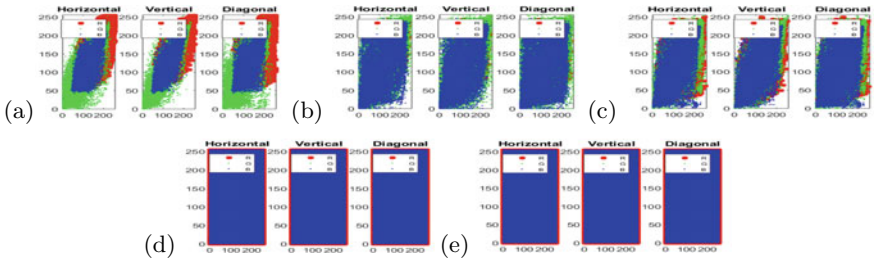


Fig. 7 Graph of correlations among the adjacent pixels of image in H, V, and D directions of: **a** real Lena, **b** real Sailboat, **c** real House, **d** amplitude of the ciphered image, and **e** phase of the ciphered image.

Table 7 Comparison of the presented cryptosystem with the related works

Performance factor	Hanis and Amutha [2]	Wua et al. [3]	Kumar et al. [15]	Proposed method
Image	Double Gray scale	Triple RGB	Gray scale	Triple RGB
Transform domain	Cellular automata	RPFrDCT	FrDCT	2D-FrDCT
PSNR of encrypted image (in dB)	Not mention	Not mention	Not mention	8.7152
PSNR of decrypted image (in dB)	38.577	Not mention	Not mention	38.9327
Sensitivity to secret keys	Yes	Yes	Yes	Yes
Key space	$2^{455} \times 2^{4096}$	$10^{70} \approx 2^{233}$	$10^{60} \approx 2^{199}$	2^{648}
SSIM of ciphered image	Not mention	Not mention	0.0008	0.0061
SSIM of deciphered image	Not mention	Not mention	0.9999	0.9547
Entropy	7.9992	Not mention	4.7453	7.9975
Differential attack	Not mention	Not mention	NPCR=99.5971 UACI=33.1616	NPCR=99.6022 UACI=33.3442
CC of encrypted image	H: −0.0009 V: 0.0007 D: 0.0007	H: 0.0200 V: 0.0048 D: −0.0026	H: 0.0944 V: 0.0057 D: 0.0067	H: 0.0083 V: −0.0008 D:−0.0004

6 Related Works Comparison

To check the performance of the presented triple color image security technique, a comparison process against related works has been performed. The presented cryptosystem is compared with the formerly developed cryptosystem [2, 3, 15]. The parameters considered for comparison are image type, transform domain, PSNR of encrypted as well as decrypted image, sensitivity to secret keys, keyspace, SSIM of encrypted as well as decrypted image, entropy, differential attack, and correlation coefficient analysis, as shown in Table 7.

7 Conclusion

This research paper presents an asymmetric cryptosystem for the security of digital triple color images using a 6D chaotic map and EMD in the 2D-FrDCT domain. In the presented cryptosystem, the parameters of the 6D chaotic map, the fractional order of 2D-FrDCT, and private key are the extra keys, so the presented cryptosystem has sufficient large keyspace and it is very sensitive concerning that keys. The results of experiment and security provided confirm that the presented technique is very secure.

References

1. Gong L, Deng C, Pan S, Zhou N (2018) Image compression-encryption algorithms by combining hyper-chaotic system with discrete fractional random transform. Opt Laser Technol 103:48–58
2. Hanis S, Amutha R (2018) Double image compression and encryption scheme using logistic mapped convolution and cellular automata. Multimed Tools Appl 77(6):6897–6912
3. Wua J, Guoa F, Liangb Y, Zhou N (2014) Triple color images encryption algorithm based on scrambling and the reality-preserving fractional discrete cosine transform. Optik 125(16):4474–4479
4. Joshi AB, Kumar D, Gaffar A, Mishra DC (2020) Triple color image encryption based on 2D multiple parameter fractional discrete Fourier transform and 3D Arnold transform. Opt Lasers Eng 133:106139. https://doi.org/10.1016/j.optlaseng.2020.106139
5. Kumar D, Joshi AB, Mishra VN (2020) Optical and digital double color-image encryption algorithm using 3D chaotic map and 2D-multiple parameter fractional discrete cosine transform. Results in Optics 1:100031
6. Joshi AB, Kumar D (2019) A new method of multi color image encryption. In: IEEE conference on information and communication technology. Allahabad, India, pp 1–5. https://doi.org/10.1109/CICT48419.2019.9066198
7. Joshi AB, Kumar D, Mishra DC, Guleria V (2020) Colour-image encryption based on 2D discrete wavelet transform and 3D logistic chaotic map. J Mod Opt 67(10):933–949
8. Chen H, Liu Z, Zhu L, Tanougast C, Blondel W (2019) Asymmetric color cryptosystem using chaotic Ushiki map and equal modulus decomposition in fractional Fourier transform domains. Opt Lasers Eng 112:7–15

9. Chen H, Zhu L, Liu Z, Tanougast C, Liu F, Blondel W (2020) Optical single-channel color image asymmetric cryptosystem based on hyperchaotic system and random modulus decomposition in Gyrator domains. Opt Lasers Eng 124:105809
10. Yadav AK, Singh P, Saini I, Singh K (2019) Asymmetric encryption algorithm for colour images based on fractional Hartley transform. J Mod Opt 66(6):629–642
11. Joshi AB, Kumar D, Mishra DC (2020) Security of digital images based on 3D Arnold cat map and elliptic curve. Int J Image Graph 20(4):2150006
12. Huang ZJ, Cheng S, Gong LH, Zhou NR (2020) Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform. Opt Lasers Eng 124:105821–105831
13. Yao L, Yuan C, Qiang J, Feng S, Nie S (2017) Asymmetric color image encryption based on singular value decomposition. Opt Lasers Eng 89:80–87
14. Ahmed N, Natarajan T, Rao KR (1974) Discrete cosine transform. IEEE Trans Comput 23(1):90–93. https://doi.org/10.1109/T-C.1974.223784
15. Kumar S, Panna B, Jha RK (2019) Medical image encryption using fractional discrete cosine transform with chaotic function. Med Biol Eng Comput 57(11):2517–2533
16. Grassi G, Severance FL, Miller DA (2009) Multi-wing hyperchaotic attractors from coupled Lorenz systems. Chaos Solitons Fractals 41(1):284–291
17. Cai J, Xueju S, Lin C (2016) Security-enhanced asymmetric optical cryptosystem based on coherent superposition and equal modulus decomposition. Opt Commun 359:26–30
18. Biham E, Shamir A (1991) Differential cryptanalysis of DES-like cryptosystems. J Cryptol 4(1):3–72
19. Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. J Selected Areas Telecommun 1(2):31–38

On the Evaluation and Implementation of LSTM Model for Speech Emotion Recognition Using MFCC



Sheetal U. Bhandari, Harshawardhan S. Kumbhar, Varsha K. Harpale, and Triveni D. Dhamale

Abstract Speech emotion recognition is an emerging research field and is expected to benefit many application domains by providing effective human–computer interface. Researchers are extensively working toward decoding of human emotions through speech signal in order to achieve effective interface and smart response by computers. The perfection of speech emotion recognition greatly depends upon the types of feature used and also on the classifier employed for recognition. The contribution of this paper is to evaluate twelve different long short-term memory (LSTM) network models as classifier based on Mel frequency cepstrum coefficient (MFCC) feature. The paper presents performance evaluation in terms of important parameters such as: precision, recall, F-measure and accuracy for four emotions like happy, neutral, sad and angry using the emotional speech databases, namely Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). The measurement accuracy obtained is 89% which is 9.5% more than reported in recent literature. The suitable LSTM model is further successfully implemented on Raspberry Pi board creating stand-alone speech emotion recognition system.

Keywords Human–computer interaction · SER · MFCC · LSTM · Speech emotion recognition

1 Introduction

The way human interacts with computers has traveled a long way and the journey continues. As a result, new designs and technologies keep appearing in market frequently. The biggest milestone in the field of human–computer interface (HCI)

S. U. Bhandari · H. S. Kumbhar · V. K. Harpale
Department of Electronics and Telecommunication, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India

T. D. Dhamale (✉)
Department of Electronics and Telecommunication, Pimpri Chinchwad College of Engineering and Research, SPPU, Ravet, Pune, India
e-mail: triveni.dhamale@pccoer.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_33

is the ability to control computing machinery through human voice. However, early technologists have largely ignored emotions in the interface, mainly because emotion is difficult to understand and hard to measure. As emotion is fundamental to human thoughts, feelings, decision-making and communication, human interface with computing machinery without considering emotions has started making frustrating experience for users.

In order to increase effectiveness of HCI, the domain of human emotion recognition gained prominence. In the literature, researchers have explored various techniques to recognize the emotional states such as facial expressions, gestures, speech, physiological signals like EEG, etc. [1, 2]. For facial expressions and gestures, the person is expected to be facing the camera [3, 4]. In case of EEG signal, it is difficult to keep wearing electrodes [5]. Due to several inherent advantages, speech emotion recognition (SER) is looked upon as promising technique among others by majority of researchers [6]. There are many applications which are expected to get benefited by SER like robot interface, Web movies, audio surveillance, computer tutorials, in-car board systems, clinical studies, call centers, computer games, etc.

The task of SER is very challenging and demands for selection of appropriate tools and techniques. Speech signal characteristics like energy, pitch, fundamental frequency for same word vary person to person, geographical region and language. Hence, extracting effective feature and designing reliable classifier is very important. In the process of deciding suitable feature and classifier, lot of experimentation is required which is carried out on emotional speech database; hence, selection of good database is very important key in the decision of right features and classifier for SER [7].

Many speech features can be used as input to the classifier. However, appropriate feature selection is necessary to provide better learning performance, low computational complexity, better generalized model, and low memory. In recent work, many common features are extracted, such as energy, pitch, formant and some spectrum features such as linear prediction coefficient (LPC), Mel frequency cepstrum coefficient (MFCC) [8, 9] and modulation spectral features. As per the short, survey as presented in Table 1, energy and pitch show the higher effectiveness. MFCC contains the energy data for each frequency filter bank and hence gives better result for classification. In this work, MFCC is selected to extract the emotional features. Akcay et al. [10] highlight all the methods, databases, features and classifiers presented in last two decades. There are challenges to implement highly accurate SER such as limited dataset, multi-language speech signal processing and in real-time systems multi-speech signals. There is need of reliable, focused and accurate SER to detect emotion from multi-language and noisy speech signals. The attention-based improved LSTM [11] reduces the computational complexity and preserves the timing relations with original speech signal as compared to traditional statistical features. The work is experimented with CASIA, eNTERFACE and GEMEP emotion corpora and observed to be significant as compared to other methods. The attention-based LSTM [12], tested for interactive emotional dyadic motion capture (IEMOCAP) database, could perform with only 73% of weighted accuracy. Thus, the accuracy of attention-based LSTM depends on reliability of databases.

Table 1 Literature survey

Refs.	ML model	Features or data used	Maximum accuracy (%)
[13]	DNN	Audio Chunk	96.97
[14]	RNN	Raw spectral, LLDs	63.5
[15]	DCNN	Mel spectrum	87.31
[16]	SVM	MFCC, LPCC, PLP, DAE	86.41
[17]	ANN, SVM	MFCC, HFCC, GFCC	78.6
[18]	ConNets	Frame based	64.78
[19]	CNN, time distributed CNN and LSTM	STFT	88.01
[20]	GMM-HMM, SGMM-HMM, DNN-HMM	MFCC and energy	87.62
[21]	CNN	MFCC	76.1
[22]	SVM, ladder network, DAE, VAE	LLD, energy, MFCC, F0, HNR	59.7
[23]	CNN, DNN-ELM	3D log mel spectrum	84.99

Multiple learning algorithms are available for classification. In recent works, use of ANN, SVM, CNN, RNN, HMM, DCNN to classify emotions is reported as given in Table 1. The proposed system uses long short-term memory (LSTM) networks, as LSTM has memory to store the previous value and MFCCs are calculated on previous and current frame of speech.

Major contributions of this work are:

- Twelve different models of LSTM are developed, and its performance in terms of precision, recall, F1 score and accuracy is evaluated for four basic emotions, viz. neutral, angry, happy and sad.
- Hardware implementation of suitable model to develop SER system on Raspberry Pi board with necessary input/output interface.

This paper is organized as follows. In Sect. 2, details of used methodology are presented. Results of experimentation are given and discussed in Sect. 3. Hardware implementation of SER system using Raspberry Pi board is given in Sect. 4. Concluding remarks are given in Sect. 5.

2 Materials and Methods

This section presents details about used feature extraction and classification method: MFCC and LSTM, respectively. Details about emotional speech database used for experimentation are also given initially to justify various selection parameters of MFCC and LSTM.

A. *Dataset*

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is dataset used in proposed system [24]. The dataset consists of voice of 24 actors comprising 12 females and 12 males. Each one has spoken 2 sentences in 8 different emotions as calm, happy, sad, angry, fearful, surprise and disgust expressions. A total of 7356 clips are available in dataset out of which 4320 are speech recordings and remaining are song recordings. All clips are in wave (.wav) file format. The sampling frequency of recorded file is 48 kHz. The signed 16-bit PCM format is used; hence, there are 16 bits per samples. It is a differentiated dataset having specific samples for specific emotion. In proposed work, data of only four emotions out of 8 is used. Number of clips used is: 192 angry, 192 happy, 192 sad and 96 neutral.

B. *Extraction of Mel Frequency Cepstrum Coefficients*

MFCC represents nonlinear, short-term power spectrum of speech signal. It is best matching for speech signal recognition as sensitivity of human ear is also nonlinear. Cochlea exhibits nonlinear sensitivity and is more sensitive to lower frequencies as compared to higher frequencies [25, 26]. Figure 1 shows the process of MFCC feature calculations which is described in detail afterward.

(1) *Pre-emphasis*: The main goal of pre-emphasis is to amplify the high-frequency component of signal. This balances the frequency spectrum and improves SNR. In this step, speech signal $s(n)$ is sent through high-pass filter:

$$s_2(n) = s(n) - a * s(n-1) \tag{1}$$

where

s(n): input signal
s_2(n): output signal

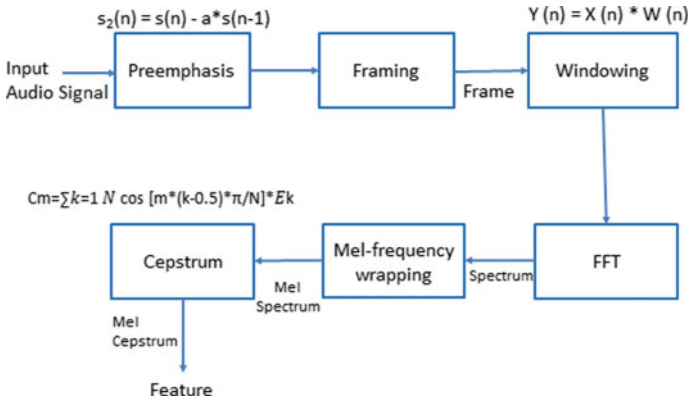


Fig. 1 MFCC calculation process

a: filter coefficient usually between 0.9 and 1.0. The value selected for implementation is 1.

2. *Framing*: during framing stage, the input speech signal is segmented into frames of duration ranging from 20 to 30 ms with optional overlap of 1/3 to 1/2 of the frame size. The frame size is a function of sample rate and frame duration as Eq. 2:

$$\text{Frame Size} = \text{Frame Duration} * \text{Sample Rate} \tag{2}$$

The sampling frequency of RAVDESS dataset is 48 kHz, and frame duration selected is 20 ms with no overlap; hence as per Eq. 2, the frame size is 960 samples.

(3) *Windowing*: Each frame has to be multiplied with a Hamming window in order to keep the continuity of the first and the last points in the frame. If the signal in a frame is denoted by $s(n)$, $n = 0, N - 1$, then the signal after Hamming windowing is $s(n) \times w(n)$, where $w(n)$ is the Hamming window defined by Eq. 3:

$$w(n) = 0.54 - 0.46 * \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq (N-1) \tag{3}$$

(4) *Fast Fourier transform*: Different tones in speech signal correspond to different energy distributions over frequencies. Therefore, FFT is performed to obtain the magnitude frequency response of each frame. 512-point FFT is used in this system.

(5) *Mel frequency wrapping*: It has triangular bandpass filtering. The Mel frequency is kept proportional to the logarithm of the linear frequency, which resembles to human ear's perception. The magnitude frequency response is multiplied by a set of 40 triangular bandpass filters to get the log energy of each triangular bandpass filter. The positions of these filters are equally spaced along the Mel frequency, which is related to the common linear frequency f by Eq. 4:

$$\text{Mel}(f) = 1127 * \ln\left(1 + \frac{f}{700}\right) \tag{4}$$

C. LSTM Architecture

LSTM is designed to handle long-term dependencies within data. LSTM network stores the information for long period of time. Figure 2 shows the structure of LSTM network. Network has three inputs $Ct - 1$, $ht - 1$ and xt. Network gives two outputs Ct and ht. LSTM network is implemented using Python language. For implementation, Keras library is used. Keras is an open-source neural network library written in Python. Following function represents the standard LSTM layer function. The inputs

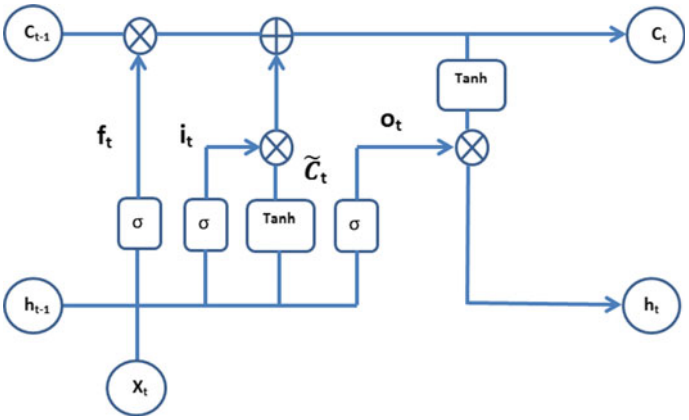


Fig. 2 LSTM architecture

for these functions are cell input, cell state and data input. This function returns hidden layer output and cell memory output.

def standard lstm (cell inputs, cell states, kernel, z = data input)

```
{
        h_tm1 = cell states [0] # previous memory state
        c _tm1 = cell states [1] # previous carry state
        z0, z1, z2, z3 = array ops.split (z, 4, axis=1)
        i = recurrent activation (z0)
        f = recurrent activation (z1)
        c = f * c tm1 + i * activation (z2)
        o = recurrent activation (z3)
        h = o * activation(c)
        return h, [h, c]
}
```

where i = input layer value, f = forgot layer value, c = cell memory value, o = output of cell, h = hidden layer output, z = array of input data for a LSTM cell.

In above pseudo-code, recurrent activation and activation are decided by external calling function; in proposed system, recurrent activation uses sigmoid function whereas activation function uses tanh function. This function is repeated till all data samples covered. In proposed system, two variants of network are implemented with two hidden layers and with four hidden layers. In first case, two hidden layers after LSTM layer are used with activation function ReLU and tanh, respectively. In second case, four hidden layers after a LSTM layer are used. These four layers have activation function ReLU, tanh, ReLU and tanh, respectively.

3 Experimentation and Results

A. *Feature Extraction*

Using the method described in Sect. 2(B), the MFCC values for the different emotions are calculated. The graph of coefficient values with respect to 39 coefficients for neutral emotion is shown in Fig. 3. MFCC is calculated for the 4 samples of RAVDESS dataset and plotted as sample representation.

The observed effect of emotions on MFCCs is as given in Table 2:

Values of coefficients vary with emotions. Angry has highest value, while neutral has lowest proving effectiveness of MFCC in SER. Early coefficients show remarkable variations and carry maximum information. Maximum variation is found till 12 coefficients. Coefficients from 12 to 28 have less variations, while variation after 28th coefficient is very less or negligible.

B. *LSTM Model and Performance Parameters*

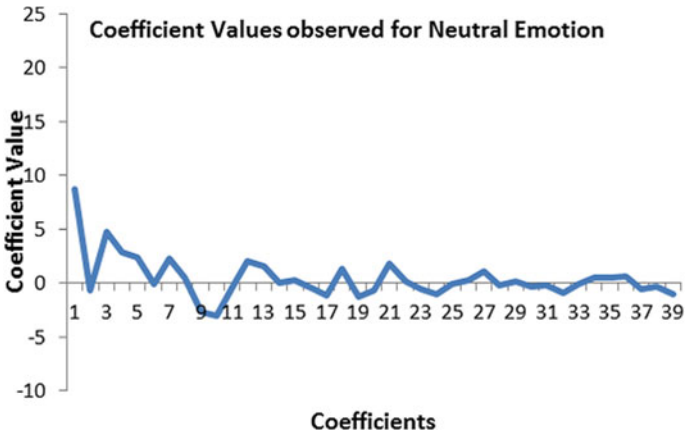


Fig. 3 Sample of MFCC—neural

Table 2 LSTM model coefficients

Model number	1	2	3	4	5	6	7	8	9	10	11	12
MFCC	39	39	39	39	28	28	28	28	12	12	12	12
Hidden layer	4	2	2	2	4	2	2	2	4	2	2	2
Output size of LSTM layer	128	64	32	16	128	64	32	16	128	64	32	16
Output size of 1st h layer	32	32	16	8	32	32	16	8	32	32	16	8
Output size of 2nd h layer	16	16	8	8	16	16	8	8	16	16	8	8
Output size of 3rd h layer	8	–	–	–	8	–	–	–	8	–	–	–
Output size of 4th h layer	8	–	–	–	8	–	–	–	8	–	–	–

During experimentation phase, twelve different models of LSTM are evaluated to identify best suitable model for SER. The combinations of no. of hidden layers, size of hidden layers and no. of MFCCs used for twelve models are given in Table 2. The models are trained, validated and tested with the RAVDESS dataset.

A total of 675 audio files from dataset are used. Each audio file is divided into 142 time slots. And 39 MFCCs are calculated for each time slot. So, total number of data points are 142 * 39 * 675 = 3,738,150. But in LSTM data given to network is in the form of sets, so total number of sets generated are 142 * 675 = 95,850. While training network, 80% of data is used for training and 20% of data used for testing. 20% of 675 means 135 audio files are used for testing.

C. *Model Evaluation Results*

As shown in Table 3, the highest precision value of 1 is observed in Model 8 for neutral emotion, Model 11 gives the highest recall for sad emotion, while highest F1 score is observed for angry emotion in Model 11 and highest accuracy is given by Model 10 and Model 11 for happy and angry emotions, respectively. But for other emotions, other models give higher performance than these models. Hence, considering average performance parameters of all models as shown in Fig. 4, Model 10 and Model 11 are found to be providing better average precision, recall, F1 score and accuracy than other models. The obtained accuracy is 9.5% greater than the recent literature [27] as depicted in Table 4. Considering performance with respect to models as considered in Table 2, it can be seen that 12 MFCCs are sufficient and increasing no. of hidden layers to four does not yield more performance. Even the Models 10 and 11 give

Table 3 Performance of implemented LSTM model

Model number		1	2	3	4	5	6	7	8	9	10	11	12
Precision	Neutral	0.89	0.76	0.7	0.83	0.7	0.87	0.79	1	0.83	0.89	0.78	0
	Angry	0.88	0.84	0.69	0.75	0.86	0.82	0.76	0.72	0.8	0.84	0.92	0.84
	Happy	0.62	0.73	0.81	0.58	0.81	0.65	0.56	0.68	0.69	0.59	0.85	0.8
	Sad	0.55	0.56	0.47	0.52	0.58	0.6	0.56	0.4	0.61	0.78	0.6	0.33
Recall	Neutral	0.61	0.68	0.5	0.36	0.57	0.71	0.54	0.14	0.36	0.54	0.5	0
	Angry	0.67	0.76	0.79	0.71	0.76	0.76	0.69	0.74	0.76	0.79	0.83	0.64
	Happy	0.71	0.69	0.74	0.6	0.71	0.69	0.71	0.54	0.83	0.89	0.8	0.57
	Sad	0.8	0.73	0.5	0.57	0.83	0.7	0.6	0.6	0.77	0.87	0.9	0.77
F1 score	Neutral	0.72	0.72	0.58	0.5	0.63	0.78	0.64	0.25	0.5	0.67	0.61	0
	Angry	0.76	0.8	0.73	0.73	0.81	0.79	0.72	0.73	0.78	0.84	0.88	0.73
	Happy	0.67	0.71	0.78	0.59	0.76	0.67	0.63	0.6	0.75	0.86	0.82	0.67
	Sad	0.65	0.64	0.48	0.54	0.68	0.65	0.58	0.48	0.68	0.7	0.72	0.46
Accuracy	Neutral	0.9	0.89	0.84	0.84	0.85	0.93	0.87	0.77	0.86	0.89	0.86	0.73
	Angry	0.87	0.88	0.82	0.84	0.89	0.88	0.84	0.83	0.87	0.9	0.93	0.85
	Happy	0.81	0.82	0.89	0.79	0.88	0.82	0.78	0.81	0.86	0.93	0.91	0.85
	Sad	0.81	0.79	0.76	0.79	0.83	0.83	0.81	0.71	0.84	0.84	0.84	0.61

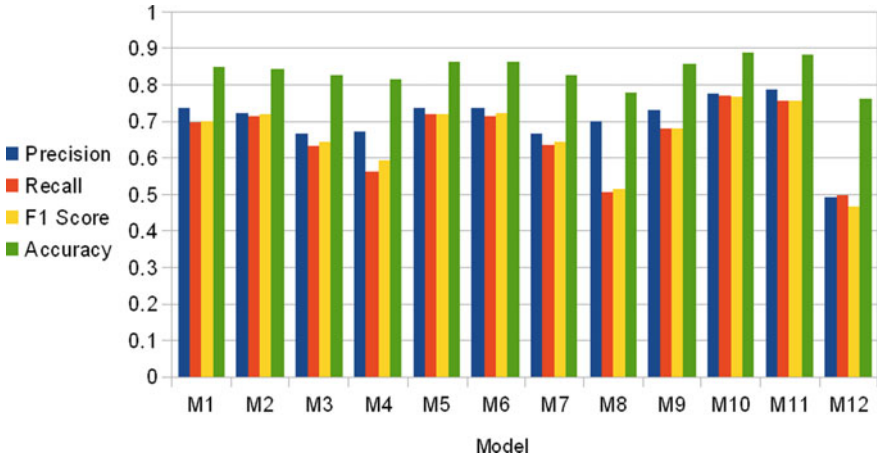


Fig. 4 Average performance parameters of 12 models

Table 4 Performance comparison

Parameters	Mustaqeem et al. [25]	Proposed method
Dataset	RAVDESS	RAVDESS
Method	Assisted deep stride convolutional neural network (DSCNN)	LSTM + MFCC
Accuracy (%)	79.5	89

comparable performance, even if the size of model 10 is bigger than Model 11. Hence, Model 11 is chosen for hardware implementation.

4 System Implementation

In order to create a stand-alone system able to recognize any of the four emotions: happy, angry, sad and neutral, from the speech of speaker in real time, the LSTM Model 11 from Sect. 3 is chosen for hardware implementation. This section provides implementation details.

A. *Block Diagram*

The high-level structure of hardware implementation is given in Fig. 5. The system contains Raspberry Pi 3B+ board as core processing element along with microphone and touch screen display. Microphone is used to get the real-time audio signal. The speech processing and deep learning algorithms are implemented in the Raspberry Pi module using Python language. To display the result, 3.5-inch resistive touch screen TFT display is used.

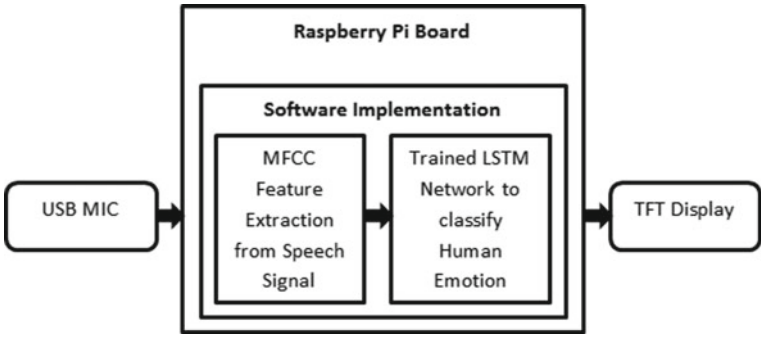


Fig. 5 High-level structure of the hardware implementation

The real-time speech signal is taken from the microphone through USB interface. The data is in the 16-bit PCM format. During speech processing MFCC, coefficients are calculated from the data. To record and calculate MFCC, AudioPy and Librosa packages are used from the Python library. The calculated MFCC value is then given to the trained LSTM Model 11.

For training and evaluation of LSTM model, Keras package from Python library is used. The emotion predicted by LSTM model is displayed on the TFT display in the form of emojis. Touch feature of TFT is used to start and stop the program. The flow diagram of system during training and evaluation mode is given in Fig. 6. Figure 6a shows the working flow of the system at the time of training the LSTM model. In this process, each file from dataset is taken and MFCCs are calculated. These calculated values are stored in an array. These values are then given to LSTM model for training purpose. The model is trained for 100 epochs. The trained model is stored with their weights. This model is used at the time of real-time processing for evaluation. Figure 6b shows the working flow of system at the time evaluation mode. In this process, a stored LSTM model is recalled and features are feed to it to predict the classes. The chunk data of audio is collected from the USB MIC; then, this is divided into frames of 20 ms to calculate MFCC, value of MFCC is calculated for each frame and is given to the LSTM model; then, the emojis are displayed according to the predicated emotion class by LSTM model.

B. *Hardware Setup and Results*

Figure 7 shows the hardware setup of SER system around Raspberry Pi. A USB microphone and display are interfaced with Raspberry Pi 3B+ Board. A real-time audio was captured through microphone and given it to the Raspberry Pi. The result obtained for different audio signals in the form of emojis is shown in Fig. 8.

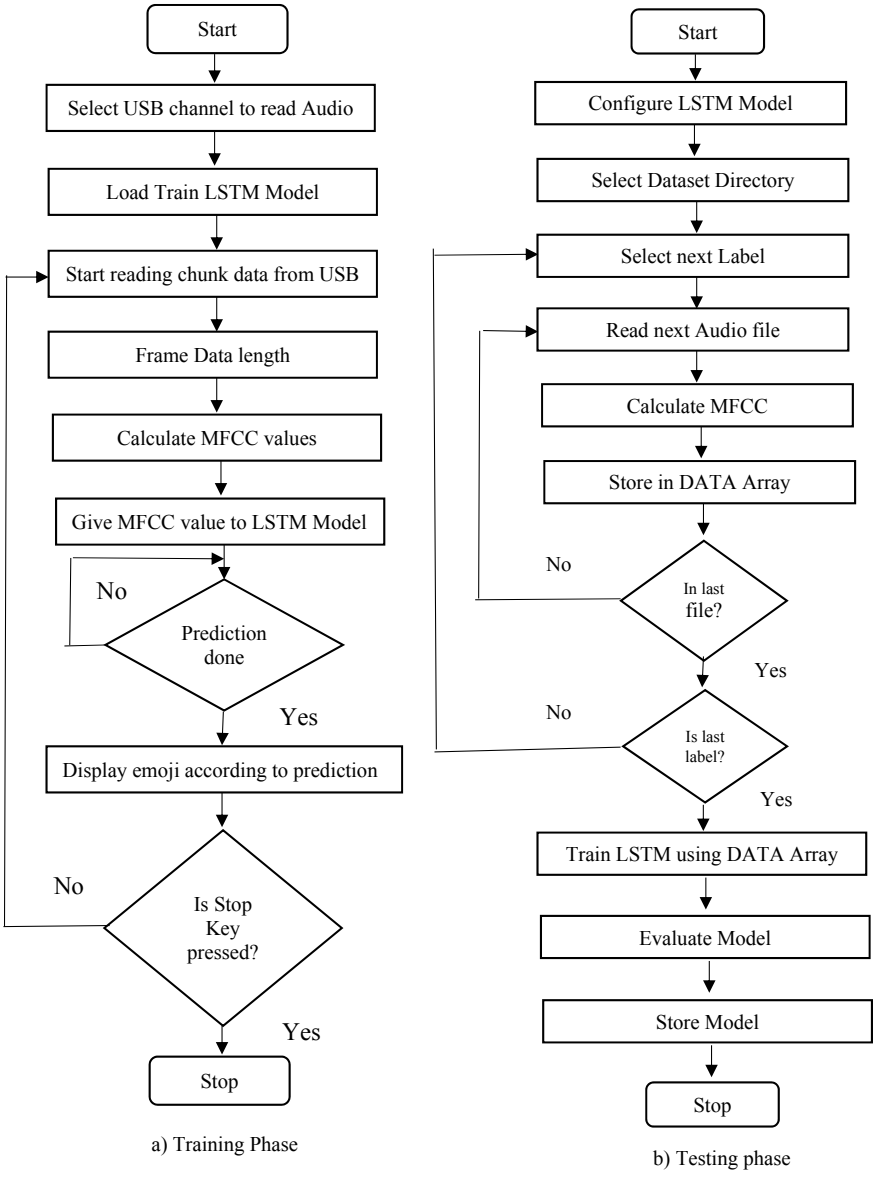


Fig. 6 Flowcharts of training and testing



Fig. 7 Experimental setup of SER system

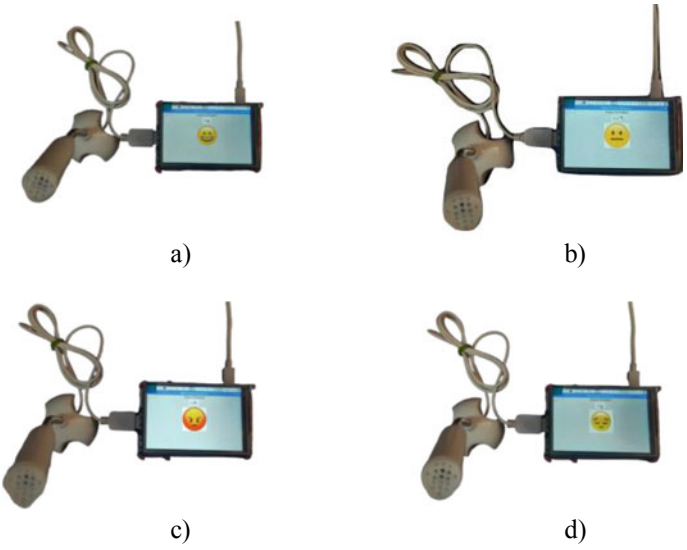


Fig. 8 Detected emotions displayed using emojis: **a** happy, **b** neutral, **c** angry, **d** sad

5 Conclusion

A stand-alone SER system around Raspberry Pi board is developed to detect happy, angry, neutral and sad emotions. Speech input is taken from microphone, and predicted emotions are displayed on TFT display using emojis. The system uses LSTM algorithm as classifier and MFCC to extract emotional features. In order to select optimum model for system implementation, performance of 12 different

LSTM models is obtained through experimentation. Based on the precision, recall, F1 score and accuracy, Model 11 is chosen for hardware implementation. The average precision, recall, F1 score and accuracy of Model 11 are 0.79, 0.76, 0.76 and 0.89, respectively. The obtained accuracy is 9.5% greater than reported in recent literature. Based on the experimentation, it can be concluded that first 12 coefficients of MFCC carry maximum information, small impact of emotions is observed on coefficient ranging between 13 and 28, while 29 to 39 coefficients have negligible impact; hence, 13–39 coefficients can be neglected without affecting the accuracy of system while keeping complexity minimum. It is observed that up to 2 hidden layers in LSTM model are sufficient. Further increase in hidden layers (4 hidden layers) increases the accuracy merely by 1–2% only.

References

1. McDonald M (2000) Active research topics in human machine interfaces. In: Technical report: U.S. Department of Energy Office of Scientific and Technical Information, 12. Available: https://www.osti.gov/biblio/773841
2. Shu L, Xie J, Yang M, Li Z, Li Z, Liao D, Xu X, Yang X (2018) A review of emotion recognition using physiological signals. Sensors (Basel, Switzerland), MDPI 18(7). Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6069143/
3. Praveen Kulkarni RTM (2020) Analysis on techniques used to recognize and identifying the human emotions. Int J Electr Comput Eng (IJECE) 10(3):3307–3314
4. Fatima Zahra Salmam AM, Kissi M (2018) Emotion recognition from facial expression based on fiducial pointsdetection and using neural network. Int J Electr Comput Eng (IJ ECE) 8(1):52–59
5. Fabian Parsia George IMS, Uddini J (2019) Recognition of emotional states using EEG signals based on time-frequency analysis and SVM classifier. Int J Electr Comput Eng (IJECE) 2(9):1012–1020
6. El Ayadi MS, Kamel M, Fakhri K (2011) Survey on speech emotion recognition: features, classification schemes, and databases. Pattern Recogn 44(3):572–587. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320310004619
7. Qadri SA, Gunawan TS, Alghifari MF, Mansor H, Kartiwi M, Janin Z (2019) A critical insight into multi-languages speech emotion databases. Bull Electr Eng Inform 8(4):1312–1323
8. Bhangale KB, Mohanaprasad K (2021) A review on speech processing using machine learning paradigm. Int J Speech Technol:1–22
9. Norhaslinda Kamaruddin AWAR, Halim KIM, Noh MHIM (2018) Driver behaviour state recognition based on speech. TELKOMNIKA 16(2):852–861
10. Akcay MB, Ouz K (2020) Speech emotion recognition: emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. Speech Commun 116:56–76. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639319302262
11. Xie Y, Liang R, Liang Z, Huang C, Zou C, Schuller B (2019) Speech emotion classification using attention-based LSTM. IEEE/ACM Trans Audio, Speech, Lang Process 27(11):1675–1685
12. Yu Y, Kim Y-J (2020) Attention-LSTM-attention model for speech emotion recognition and analysis of IEMOCAP database. IEEE/ACM Trans Audio, Speech, Lang Process 9(5):713
13. Harr P, Burget R, Dutta M (2017) Speech emotion recognition with deep learning. In: 2017 4th International conference on signal processing and integrated networks (SPIN), 02, pp 137–140
14. Mirsamadi S, Barsoum E, Zhang C (2017) Automatic speech emotion recognition using recurrent neural networks with local attention. In: 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP), pp 2227–2231

15. Zhang S, Zhang S, Huang T, Gao W (2018) Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching. IEEE Trans Multimedia 20(6):1576–1590
16. Fei W, Ye X, Sun Z, Huang Y, Zhang X, Shang S (2016) Research on speech emotion recognition based on deep auto-encoder. In: 2016 IEEE International conference on cyber technology in automation, control, and intelligent systems (CYBER), pp 308–312
17. Sugan N, Sai Srinivas NS, Kar N, Kumar LS, Nath MK, Kanhe A (2018) Performance comparison of different cepstral features for speech emotion recognition. In: 2018 International CET conference on control, communication, and computing (IC4), pp 266–271
18. Fayek HM, Lech M, Cavedon L (2017) Evaluating deep learning architectures for speech emotion recognition. Neural Netw 92:60–68
19. Lim W, Jang D, Lee T (2016) Speech emotion recognition using convolutional and recurrent neural networks. In: 2016 Asia-Pacific signal and information processing association annual summit and conference (APSIPA), pp 1–4
20. Mao S, Tao D, Zhang G, Ching PC,Lee T (2019) Revisiting hidden Markov models for speech emotion recognition. In: ICASSP 2019—2019 IEEE International conference on acoustics, speech and signal processing (ICASSP), pp 6715–6719
21. Tripathi S, Kumar A, Ramesh A, Singh C, Yenigalla P (2019) Deep learning-based emotion recognition system using speech features and transcriptions
22. Tao J-H, Huang J, Li Y, Lian Z, Niu M-Y (2019) Semi-supervised ladder networks for speech emotion recognition. Int J Autom Comput 16(4):437–448. https://doi.org/10.1007/s11633-019-1175-x
23. Meng H, Yan T, Yuan F, Wei H (2019) Speech emotion recognition from 3d log-mel spectrograms with deep learning network. IEEE Access 7:125868–125881
24. Livingstone SR, Russo FA (2018) The Ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. PLOS ONE 13(5):e0196391. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0196391
25. Sonawane A, Inamdar MU, Bhangale KB (2017)Sound based human emotion recognition using MFCC and multiple SVM. In: 2017 International conference on information, communication, instrumentation and control (ICICIC). IEEE, pp 1–4
26. Bhangale KB, Titare P, Pawar R, Bhavsar S (2018) Synthetic speech spoofing detection using MFCC and radial basis function SVM. IOSR J Eng (IOSRJEN) 8(6):55–62
27. Mustaqeem M, Kwon S (2020) A CNN-assisted enhanced audio signal processing for speech emotion recognition. Sensors (Basel, Switzerland) 20(1)

Analyzing the Effects of Text Representations on the Performance of Document Clustering in Public Health Tweets



Aakansha Gupta and Rahul Katarya

Abstract When working with large amounts of textual data from social media platforms, it is essential to identify topics quickly and accurately. Different methods have been developed to collect information from online platforms, but not all the approaches have provided a good result. This paper investigates various text representation methods based on the vector space model (VSM) and word embedding models for embedding short texts into low-dimensional representations for semantic similarities. Next, we evaluate topic modeling and three document clustering algorithms, trained on various text representation models, to discover topics from the textual data collected from Twitter. We compare six feature representations obtained from term frequency–inverse document frequency (tf-idf) matrices and word embedding models, as well as a latent Dirichlet allocation topic model. Obtained results suggest that clustering techniques applied to VSM-weighted word embedding feature representations methods outperform traditional word embedding models. Further, we interpret the usefulness of the clustering results by analyzing tf-idf-based top terms. Finally, we create the word clouds for a better understanding of clusters.

Keywords Text representation · Vector space model · Word embeddings · Document clustering · Topic modeling · Twitter topics discovery

1 Introduction

The massive amount of data generated by social media sites contributes to developing methods of topic extraction and event detection [1]. However, the content created by these platforms is poorly grammatical, unorganized, comprises incorrect sentences, misspellings, out of vocabulary words, redundant details, and other metadata that

A. Gupta (✉) · R. Katarya
Big Data Analytics and Web Intelligence Laboratory, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, India

R. Katarya
e-mail: rahulkatarya@dtu.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_34

can influence its possible meaning. Traditional natural language processing technologies face various obstacles when dealing with Twitter data. For example, parsers and other traditional NLP tools are frequently insufficient for processing abbreviations, Internet slang, etc. Tweets may make explicit or implicit references to specific locations, events, and other specified entities. As a result, pre-defined entity lists and complicated named entity recognition methods are ineffective. Although many techniques, including text mining, topic modeling, and neural embedding approaches, have been used to extract the salient knowledge from the texts, our work is focused on evaluating the performance of document clustering algorithms such as k-means, NMF, and HAC embedded with different text representation models. We also evaluated one topic modeling approach—LDA [2], for the topic discovery task. To enhance clustering algorithms' performance, we integrated them with various text representation models, some of which are generated by combining the word embedding models with VSM.

Document clustering methods aggregate documents based on a certain feature matrix to ensure that the documents in a cluster are identical to those in other clusters. Clustering methods in such contexts build hierarchies or partitions. Hierarchical methods used in this paper are NMF, HAC, and partitioning methods which include k-means. Document clustering methods use either vector space models or Language models for representation. The VSM [3] is the most commonly used model to represent the document. Each term in the document represents a feature, and there are many ways to determine the weight of the terms. A common approach for weight calculation is to use the tf-idf method.

In machine learning, topic modeling algorithms are a statistical method for discovering hidden semantic structures and uncover meaning from a large collection of documents [4]. Like document clustering, topic models can group the documents given a set of topics for each document. Among the topic modeling techniques available, LDA is the most commonly used method for topic discovery [1].

Our work provides an analysis of the performance of various clustering methods trained with different text representation methods. We also include one topic model for comparison. Furthermore, these methods are evaluated using different performance metrics such as normalized mutual information (NMI), adjusted mutual information (AMI), and adjusted Rand index (ARI). Lastly, we provide a visual summary of the resultant clusters by displaying them as word clouds.

The remaining paper is structured as follows. Section 2 is dedicated to discuss the literature in this research field. Section 3 discusses the methodology followed for this paper, including an overview of data collection, preprocessing and preparation, employed techniques for text representation and clustering, topic modeling approach, along with performance measures. Section 4 presents the comparison result among topic modeling and clustering algorithms trained with different text representation approaches. Section 5 represents word clouds formation to provide better visualization of clusters. Finally, Sect. 6 ends with a conclusion.

2 Related Research

Since there has been a significant increase in social media users exchanging information, researchers have a lot of traction analyzing social media behaviors for public health purposes. For example, Kwak et al. [5] investigated the potential of Twitter as a platform for information sharing.

There are already many studies that deal with feature representation while discovering topics from social media data. For example, Word2vec is the most commonly used word embedding model in a variety of text processing applications. However, word embedding methods like Word2vec are commonly applied for feature representation, but VSM-weighted FastText and VSM-weighted Doc2vec cannot be easily found.

Clustering is another major research theme in topic discovery. In a study, the authors compared the performance of various clustering methods on Twitter data centered around the corruption crisis in the FIFA federation [6]. Three clustering algorithms are applied for comparison, namely k-means, k-medoids, and NMF. After evaluation, it was clear that the NMF algorithm presented the best results in the majority of cases. Another study used the agglomerative clustering algorithm for topic detection [7] on the tweets dataset and showed that 80% accuracy was achieved for detecting the topics. Phan et al. [8] have used LDA and proposed a method to detect latent topics. As far as the idea is concerned, all the above studies have performed clustering using traditional methods. We also performed clustering, but we trained traditional clustering algorithms with various text representation methods for performance improvement. Although cluster analysis is frequently employed, some studies have directly compared clustering algorithms' performance. Still, comparing the topic model and clustering algorithms fed with text representation methods cannot be easily found.

3 Methodology

This section describes the dataset collection, data preprocessing and preparation steps, text representation, and clustering algorithms, and evaluation measures. The final output is a set of word clouds that visualizes the major sub-topics discussed on the public health topic. Figure 1 is outlining the methodology followed for our research work.

We recorded tweets containing the keyword "public health" or the hashtag "publichealth" between October 16, 2019, and November 15, 2019. Thereby, a total of 79,085 tweets were collected. Next, we removed the stop words, URLs, and performed stemming, and considered terms with more than two characters. As our data do not contain any Ground Truth (GT) label, so we provided the topic label to the top 50 high-count hashtags. We used these topic labels as the Ground Truth (GT) labels. From the dataset, we removed the hashtag #publichealth, #health, #APHA,

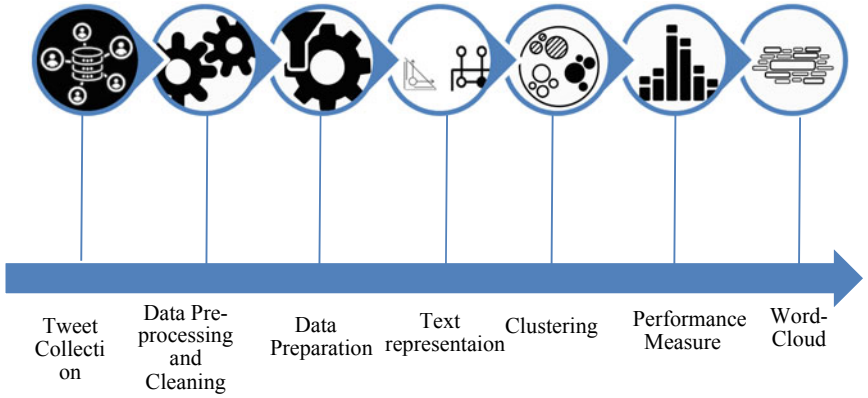


Fig. 1 Process pipeline for tweet clustering and topic discovery

#APHA2019, #EPA, and #science. The top 50 hashtags with at least 200 related tweets under one label were then filtered out. We considered the related hashtags under one label. As a result, we obtained 9,438 tweets having 15 hashtags representing topic labels. We summarize the hashtags' results with assigned labels and their counts in Table 1.

In text representation, word embeddings are a popular approach to bring extra semantic features where words with similar meanings have similar representations. However, since word embeddings treat each word equally, it becomes difficult to determine the words holding higher value over others concerning the particular document being classified. Therefore, we use word embeddings for individual word representation and tf-idf for determining their weights. Next, weights are applied to each word vector and then averaged across the document to achieve an overall text representation. The steps followed for creating weighted vector representations are

Table 1 Hashtags from the dataset with labels and counts

Label#	Hashtags	Label#	Hashtags
0	#wevapewevote, #vaping, #vapingsaveslives, #vapeban, #cannabis	8	#polio, #endpolio
1	#climatechange, #climatecrisis	9	#flushot
2	#airpollution, #delhipollution, #pollution, #environment, #delhiairquality, #airquality, #delhichokes, #delhiairemergency	10	#opioidcrisis, #addiction
3	#mentalhealth	11	#cancer
4	#vaccineswork, #vaccines	12	#gunviolence
5	#hiv	13	#amr
6	#alcoholawareness	14	#youngalzact
7	#nutrition		

Table 2 Outline of methods used for text representation

S. No.	Methods
1	Word2vec
2	FastText
3	Doc2vec
4	VSM-weighted Word2vec
5	VSM-weighted FastText
6	VSM-weighted Doc2vec

explained in Algorithm 1. Table 2 outlines the text representation methods used in this study.

Algorithm 1 Weighted vector representation

Input: a collection of tweets

Output: topically similar weighted word vectors

Begin

1: For each tweet T_j do

2: For each word t_i in T_j do

3: Assign the weight w_{ij} using tf-idf

4: apply learned weight w_{ij} to the word vector wv_{ij} from word embedding

5: Average weighted word vector across the tweet T_j

6: End For

7: End For

End

3.1 Word Embedding Models

A Word2vec is a two-layer neural network to produce a mapping of each token in a corpus to a low-dimensional vector representation in the fixed-size vector space. There are two model architectures of Word2Vec that can be used for computing continuous vector representations of words: (1) continuous bag of words model (CBOW) and (2) skip-gram model. We trained Word2vec models with the CBOW method [9] as it not only provides better accuracy and representations for frequent words but also uses words from the past and the future.

FastText is a method for performing word representations and sentence classification using character-level information. Like Word2vec, this method also supports training CBOW or skip-gram models. For the FastText method, we experimented with the CBOW method.

Next, we trained our text data using the Doc2vec model. A Doc2vec model is a neural network that transforms each document in a corpus with a fixed dimension to a dense vector [10]. Doc2vec, unlike Word2vec and FastText, attempts to teach users how to project a document into vector space. The distributed bag of words (dbow) approach was used to train the Doc2vec model. Since it trains word vectors and document vectors in the same vector space, this approach improves document embedding interpretation.

For all the text representation models, we analyzed the performance change with the number of epochs. As a result, we obtain the following value for the number of optimum epochs: Word2vec (350), FastText (400), Doc2vec (250), VSM + Word2vec (150), VSM + FastText (100), and VSM + Doc2vec (100).

3.2 Clustering

After obtaining the text representation in the vector form for the collected data, the next step is to group similar words to identify the most-posted topics in the dataset. We perform our experiments on three different clustering algorithms: k-means, NMF, and HAC, along with LDA, a topic modeling approach. The clustering algorithms were trained with feature representation methods using the optimum epoch values obtained. All clustering algorithms are executed using the Python language version 3.7.4.

The k-means [11] is the most widely used clustering algorithm that aims to divides the "n" data points into "k" clusters such that the sum of distances of each data point within the cluster and to their cluster's center is minimum. In this paper, we implemented a cosine distance with k-means. In the paper, the value of k is 15, i.e., equal to the number of topic labels.

Next, we cluster tweets using bottom-up agglomerative clustering [12]. In agglomerative techniques, the clustering of texts is based on their similarity in the vector space. We begin implementing this algorithm with the cosine metric and average linkage.

Lastly, non-negative matrix factorization is applied for the clustering of tweets. The NMF [13] method is a matrix factorization method, the constraint of the matrices to be non-negative. In the paper, we also included one topic modeling approach, i.e., LDA, for clustering the tweets. The number of clusters or topics used in the LDA model was the same as the number of assigned labels in the evaluated data. The remaining hyper-parameters were possessed with the default values from the Gensim 3.8.1 package.

3.3 Performance Measures

Normalized mutual information (NMI) is a commonly used measure that normalizes the mutual information between two random variables to take normalized values in the range [0, 1], with 1 representing mutual information and 0 representing disagreement. The formulation for NMI is given as follows:

$$\mathrm{NMI}(X, Y) = \frac{\mathrm{MI}(X, Y)}{\sqrt{H(X) + H(Y)}} \tag{1}$$

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log(p(x_i)) \tag{2}$$

where

X denotes class labels;
Y denotes cluster labels;
$H(X)$ and $H(Y)$ are the entropies of X and Y.

ARI is an external cluster validation measure calculated as:

$$\mathrm{ARI} = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)} \tag{3}$$

a represents the number of objects that are assigned to the same cluster in X and Y;
b represents total pairs that are correct dissimilar in X and Y;
c represents the total pairs that belong to the same class of X but to different classes of Y;
d represents the total pairs that belong to different classes of X and the same class of Y.

Adjusted mutual information (AMI) is an adjustment of the mutual information, which has been computed as:

$$\mathrm{AMI} = \frac{\mathrm{MI} - E\{\mathrm{MI}\}}{\max\{\mathrm{MI}\} - E\{\mathrm{MI}\}} \tag{4}$$

where $E\{\mathrm{MI}\}$ denotes the expected value of MI [14]. Like ARI, it also takes values between 0 and 1.

4 Analysis and Results

These clustering algorithms' performance with different text representation methods is represented in Table 3. It is clear from the figure that unweighted FastText gave the best performance than any other text representation method. Also, we found that VSM-weighted methods performed better than unweighted methods for Word2vec and Doc2vec. The embedding of tf-idf scores has lifted the performance lifted, especially for Word2vec. We run the NMF algorithm for all the text representation methods. Even for this algorithm, FastText has performed far better than other methods. This result is the same as the previous algorithms, where FastText has shown the best results. Even for NMF weighted method resulted in better performance than weighted methods, but the same result is not observed for FastText. A relatively significant drop was observed between the NMI and AMI values for some methods, indicating the importance of chance adjustment for AMI.

Table 3 presents results for the k-means, HAC, and NMF algorithm: The first three columns indicate the NMI score, the next three columns give the AMI score, and the last three columns provide the AMI score. The last row of the table is representing the performance of the LDA model. It is clear from the table that the k-means clustering trained with FastText method exceeds the other methods with NMI, AMI, and ARI scores of 0.3646, 0.3426, and 0.2518, respectively, and unweighted Word2vec gave the worst performance with AMI and ARI scores of 0.0052 and 0.0019 from HAC algorithm and NMI of 0.0078 with NMF algorithm.

All the algorithms gave a close result for both weighted and unweighted methods of Word2vec. Even for the FastText method, k-means and HAC gave very close scores on all three performance measures. But, we can note for both Doc2vec methods, k-means and HAC gave different results. HAC performed better with the weighted Doc2vec method, whereas k-means performed better for unweighted Doc2vec. Also,

Table 3 Comparison of different methods in terms of NMI, AMI, and ARI

Text representation method	NMI			AMI			ARI		
	k-means	HAC	NMF	k-means	HAC	NMF	k-means	HAC	NMF
Word2vec	0.0099	0.0097	0.0078	0.0054	0.0052	0.0052	0.0019	0.0019	0.0020
VSM + Word2vec	0.0776	0.0777	0.0758	0.0459	0.0458	0.0453	0.0214	0.0201	0.0211
FastText	**0.3646**	0.3641	0.3377	**0.3426**	0.3382	0.3192	**0.2518**	0.247	0.2058
VSM + FastText	0.0771	0.0783	0.0925	0.0463	0.0468	0.0283	0.0206	0.0203	0.0091
Doc2vec	0.0711	0.0522	0.0606	0.0636	0.0451	0.0541	0.0295	0.0162	0.0169
VSM + Doc2vec	0.0776	0.0727	0.0765	0.0459	0.0581	0.0459	0.0214	0.0225	0.0211
LDA	0.0552			0.0482			0.0384		

The bold values represent the best result for NMI, AMI, and ARI

it can be noted that NMF performance for all the weighted Doc2vec methods was similar to k-means.

It can be noted from Table 3 that weighted methods for every clustering method gave better NMI, AMI, and ARI scores than unweighted methods. But, the same was not observed for the FastText method, where the unweighted performed better than weighted. It could be because, for FastText, a word may have several n-grams and have lost the information by breaking the text into separate words. Again, though, the weighted FastText performed better than other weighted methods. So, we can conclude that training the word embedding methods with VSM may give more accurate and faster results on short text data.

It is also noted that almost all the text representation methods embedded with clustering algorithms gave better results than the LDA topic model for NMI. Still, a comparable performance is observed for both AMI and ARI. Thus, overall, we can say k-means have given the best results with most text representation methods.

4.1 Topic Interpretation

The work done in this paper establishes the baseline for the topic interpretation. The practicality of a topic identification model can be decided by how accurately identifiable the resulting topics are. This section addresses this question and explores the resulting clusters from the k-means algorithm embedded with FastText representation.

The topic labels are listed in Table 4. The first two columns represent the assigned label and the top five hashtags of each cluster we created in Sect. 3.3, respectively. The third and fourth columns contain the top five hashtags, one based on the frequency, and other based on tf-idf scores, of each cluster from k-means with FastText.

Now, we can conclude that the FastText clustering has correctly represented the latent label hashtags.

5 Word Cloud

This section computed each cluster's word cloud obtained from k-means with the FastText method using WordItOut for empirical evaluation.

Figure 2 presents the terms—as word clouds—that appear in each cluster of the k-means algorithm fed with FastText. The following topics can be identified:

Table 4 Top five hashtags for each cluster

Label	Top hashtags for each cluster (by frequency) Method: manual	Top hashtags for each cluster (by frequency) Method: k-means + FastText	Top hashtags for each cluster (by tf-idf score) Method: k-means + FastText
0	Wevapewevote, vaping, vapingsaveslive, vapeban, notabot	Wevapewevote, vaping, vapingsaveslives, vapeban, notabot	Wevapewevote, vaping, vapingsaveslives, vapeban, notabot
1	Climatechange, climatecrisis, climate, climateemergency, climateaction	Climatechange, climatecrisis, climate, climatechangeshealth, climateemergency	Climatechange, climatecrisis, climate, climatechangeshealth, climateemergency
2	Airpollution, Delhipollution, Delhi, Delhiairquality, pollution	Airpollution, Delhipollution, Delhi, Delhiairquality, pollution	Airpollution, Delhipollution, Delhi, Delhiairquality, pollution
3	Mentalhealth, healthcare, mentalhealthawareness, suicideprevention, nutrition	Mentalhealth, healthcare, environmentalhealth, nutrition, environment	Mentalhealth healthcare, environmentalhealth, nutrition, environment
4	Vaccineswork, vaccines, vaccination, vaccine, vaccinessavelives	Vaccineswork, vaccines, vaccinessavelives, vaccine, vaccination	vaccineswork, vaccines, vaccination, vaccinessavelives, vaccine
5	Hiv, prep, aids, hcv, harmreduction	Hiv, flushot, mentalhealth, nutrition, amr	Hiv, flushot, mentalhealth, nutrition, amr
6	Alcoholawareness, alcohol, alcoholact, alcoholawarenessweek, alcoholic	Alcoholawareness, alcohol, fetalalcoholsyndrome, youngcarers, alcoholawarenessweek	alcoholawareness, alcohol, alcoholawarenessweek, fetalalcoholsyndrome, youngcarers
7	Nutrition, vegetarian, plantbaseddiet, wholefood, eatrawfood	Vegetarian, wholefooddiet, plantbaseddiet, wholefood, eatrawfood	eatrawfood, plantbaseddiet, vegetarian, wholefooddiet, wholefood
8	Polio, endpolio, vaccineswork, Pakistan, poliofreepakistan	Polio, vaccineswork, endpolio, Pakistan, poliofreepakistan	Polio, vaccineswork, endpolio, pakistan, poliofreepakistan
9	Flushot, flu, fluseason, influenza, fluvaccine	Flushot, fluvaccine, flu, fluvaccine, influenza	Flushot, fluvaccine, flu, fluvaccine, influenza
10	Opioidcrisis, opioidepidemic, opioid, opioids, addiction	Opioidcrisis, addiction, opioidepidemic, hiv, opioid	Opioidcrisis, addiction, opioidepidemic, hiv, opioid
11	Nciods, cancer, breastcancer, radon, hpv	Nciods, gunviolence, vaping, mentalhealth, airpollution	Nciods, gunviolence, mentalhealth, vaping, airpollution
12	Gunviolence, violence, endgunviolence, hkpolice, standwithhongkong	Wevapewevote, vaping, gunviolence, airpollution, hiv	Wevapewevote, vaping, gunviolence, airpollution, hiv
13	Amr, antibioticresistance, antibiotics, antimicrobialresistance, microbiology	Cannabis, hiv, amr, antimicrobialresistance, antibiotics	Cannabis, hiv, antibioticresistance, amr, antibiotics

(continued)

Table 4 (continued)

Label	Top hashtags for each cluster (by frequency) Method: manual	Top hashtags for each cluster (by frequency) Method: k-means + FastText	Top hashtags for each cluster (by tf-idf score) Method: k-means + FastText
14	Youngalzact, hope4alzact, alzheimers, endalz, boldalzheimer	Youngalzact, hope4alzact, boldalzheimer, endalz, youngeralzact	Youngalzact, hope4alzact, boldalzheimer, endalz, youngeralzact



Fig. 2 Word cloud for each cluster generated by k-means + FastText

6 Conclusion

An evaluation approach of document clustering by clustering algorithms, trained using unweighted and tf-idf weighted feature representation methods and topic modeling methods, is presented in this paper. The results showed that the k-means

clustering trained with the FastText embedding might perform well on social media collected text data. In the end, we represented the results using the visualization tool WordItOut to provide visual summarization of the entire collection of tweets. We identified that most tweets discussed the poor climate conditions for the tweets collected with public health hashtag and keywords from the clustering and the word cloud.

References

1. Stieglitz S, Mirbabaie M, Ross B, Neuberger C (2018) Social media analytics—challenges in topic discovery, data collection, and data preparation. Int J Inf Manage 39:156–168. https://doi.org/10.1016/j.ijinfomgt.2017.12.002
2. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022. https://doi.org/10.1016/b978-0-12-411519-4.00006-9
3. Lee DL, Chuang H, Seamons K (1997) Document ranking and the vector-space model. IEEE Softw 14:67–75. https://doi.org/10.1109/52.582976
4. Blei DM (2012) Probabilistic topic models. In: Communications of the ACM. pp. 77–84. https://doi.org/10.1145/2133806.2133826
5. Kwak H, Lee C, Park H, Moon S (2011) What is Twitter, a social network or a news media? Arch Zootec 60:297–300. https://doi.org/10.4321/S0004-05922011000200015
6. Godfrey D, Johns C, Meyer C, Race S, Sadek C (2014) A case study in text mining: interpreting Twitter data from world cup tweets
7. Ifrim G, Shi B, Brigadir I (2014) Event detection in Twitter using aggressive filtering and hierarchical tweet clustering
8. Phan XH, Nguyen LM, Horiguchi S (2008) Learning to classify short and sparse text and web with hidden topics from large-scale data collections. In: Proceeding of the 17th international conference on world wide web 2008, WWW'08, pp 91–99. https://doi.org/10.1145/1367497.1367510
9. 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
10. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In 31st International conference on machine learning, ICML 2014, pp 2931–2939
11. MacQueen J (1967) Some methods for classification and analysis of multivariate observations
12. Ward JH (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc 58:236–244. https://doi.org/10.1080/01621459.1963.10500845
13. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791. https://doi.org/10.1038/44565
14. Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res

Learning from Imbalanced Healthcare Data Using Overlap Pattern Synthesis



Lincy Mathews and Hari Seetha

Abstract Diagnosis in healthcare has become a daunting task owing to the exponential growth of diseases and the complexity it brings along with it. Most diagnostic data are heavily imbalanced. An evident fall out of the traditional learning techniques is that they are mostly subdued by the well-represented class than the under-represented class. Such fall out needs rectification without compromising the accuracy of all the classes within the imbalanced data. Our novel technique "Pattern_Syn" reduces the skew distribution of the imbalanced healthcare data solely relying on the properties of the data. The technique is works on the principle of overlap pattern synthesis to generate new similar patterns for the minority class. The technique is, thus, classifier-independent, data-driven and can produce effective results for advanced machine learning techniques. The implementation ran across various imbalanced healthcare datasets showing a considerable improvement of up to 12% increase in efficiency concerning relevant performance measures. The superiority of the method also is showcased in comparison with SMOTE technique.

Keywords Imbalance · Healthcare · Synthetic instances

1 Introduction

Traditional classifiers have played a pivotal role in many healthcare domains for effective and efficient diagnosis. Widely available repositories are a major source of data for researchers and analysts. One of the basic assumptions of learning from

L. Mathews
Research Scholar, Vellore Institute of Technology, University, Vellore, Tamil Nadu, India

H. Seetha
SCOPE, Vellore Institute of Technology, VIT - AP, Amaravati, Andhra Pradesh, India
e-mail: seetha.hari@vitap.ac.in

L. Mathews (✉)
Department of ISE, MS Ramaiah Institute of Technology, Bengaluru, India
e-mail: lincymm99@gmail.com; lincymm99@msrit.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_35

various data across repositories is the balanced distribution among the data of various categories.

In a two class data, when data of one class do not have enough data to represent itself when compared to the other class, the dataset is deemed as imbalanced. The public healthcare datasets used for disease prediction and analysis are relatively imbalanced in nature.

Learning from imbalanced healthcare data is categorized mainly into two—preprocessing strategy and algorithmic technique. Preprocessing strategy includes sampling strategies. SMOTE and its variants such as M-SMOTE, B-SMOTE [5, 6] have been one of the most successful techniques in various applications of imbalance in data. Drawbacks of the sampling techniques include loss of information [12] when undersampling is performed and possibility of over-fitting with oversampling [8]. Algorithmic techniques adjust the costs of the minority instances by implementing misclassification penalty in most cases [3, 9, 11]. The prerequisite is to form the adaptive cost-sensitive matrix based on the misclassifications of the learning algorithm. The misclassification costs need to be identified in prior to evaluation that is a major limitation for cost-sensitive learning [13, 14, 17]. In paper [16], they proposed bagging ensemble with improved random subspace method to increase the efficiency of the classifier (SVM) in the field of bio-medicine and diabetes. In [15], boosting and bagging techniques were found to be ineffective for cardiac surgery classification. Undersampling and oversampling were found to be effective using the C5 and CHAID classifiers. Dimension reduction in combination with up sampling had been studied by [10] for cytogenetic domain classification. Balanced subsets are created using hierarchical decomposition. The authors of [7] employed semi-supervised technique for gene prediction with logistic regression-based two classifiers. In [2], SMOTE worked the best on the highly imbalanced osteoporosis diagnosis and oversampling techniques with random forest classifier. The authors in have addressed imbalance in healthcare data with categorical attributes.

Our work proceeds to identify a novel pattern overlap technique for imbalanced healthcare data. The technique is classifier-independent and implemented across various healthcare data.

2 Proposed Mining Model—Overlap Pattern Synthesis

This section aims to cover basic principles and assumptions of our proposed methodology. The challenges of imbalanced data and decision tree classifier are put forth to understand the efficiency of our proposed method. This section also briefly explains the method with an illustration.

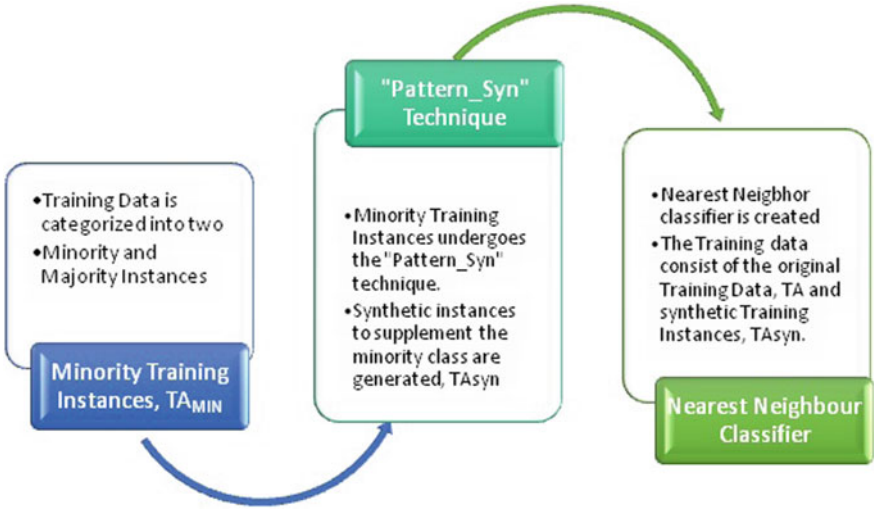


Fig. 1 Proposed methodology—overlap pattern synthesis

2.1 Notations and Definitions

The basic notations and assumptions carried out in the implementation are described in brief. The imbalanced data are denoted as D_{IB}. The imbalanced dataset, D_{IB}, over a set of features, F, is a binary class data. The set of features F is defined over $\{f_1, f_2, \ldots, f_d\}$. The binary class imbalanced data, D_{IB}, contain instances belonging to majority and minority class represented by $C = \{C_{ma}, C_{mi}\}$, respectively. Imbalanced training data, TA, can be split into binary sets of minority training data, TA_{MIN}, and majority training data, TA_{MAJ}.

It can also be represented as follows, $TA = TA_{MIN} + TA_{MAJ}$. Let the number of training instances, $|TA|$, be p, such that $p = q + s$, where $|TA_{MIN}| = q$ and $|TA_{MAJ}| = s$. Test set, represented by TS, is a combination of minority and majority instances.

2.2 Proposed Model

Figure 1 indicates the flow of the proposed model toward classifying imbalanced binary class data using decision tree classifier. The minority training instances are identified from the original training instances to undergo the "Pattern_Syn" technique. This technique produces a set of instances identified as artificial or synthetic instances. The original training instances are supplemented with the new set of minority synthetic instances to even out the skewed distribution of training data.

The "Pattern_Syn" technique consists of the following.

- A three-partition block, represented as π of F, is built based on the minority training instances,TA_{MIN}.
- The partition $\pi = \{M, N, O\}$ creates a vertical partition among the data based on certain conditions of features.
- This concept is also categorized as overlap-based pattern synthesis.

The sections followed will briefly explain the techniques of pattern synthesis.

3 Overlap Pattern Synthesis—Main Principles

For a given feature set, F, for an imbalanced dataset, DIM as mentioned above, a three-block partition is created, $\pi = \{M, N, O\}$, holding the following properties.

- There is dependency or correlation between features of Blocks M and N within data for a given class.
- For features between Blocks M and O and vice versa, there is correlation between the same only through attributes of Block N.
- The existence of the synthetic samples generated might or can exist in the natural state of the data. The non-existence of the synthetic samples could be due the limited collection of the data.

The above method is directed as overlapping synthesis as the overlap between feature set of instance $S1$ and $S2$ can be achieved only if the common feature set "n'' of both the instances, respectively, maps to Block N.

A simple illustration containing the geometric pattern of the OCR number "7" is shown below in Fig. 2. Two samples S1 and S2 of different handwritten patterns in a 2-D rectangular grid of size 5 * 4 are shown.

The pattern or samples $S1$ and $S2$ in row-wise are as follows.

$S1 = \{1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1\}$

$S2 = \{0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0\}$

The region of overlap, the shaded portion, shows the common regions of both the patterns $S1$ and $S2$. The synthetic instances $Syn1$ and $Syn2$ are generated using the "Pattern_Syn" technique by applying the overlap technique.

The synthetic instances generated are illustrated in Fig. 3. The common regions remain the same, while the uncommon regions are overlapped to form new patterns.

Continuing with the above row-wise representation, where feature set $F = \{f_1, f_2, \ldots, f_{20}\}$, let us consider Block $M = \{f_1\}$, $N = \{f_2, f_3, \ldots, f_{14}\}$ and $BlockO = \{f_{15}, f_{16}, \ldots, f_{20}\}$. Thus, $S1$ and $S2$ overlap to form two synthetic samples, $Syn1$ and $Syn2$, where

$Syn1 = \{0, \mathbf{1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0}, 0, 1, 0, 0, 0, 1\}$ and

$Syn2 = \{1, \mathbf{1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0}, 1, 0, 0, 0, 1, 0\}$,

respectively, also as shown figuratively in Fig. 3. The portion in bold forms the overlapping region over which the synthetic instances are created.

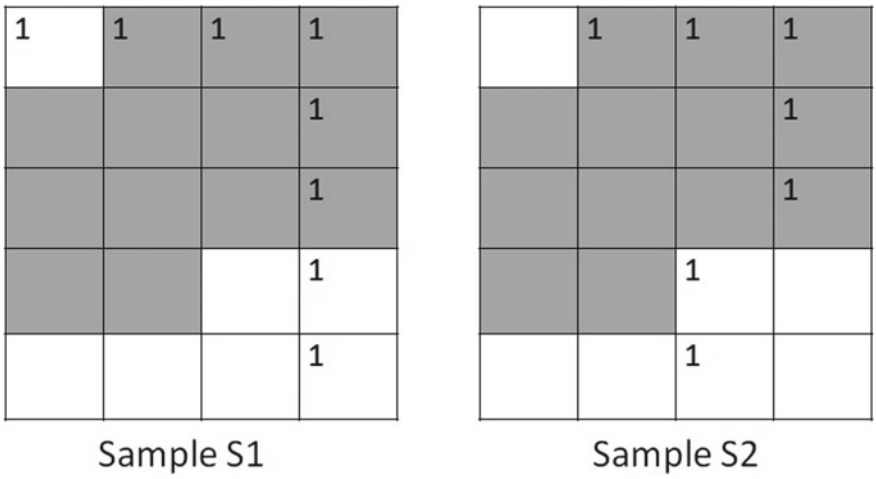


Fig. 2 Illustration of patterns for digit "7"

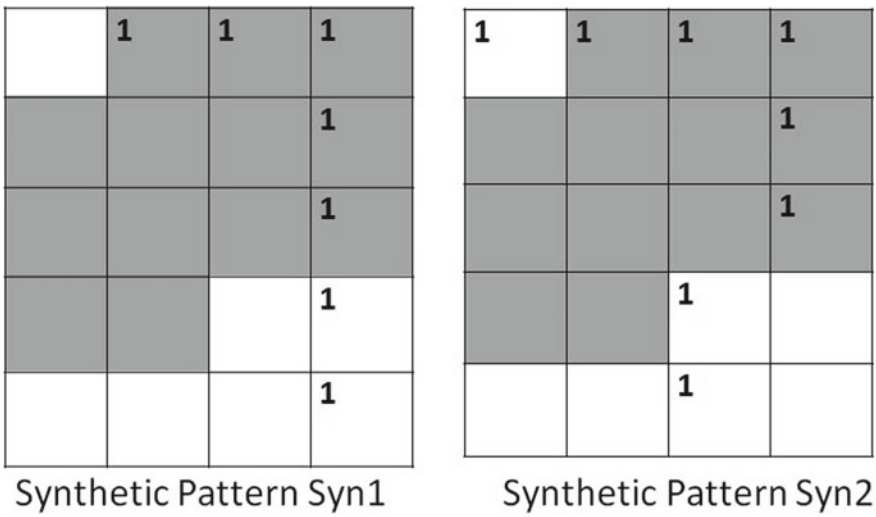


Fig. 3 Illustration of synthetic patterns for digit "7"

3.1 Formal Procedure on Pattern Synthesis—Overlap Principle

For the formal description of a pattern or a sample, P is a feature vector where $P = \{x_1, x_2, \ldots, x_d\}$. The pattern P maps to the feature set F, where $F = \{f_1, f_2, \ldots, f_d\}$. Here, there is an assumption that all instances are drawn from the same class indicating the same distribution within the set of instances.

Pattern $P[f_i]$ defines the value of attribute f_i for the instance P. For the instance $P = \{x_1, x_2, \ldots, x_d\}$, $P[f_i]$ maps to value x_i. Let a subset of features from F be denoted as F_s. Then, the projection of P for the feature set F_s is denoted by P_{F_s}.

Let partition, $\pi = \{M, N, O\}$, be a three partition block of F. M,N and O indicate the three-feature vector. Thus, the projection of feature vectors P, Q and R onto the sub-feature sets M, N and O is indicated as $W = \{P_M, Q_N, R_O\}$.

$W[fi] = P[f_i]\,, if\ f_i \epsilon M$

$= Q[f_i]\,, if\ f_i \epsilon N$

$= R[f_i] if\ f_i \epsilon O$

Let $P = \{P_1, P_2, P_3, \ldots, P_q\}$ be the set of random patterns from the minority training instances TA_{MIN} belonging to class C_{mi}. The synthetic instances generated from P are denoted as P_syn for a three-block partition $\pi = \{M, N, O\}$.

1. Initially, $TA_{syn} = P$
2. Consider a pair of minority training patterns (P_i, P_j) for $1 \leq i < j \leq q$, since $|TA_{MIN}| = q$.
3. If for sub-block N , $P_N^i = P_N^j$, then for two minority patterns of the form (P_M^i, P_N^i, P_O^i) and (P_M^j, P_N^j, P_O^j), two synthetic instances are generated of the form $TA_{syn} = \{(P_M^i, P_N^i, P_O^j), (P_M^j, P_N^j, P_O^i)\}$.

3.2 *Overlap Pattern Synthesis—Illustration and Approximation*

A three-block partition was implemented for experimentation. The number of partitions was decided based on the work followed in ??. Experimentally it was proved that partitions beyond three did not improve or add to the efficiency of the classifier.

For a three-block partition $\pi = \{M, N, O\}$, M is conditionally independent from O given N and belongs to class C_{mi}. The notation for a given random variable N, $I(M, O|N)$ denotes the conditional independence of two random variables M with O. To begin with, assume a three-block feature partition considering the minority training instances only. To formalize the influence between features, correlation coefficient is employed to understand the dependency between features within the data.

The technique of approximation between the features is using the correlation coefficient between attributes for the feature set F. The intra-correlation coefficient of features within each block is high, whereas the inter-correlation coefficient of the adjacent blocks will be low. The objective is to partition the features between the three blocks such the correlation between features within blocks is high.

3.3 Pattern_Syn Technique

After the three-block partition is constructed, the next phase of proposed model is implemented to generate synthetic instances. The synthetic instances are generated based on the three-block partition. The approach is illustrated below with an example.

INPUT

- $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$
- $X = \{(m_1, m_2, m_3, m_4, m_5, m_6), (m_7, m_2, m_3, m_8, m_5, m_9), (m_{10}, m_{11}, m_3, m_8, m_5, m_{11})\}$
- $\pi = \{M, N, O\}$, where $M = \{f_6\}$,$N = \{f_1, f_2, f_4, f_5\}$, $O = \{f_3\}$

OUTPUT

- Synthetic dataset , TA_{syn}

PROCESS

1. Initial set
   $SI_1(X) = \{(m_1, m_2, m_3, m_4, m_5, m_6), (m_7, m_2, m_3, m_8, m_5, m_9), (m_{10}, m_{11}, m_3, m_8, m_5, m_{11})\}$
2. In Block N, f_1, f_2 are two continuous ordered pair of features. However, no two patterns in X have any common values between f_1 and f_2. Hence, $SI_2(X) = X$
3. In Block N, f_4, f_5 are two continuous ordered pair of features. Hence, two samples in $SI_1(X)$ have common values. Overlapping the values of these features, we get two more samples.
   $SI_3(X) = \{(m_1, m_2, m_3, m_4, m_5, m_6), (m_7, m_2, m_3, m_8, m_5, m_9), (m_{10}, m_{11}, m_3, m_8, m_5, m_{11}), (m_7, m_2, m_3, m_8, m_5, m_{11}), (m_{10}, m_{11}, m_3, m_8, m_5, m_9)\}$
4. In Block M, the commonality of values for feature f_6 is observed. However, by overlapping on the same feature, we get instances that are already present in $SI_3(X)$. Hence, no samples will be added in the same set.
   $SI_4(X) = SI_3(X)$
5. In Block O , the commonality of values for feature f_3 is observed. $SI_5(X) = \{(m_1, m_2, m_3, m_4, m_5, m_6), (m_7, m_2, m_3, m_8, m_5, m_9), (m_{10}, m_{11}, m_3, m_8, m_5, m_{11}), (m_7, m_2, m_3, m_8, m_5, m_{11}), (m_{10}, m_{11}, m_3, m_8, m_5, m_9), (m_{10}, m_{11}, m_3, m_{10}, m_{11}, m_3), (m_1, m_2, m_3, m_8, m_5, m_{11})\}$

Table 1 Properties of imbalanced data

Datasets	Features	Imbalance ratio (+ve vs. −ve samples)
Blood transfusion	4	178:570
Ionosphere	34	126:225
Wisconsin	9	239:444
Breast cancer	90	24:336

4 Experimental Details

4.1 Data Description

The data collected for the implementation are from the well-known benchmarked repositories [1, 4]. The dataset description is as given below in Table 1.

4.2 Observations

For evaluating the efficiency and goodness of "Pattern_Syn" technique, decision tree classifier was executed on the original imbalanced data, data balanced by SMOTE technique and data balance improved by "Pattern_Syn" technique. Figure 4 shows the performance of the classifier.

1. The original training data were imbalanced in nature. The decision tree model constructed on the imbalanced training data showed a high misclassification rate among the healthcare and non-healthcare datasets. Looking at the values of recall of Blood Transfusion , only 32% of the samples were accurately classified. Comparatively, decision tree showed better performance of classification for positive samples in Wisconsin and New Thyroid1.
2. Evaluation of our novel technique "Pattern_Syn" has indicated that the technique is efficient for imbalanced healthcare data. The performance measures have indicated an increase from 2 to 8% increase in performance. Healthcare data such as Wisconsin, Blood Transfusion have showed higher accuracy with data modified by "Pattern_Syn" than SMOTE.
3. The technique worked with datasets that have redundant patterns within features. The occurrence of redundant patterns between features is involved in creating synthetic patterns. The existence of sub-patterns within data is a prerequisite to the satisfactory working of the technique.
4. SMOTE performance can be compared to "Pattern_Syn" in improving the skew distribution of the imbalance in data for datasets related to Blood Cancer and

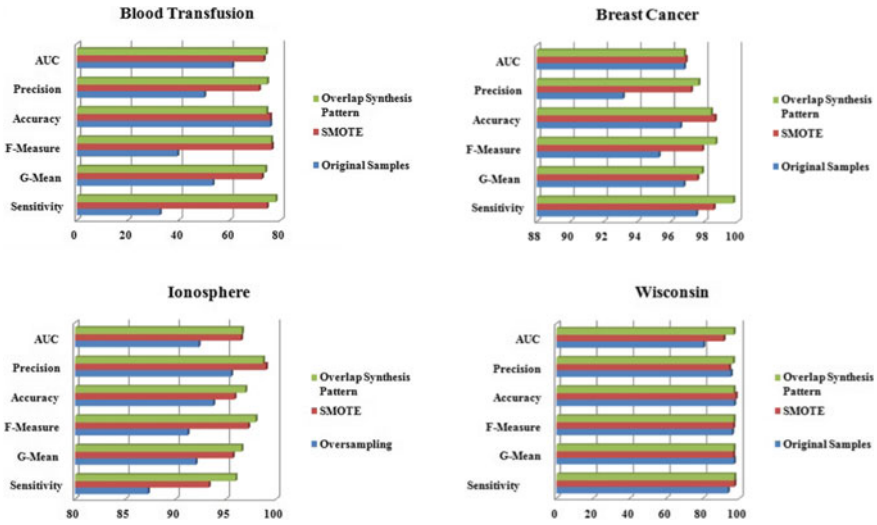


Fig. 4 Performance analysis of decision tree on benchmarked datasets

Blood Transfusion. In terms of computing and complexity, our technique is less complex as it depends on identification of patterns.

5 Conclusion

The paper introduces a novel technique known as the "Pattern_Syn". The preprocessing oversampling technique creates synthetic instances for the minority class. The method initially identifies the redundant patterns inherent in the data. The similar artificial instances for the minority class are generated from inherent patterns based on the pattern overlap principle.

The "Pattern_Syn" technique was investigated and tested on six benchmarked imbalanced data. The technique was tested against the original imbalanced data and data improvised through the SMOTE technique. Our results indicate the visible increase in performance of the classifier over the adopted performance measures. One limitation of the technique is that the efficiency of the technique is based on existence of redundant patterns occurring within the data. If the imbalanced data do not have any inherent patterns among features, the technique might not lead to qualitative synthetic samples for the minority class.

For the future directions, the work can be enhanced to imbalanced big data. The number of features and data increases exponentially. The outcome of the technique can also be modified and monitored for multiclass imbalanced data.

References

1. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic Soft Comput 17
2. Bach M, Werner A, Żywiec J, Pluskiewicz W (2017) The study of under-and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis. Inf Sci 384:174–190
3. Cao P, Zhao D, Zaiane O (2013) An optimized cost-sensitive svm for imbalanced data learning. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 280–292
4. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
5. Fernández A, Garcia S, Herrera F, Chawla NV (2018) Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 61:863–905
6. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
7. Herndon N, Caragea D (2016) A study of domain adaptation classifiers derived from logistic regression for the task of splice site prediction. IEEE Trans Nanobiosci 15(2):75–83
8. Holte RC, Acker L, Porter BW et al (1989) Concept learning and the problem of small disjuncts. In: IJCAI, vol 89, pp 813–818. Citeseer
9. Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2017) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Netw Learn Syst 29(8):3573–3587
10. Lerner B, Yeshaya J, Koushnir L (2007) On the classification of a small imbalanced cytogenetic image database. IEEE/ACM Trans Comput Biol Bioinform 4(2):204–215
11. Qiu C, Jiang L, Li C (2017) Randomly selected decision tree for test-cost sensitive learning. Appl Soft Comput 53:27–33
12. Tang Y, Zhang YQ, Chawla NV, Krasser S (2008) Svms modeling for highly imbalanced classification. IEEE Trans Syst, Man, Cybern, Part B (Cybernetics) 39(1):281–288
13. Wang T, Qin Z, Jin Z, Zhang S (2010) Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. J Syst Softw 83(7):1137–1147
14. Wang T, Qin Z, Zhang S, Zhang C (2012) Cost-sensitive classification with inadequate labeled data. Inf Syst 37(5):508–516
15. Yap BW, Abd Rani K, Abd Rahman HA, Fong S, Khairudin Z, Abdullah NN (2014) An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In: Proceedings of the first international conference on advanced data and information engineering (DaEng-2013). Springer, pp 13–22
16. Yu H, Ni J (2014) An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data. IEEE/ACM Trans Comput Biol Bioinform 11(4):657–666
17. Zhang S (2010) Cost-sensitive classification with respect to waiting cost. Knowl-Based Syst 23(5):369–378

Polynomial Formal Verification of Arithmetic Circuits



Rolf Drechsler, Alireza Mahzoon, and Lennart Weingarten

Abstract The size and the complexity of digital circuits are increasing rapidly. This makes the circuits highly error-prone. As a result, proving the correctness of a circuit is of utmost importance after the design phase. Arithmetic circuits are among the most challenging designs to verify due to their high complexity and big size. In recent years, several formal methods have been proposed to verify arithmetic circuits. However, time and space complexity bounds are still unknown for most of these approaches, resulting in performance unpredictability. In this paper, we clarify the importance of polynomial formal verification for digital designs particularly arithmetic circuits. We also introduce an *Arithmetic Logic Unit* (ALU) and prove that formal verification of this circuit is possible in polynomial time. Finally, we confirm the correctness of the complexity bounds by experimental results.

Keywords Formal verification · Complexity · Polynomial · Arithmetic circuits · Arithmetic logic unit · Binary decision diagram

1 Introduction

With the invention of the transistor back in 1947, the cornerstone for the digital revolution was laid. As a fundamental building block, the transistor enabled the development of digital circuits. Their mass production revolutionized the field of electronics, finally leading to computers, embedded systems, as well as the Internet. Hence, the impact of digital hardware on society, as well as economy, was and

R. Drechsler · A. Mahzoon
Institute of Computer Science, University of Bremen, Bremen, Germany
e-mail: mahzoon@uni-bremen.de

R. Drechsler (✉) · L. Weingarten
Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
e-mail: drechsle@uni-bremen.de

L. Weingarten
e-mail: len_wei@uni-bremen.de

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1_36

is tremendous. Over the last decades, the enormous growth of complexity of integrated circuits continues as expected. As modern electronic devices are getting more and more ubiquitous, the fundamental issue of functional correctness becomes more important than ever. This is evidenced by many publicly known examples of electronic failures with disastrous consequences. This includes the Intel Pentium bug in 1994, the New York blackout in 2003, and a design flaw in Intel's Sandy Bridge chipset in 2011.

Such costly mistakes can only be prevented by applying rigorous verification to the circuits before they get to production [4, 5]. A lot of effort has been put into developing efficient verification techniques by both academic and industrial research. Only recently, the industry has recognized the great importance of automated formal verification (see e.g. functional safety standards such as ISO 26262 [30]). Hence, in the last few years, this research area has become increasingly active. Essentially, the aim of automated formal verification is to automatically prove that an implementation is correct with respect to its specification. Depending on what is an implementation and what comprises a specification, different verification problems arise.

Automated formal verification of arithmetic circuits is one of the most important goals of the verification community in recent years. Arithmetic circuits are usually involved in many applications that require intense computations (e.g., cryptography and signal processing) as well as in architectures for artificial intelligence (e.g., machine learning). They include a wide range of different designs like adders, subtractors, multipliers, and dividers. The function of an arithmetic unit is unique, e.g., a multiplier always computes the product of its inputs. However, the architecture and size of them vary based on the application and design parameters. Most of these circuits are highly parallel and architecturally complex [17, 32]. Therefore, the high complexity and the big size of arithmetic circuits make them very challenging to verify.

During the last 20 years, researchers have proposed several formal methods to verify arithmetic circuits: (a) *Binary Decision Diagram* (BDD) [2, 10] verification methods extract the BDDs for the outputs of an arithmetic circuit with the help of symbolic simulation, then, the output BDDs are evaluated for the correctness, (b) *Boolean Satisfiability* (SAT) [3, 11] verification methods translate the implementation and the specification into one Conjunctive Normal Form (CNF) which is satisfiable if the implementation is correct, (c) *Binary Moment Diagram* (*BMD and K*BMD) [8, 12] approaches use word-level graphs to prove the equivalence between the word-level design specification and the bit-level circuit, (d) term rewriting [13, 27, 28] techniques take advantage of a library of rewrite rules to prove the correctness of arithmetic circuit by several rewritings in a theorem proving system, (e) reverse engineering techniques using *Arithmetic Bit-Level* (ABL) [24, 26] extract an arithmetic bit-level description of the circuit, and then use it for a fast equivalence checking, and (f) *Symbolic Computer Algebra* (SCA) [15, 20–23] methods capture the logical gates as polynomials, then prove the correctness of arithmetic circuit by dividing the word-level specification by the gate polynomials.

Despite the success of formal verification approaches in proving the correctness of a wide variety of arithmetic circuits, the complexity bounds for most of these

approaches are still unknown. It raises serious questions about the scalability of the verification methods. Furthermore, unknown time and space complexities make it difficult to compare two approaches and choose the best one with respect to the type of arithmetic circuit, e.g., which formal method is suitable for verification of an integer adder. In this paper, we first clarify the importance of polynomial formal verification. Then, we review the known researches about polynomial formal verification of arithmetic circuits. Subsequently, we introduce a simple *Arithmetic Logic Unit* (ALU) with 8 operations and prove that its formal verification is possible in polynomial time. For the first time, we also calculate the complexity bounds for verifying a subtractor (i.e., one of the ALU units). We confirm the correctness of obtained complexity bounds by experimental results. Finally, we propose a verification strategy for more advanced ALUs performing complicated operations such as multiplication.

2 Polynomial Formal Verification

In this section, we first clarify the importance of polynomial formal verification. Subsequently, we review the three important works in the field of polynomial formal verification of arithmetic circuits.

2.1 Importance of Polynomial Verification

The state-of-the-art formal verification techniques often give satisfying results for a specific type of arithmetic circuits: (a) BDD and SAT-based verification methods report very good results for different types of adder architectures, (b) *BMDs are used to verify multipliers, and (c) SCA-based approaches are employed for the verification of complex multipliers and dividers.

However, the main shortcoming of these techniques is the unpredictability in performance, leading to several verification problems:

1. It cannot be predicted before actually invoking the verification tool whether it will successfully terminate or run for an indefinite amount of time; e.g., it is not clear whether SAT-based verification can successfully verify all types of adders or it runs forever for some of them.
2. The scalability of these techniques remains unknown, i.e., it is not predictable how much the run-time and the required memory increase when the size of the circuit under verification grows; e.g., it is not obvious how scalable is a *BMD-based verification technique when it comes to proving the correctness of an integer multiplier.
3. It is not possible to compare the performance of verification methods for a specific design and choose the best one, e.g., it is not clear which verification technique

has a better performance when it comes to proving the correctness of an integer adder.

In order to resolve the unpredictability of a verification method, its time and space complexity should be calculated. Knowing the complexity bounds for a verification technique alleviates the three aforementioned verification problems. We are particularly interested in optimized time and space bounds with the smallest possible polynomial order, i.e., $\mathcal{O}(n^m)$ where n is the number of input bits and m is a positive number. A formal verification method with a polynomial complexity (time and space) is scalable and can be carried out successfully.

In the next section, we review the three works in the field of polynomial formal verification of arithmetic circuits and illustrate their advantages and disadvantages.

2.2 Research Works in Polynomial Verification

During the last 20 years, researchers have proposed several formal verification methods to verify different types of arithmetic circuits. Despite rapid progress in developing formal methods, the research on the complexity bounds of verification methods is very limited. We now review the three notable works on the polynomial formal verification of arithmetic circuits.

2.2.1 Polynomial Formal Verification of Adders Using BDDs

It is known for a long time that BDDs are very efficient in practice when it comes to the verification of adders. However, it has not been proven theoretically until recently. PolyAdd [6] showed for the first time that the complete formal verification process for three types of adders (i.e., ripple carry adder, conditional sum adder, and carry look-ahead adder) can be carried out polynomially. It was achieved by proving that the size of BDDs remains polynomial with respect to the number of input bits during the symbolic simulation. Although PolyAdd successfully shows that the polynomial formal verification of the three adder architectures is possible, it has two limitations: The proofs cannot be extended to the other adders, e.g. parallel prefix adders, and it does not calculate the exact order of complexity bounds. Recently, some progress has been made in calculating the exact verification complexities of adder architectures, e.g., the exact complexity bounds have been obtained for ripple carry adder and conditional sum adder in [18, 19].

2.2.2 Polynomial Formal Verification of Multipliers Using *BMDs

Verification of multipliers became possible after the development of word-level decision diagrams. Particularly, *BMDs and K*BMDs reported very good results for

verifying several types of integer multipliers. However, [16] is the only work on the polynomial formal verification of multipliers using *BMDs. In the paper, the authors analyzed *BMD-based verification by backward construction applied to the class of Wallace-tree-like multipliers. They formally proved polynomial upper bounds on run-time and space requirements with respect to the number of input bits. They showed that the whole verification process is bounded by $\mathcal{O}(n^2)$ and $\mathcal{O}(n^4)$ with respect to space and time, where n in the number of input bits. The proof in the paper is only for Wallace-tree-like multipliers and it does not support other classes of multipliers.

2.2.3 Polynomial Formal Verification of Multipliers Using SCA

Recently, the SCA-based verification methods have been very successful in proving correctness of a large range of multiplier architectures. AMulet [15] is one of the SCA-based methods that is developed for verification of complex multipliers. The authors have shown in [14] that AMulet has a polynomial complexity when it comes to the verification of btor-multipliers (Generated by Boolector) and Wallace-tree multipliers with Booth encoding. However, the proof for the second multiplier type is empirical, and it has not been done theoretically due to the irregular structure of the multiplier.

In addition to these three works, some research works have been recently done on polynomial BDD construction of totally symmetric functions [9] and polynomial formal verification of tree-like circuits [7].

3 Polynomial Formal Verification of a Simple ALU

In this section, we first introduce a simple ALU with 8 operations. Then, we give a brief overview of the BDD-based verification. Subsequently, we calculate the time complexity of verifying the ALU and show that polynomial formal verification is possible for this architecture. Finally, we confirm our theoretical calculations by experimental results.

3.1 ALU Overview

An ALU is a combinational digital circuit that performs arithmetic and bitwise operations on integer binary numbers. The type and the number of supported operations in an ALU depend on the application. Figure 1 shows the symbolic representation of a general ALU. It receives two n-bit inputs a and b as well as an input carry c. The operation between the inputs is determined by an m-bit $select$.

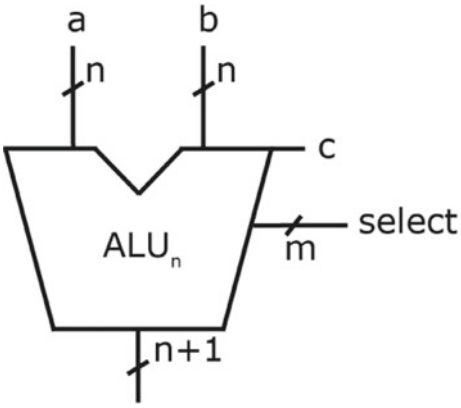


Fig. 1 Symbolic representation of the ALU

Table 1 List of supported operations

s_2	s_1	s_0	function
0	0	0	$0 \ldots 0$
0	0	1	$b - a$
0	1	0	$a - b$
0	1	1	$a + b + c$
1	0	0	$a \oplus b$
1	0	1	$a \vee b$
1	1	0	$a \wedge b$
1	1	1	$1 \ldots 1$

In this paper, we consider a simple ALU with 8 operations, i.e. the *select* signal has 3 bits. The complete list of supported operations is depicted in Table 1. The ALU can perform two arithmetic operations (i.e., addition and subtractions) as well as three bitwise logical operations (i.e., XOR, OR, and AND). The detailed architecture of the ALU is shown in Fig. 2.

3.2 BDD-based Verification

Definition 1 A *Binary Decision Diagram* (BDD) is a directed, acyclic graph. Each node of the graph has two edges associated with the values of the variables 0 and 1. A BDD contains two terminal nodes (leaves) that are associated with the values of the function 0 or 1.

Definition 2 An *Ordered Binary Decision Diagram* (OBDD) is a BDD, where different variables appear in the same order in each path from the root to a leaf.

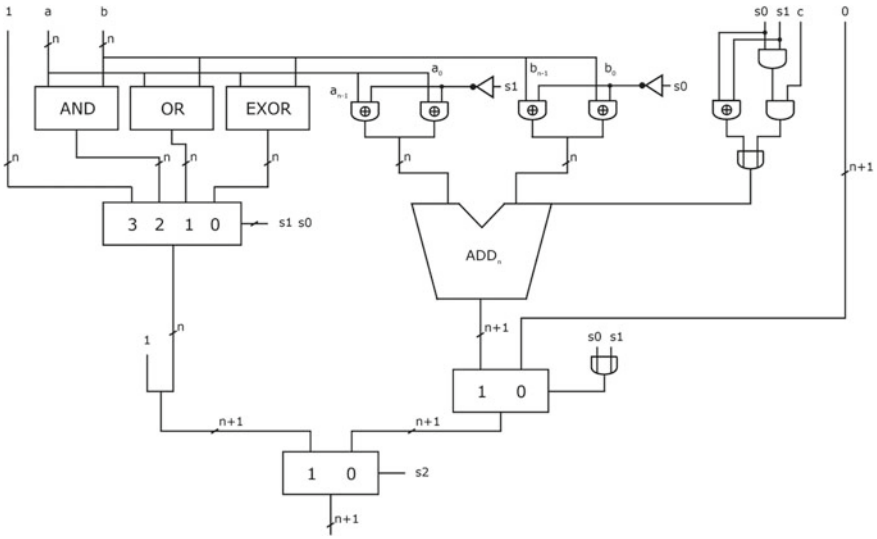


Fig. 2 ALU architecture

Definition 3 A *Reduced Ordered Binary Decision Diagram* (ROBDD) is an OBDD that has a minimum number of nodes for a given variable order. The ROBDD of a Boolean function is always unique.

The ITE operator (If-Then-Else) is used to calculate the results of the logical operations in BDDs:

$$ITE(f, g, h) = (f \wedge g) \vee (\bar{f} \wedge h) \tag{1}$$

The basic binary operations can be translated into the ITE operator:

$$\begin{aligned} f \wedge g &= ITE(f, g, 0), \\ f \vee g &= ITE(f, 1, g), \\ f \oplus g &= ITE(f, \overline{g}, g), \\ f \odot g &= ITE(f, g, \overline{g}), \\ \bar{f} &= ITE(f, 0, 1) \end{aligned} \tag{2}$$

ITE can be also used recursively in order to compute the results:

$$ITE(f, g, h) = ITE(x_i, ITE(f_{x_i}, g_{x_i}, h_{x_i}), ITE(f_{\overline{x}_i}, g_{\overline{x}_i}, h_{\overline{x}_i})) \tag{3}$$

where f_{x_i} ($f_{\overline{x}_i}$) is the positive (negative) cofactor of f with respect to x_i, i.e., the result of replacing x_i by the value 1 (0).

Algorithm 1 If-Then-Else (ITE)

Require: f, g, h BDDs
Ensure: ITE BDD
1: **if** terminal case **then**
2: **return** $result$
3: **else if** computed-table has entry $\{f, g, h\}$ **then**
4: **return** $result$
5: **else**
6: $v =$ top variable for f, g, or h
7: $t = ITE(f_{v=1}, g_{v=1}, h_{v=1})$
8: $e = ITE(f_{v=0}, g_{v=0}, h_{v=0})$
9: $R = FindOrAddUniqueTable(v, t, e)$
10: $InsertComputedTable(\{f, g, h\}, R)$
11: **return** R

ITE operations can be computed with the help of Algorithm 1. The result is obtained recursively based on Eq. (3) in this algorithm. During the calculations, the sub-diagrams of f, g, and h are the arguments for subsequent calls to the ITE sub-routine. The number of sub-diagrams in a BDD is equal to the number of nodes. For each of the three arguments, the sub-routine is called at most once. Assuming that *Unique Table* is searched at a constant time, the computational complexity of the ITE algorithm, even in the worst-case, does not exceed $\mathcal{O}(|f| \cdot |g| \cdot |h|)$, where $|f|$, $|g|$ and $|h|$ denote the size of the BDDs in terms of the number of nodes [1].

In order to formally verify a circuit, we need to have the BDD representation of the outputs. Symbolic simulation helps us to obtain the BDD for each primary output. During a simulation, an input pattern is applied to a circuit, and the resulting output values are checked to see whether they match the expected values. On the other hand, symbolic simulation verifies a set of scalar tests (which usually cover the whole input space) with a single symbolic test. Symbolic simulation using BDDs is done by generating corresponding BDDs for the input signals. Then, starting from primary inputs, the BDD for the output of a gate (or a building block) is obtained using the ITE algorithm. This process continues until we reach the primary outputs. Finally, the output BDDs are evaluated to see whether they match the BDDs of the circuit.

We now prove that verification of the introduced ALU is possible in polynomial time if we use the BDD-based verification. To do this, we should first calculate the complexity bounds for each ALU operation.

3.3 Complexity Bounds of Arithmetic Units

The introduced ALU can perform two arithmetic operations, i.e., addition and subtraction. We first focus on addition. We assume that the adder is a ripple carry adder as shown in Fig. 3.

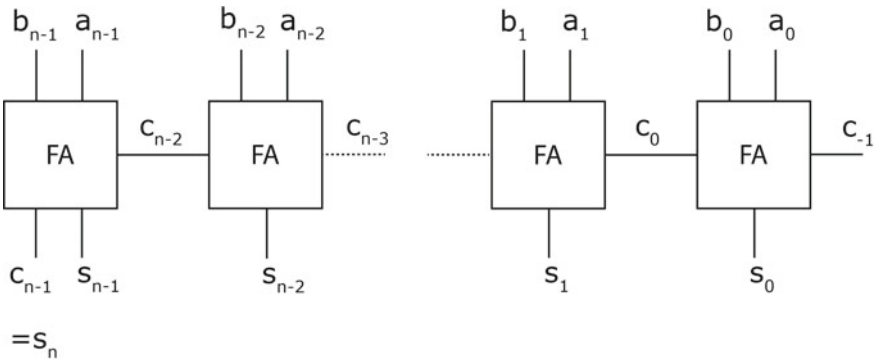


Fig. 3 Ripple carry adder

In order to obtain the computational complexity of an n-bit ripple carry adder, we first calculate the complexity of BDD construction for a single FA. The sum and carry bits of a FA can be shown by ITE operations as follows:

$$S_i = A_i \oplus B_i \oplus C_{i-1} = ITE(C_{i-1}, A_i \odot B_i, A_i \oplus B_i) = ITE(C_{i-1}, ITE(A_i, B_i, \overline{B}_i), ITE(A_i, \overline{B}_i, B_i)), \quad (4)$$

$$C_i = (A_i \wedge B_i) \vee (A_i \wedge C_{i-1}) \vee (B_i \wedge C_{i-1}) = ITE(C_{i-1}, A_i \vee B_i, A_i \wedge B_i) = ITE(C_{i-1}, ITE(A_i, 1, B_i), ITE(A_i, B_i, 0)) \quad (5)$$

The ITE operations are computed by Algorithm 1 to get the BDDs for the S_i and C_i signals. Assuming that f, g and h are the input arguments of an ITE operator, the computational complexity is computed as $|f| \cdot |g| \cdot |h|$. As a result, the complexity of computing S_i and C_i is as follows:

$$\text{Complexity}(S_i) = |C_{i-1}| \cdot |A_i|^2 \cdot |B_i|^2 \cdot |\overline{B}_i|^2 = 729 \cdot |C_{i-1}| \quad (6)$$

$$\text{Complexity}(C_i) = |C_{i-1}| \cdot |A_i|^2 \cdot |B_i|^2 = 81 \cdot |C_{i-1}| \quad (7)$$

where A_i, B_i, and $\overline{B}_i$ BDDs have only one internal node and two terminal nodes; thus, the size of them is the same and equals 3.

It has been proven in [29] that the BDD size of the ith carry bit (C_i) is $3 \cdot i + 6$. Thus, the overall complexity of verifying a ripple carry adder can be obtained as follows:

$$\text{complexity}_{[RCA]} = 810 \cdot \sum_{i=0}^{n-1} |C_{i-1}| = 2430 \cdot \sum_{i=0}^{n-1} (i+1) = 1215n^2 + 1215n \quad (8)$$

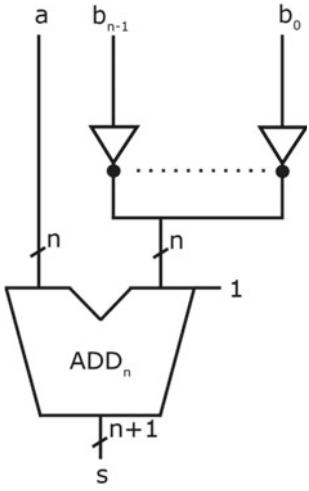


Fig. 4 Subtractor structure

We can conclude that the order of the verification complexity is $\mathcal{O}(n^2)$, where n is the number of bits per input (i.e., size of the adder). As a result, proving the correctness of a ripple carry adder has quadratic time complexity.

Figure 4 shows the structure of a subtractor. In a subtractor architecture, the bits of one of the inputs are negated and the carry signal is set to one. The time complexity of negating n bits is of linear order with respect to the number of input bits. After the negation, the size of the BDD does not change, thus the same calculations for obtaining the complexity bounds of an adder are applicable to a subtractor. As a result, formal verification of a subtractor has quadratic time complexity, i.e., $\mathcal{O}(n^2)$.

3.4 Complexity Bounds of Logic Units

The introduced ALU has three bitwise logical operations, i.e., XOR, OR, and AND. Each bitwise operation is done by n gates. The time complexity of obtaining the output BDD of a gate (e.g., $a_i \oplus b_i$) is constant. Thus, the overall time complexity is linear (i.e., $\mathcal{O}(n)$) with respect to the number of input bits.

The arithmetic units have a bigger order of time complexity in comparison to the logic units. Therefore, they determine the overall bounds. Consequently, verification of the simple ALU has quadratic time complexity, i.e., $\mathcal{O}(n^2)$.

Table 2 Run-time of verifying ALUs (seconds)

Size	Run-time
1024	37.51
2048	83.51
3072	100.67
4096	111.30
5120	129.93
6144	146.82
7168	160.25
8192	171.81
9216	187.57
10240	206.79

3.5 Experimental Results

We have implemented the ALU in Fig. 2 in Verilog. The size of the ALU is a parameter, and it can be set before the synthesis. Thus, we can easily generate ALUs of different sizes. The design has been synthesized using Yosys [31]. We have also implemented the BDD-based verifier in C++. The tool takes advantage of the symbolic simulation to obtain the BDDs for the primary outputs. In order to handle the BDD operations, we used the CUDD library [25]. All experiments are performed on an Intel(R) Xeon(R) CPU E3-1275 with 3.60 GHz and 64 GByte of main memory.

In order to verify the ALU, we first set a value to *select* signal (e.g., 011), thenwe obtain the output BDDs using our verifier. Finally, we evaluate the BDDs to check whether they match the corresponding operation (e.g., addition). We repeat the process for all possible values of *select* to cover all operations.

Table 2 reports the verification run-times for simple ALUs. We have done the experiments for 10 ALUs of different sizes. The first column **Size** denotes the size of the ALU based on the number of bits per input. The run-time (in seconds) of the BDD-based verification method is reported in the second column **Run-time**.

It is evident in Table 2 that the BDD-based based verification reports very good results. An ALU with 10240 bits per input, which consists of more than 700K gates, can be verified in less than 4 min. Thus, the experimental results for the simple ALU confirm the scalability of the BDD-based verification method.

4 Polynomial Formal Verification of Advanced ALUs

In the previous section, we proved that polynomial formal verification of a simple ALU is possible. However, the ALUs which are used in practice are more complicated, and they contain arithmetic units such as multipliers. Thus, the BDD-based

verification fails for these advanced ALUs, as it has an exponential time and space complexity for multiplier units. It is a general problem of monolithic verification strategies in which only one formal verification technique is used to prove the correctness of the whole design.

As we mentioned in Sect. 2.1, one of the advantages of knowing the complexity bounds is the possibility of comparing verification methods and choosing the best one for a specific design. When it comes to the verification of an advanced ALU, we can use different verification methods to prove the correctness of each operation based on the *select* signal, i.e., a verification method with the smallest order of complexity is employed for verifying a specific operation. This hybrid verification strategy (i.e., using different verification methods to verify each operation) makes the polynomial formal verification of an advanced ALU possible. Moreover, it also keeps the run-time and memory usage low.

As an example, a hybrid verification strategy can be used to verify an advanced ALU containing a multiplier unit. *BMD-based verification method reported good results for proving the correctness of multipliers, and its time and space complexity are polynomial [16]. On the other hand, we proved that polynomial formal verification of adder, subtractor, and logic units is possible using BDD-based verification. Consequently, we use *BMD-based verification for the multiplication and the BDD-based verification for the rest of the operations.

In the future, we plan to calculate the complexity bounds of more verification methods. It helps us to compare the complexities and choose the best technique for the verification of a specific operation. As a result, the hybrid verification achieves better performance in terms of run-time and memory usage.

5 Conclusion

In this paper, we clarified the problem of performance unpredictability in the field of formal verification. We then discussed the importance of polynomial formal verification of arithmetic circuits and reviewed the most notable works in this research area. Subsequently, as an example, we proved the complexity bounds of a simple ALU using BDD-based verification and confirmed the theoretical calculations by experimental results. Finally, we proposed the idea of hybrid verification to prove the correctness of advanced ALUs containing complicated arithmetic units such as multipliers.

Acknowledgements This work was supported by the German Research Foundation (DFG) within the Reinhart Koselleck Project *PolyVer: Polynomial Verification of Electronic Circuits* (DR 287/36-1).

References

1. Brace KS, Rudell RL, Bryant RE (1990) Efficient implementation of a BDD package. In: Design automation conference, pp 40–45
2. Bryant RE (1995) Binary decision diagrams and beyond: enabling technologies for formal verification. In: International conference on computer-aided design, pp 236–243
3. Disch S, Scholl C (2007) Combinational equivalence checking using incremental SAT solving, output ordering, and resets. In: ASP design automation conference, pp 938–943
4. Drechsler R (2004) Advanced formal verification. Kluwer Academic Publishers
5. Drechsler R (2017) Formal system verification: state-of the-art and future trends. Springer
6. Drechsler R (2021) PolyAdd: polynomial formal verification of adder circuits. In: IEEE symposium on design and diagnostics of electronic circuits and systems, pp 99–104
7. Drechsler R (2021) Polynomial circuit verification using BDDs. arXiv:2104.03024
8. Drechsler R, Becker B, Ruppertz S (1997) The K*BMD: a verification data structure. IEEE Des Test Comput 14(2):51–59
9. Drechsler R, Dominik C (2021) Edge verification: ensuring correctness under resource constraints. In: Symposium on integrated circuits and system design
10. Drechsler R, Sieling D (2001) Binary decision diagrams in theory and practice. Int J Softw Tools Technol Transf 3(2):112–136
11. Goldberg EI, Prasad MR, Brayton RK (2001) Using SAT for combinational equivalence checking. In: Design, automation and test in Europe, pp 114–121
12. Höreth S, Drechsler R (1999) Formal verification of word-level specifications. In: Design, automation and test in Europe, pp 52–58
13. Kapur D, Subramaniam M (1998) Mechanical verification of adder circuits using rewrite rule laboratory. Formal Meth Syst Des 13(2):127–158
14. Kaufmann D, Biere A (2020) Nullstellensatz-proofs for multiplier verification. In: Computer algebra in scientific computing. Lecture notes in computer science, vol 12291. Springer, pp 368–389
15. Kaufmann D, Biere A, Kauers M (2019) Verifying large multipliers by combining SAT and computer algebra. In: International conference on formal methods in CAD, pp 28–36
16. Keim M, Drechsler R, Becker B, Martin M, Molitor P (2003) Polynomial formal verification of multipliers. Formal Meth Syst Des 22(1):39–58
17. Koren I (2001) Computer arithmetic algorithms. A. K. Peters, Ltd., 2nd edn
18. Mahzoon A, Drechsler R (2021) Late breaking results: polynomial formal verification of fast adders. In: Design automation conference
19. Mahzoon A, Drechsler R (2021) Polynomial formal verification of area-efficient and fast adders. In: Reed-Muller workshop
20. Mahzoon A, Große D, Drechsler R (2018) PolyCleaner: clean your polynomials before backward rewriting to verify million-gate multipliers. In: International conference on computer-aided design, pp 129:1–129:8
21. Mahzoon A, Große D, Drechsler R (2019) RevSCA: Using reverse engineering to bring light into backward rewriting for big and dirty multipliers. In: Design automation conference, pp 185:1–185:6
22. Mahzoon A, Große D, Drechsler R (2021) RevSCA-2.0: SCA-based formal verification of non-trivial multipliers using reverse engineering and local vanishing removal. IEEE Trans Comput-Aided Des Integr Circuits Syst
23. Mahzoon A, Große D, Scholl C, Drechsler R (2020) Towards formal verification of optimized and industrial multipliers. In: Design, automation and test in Europe, pp 544–549
24. Pavlenko E, Wedler M, Stoffel D, Kunz W, Wienand O, Karibaev E (2008) Modeling of custom-designed arithmetic components in ABL normalization. In: Forum on specification and design languages, pp 124–129
25. Somenzi F (2018) CUDD: CU decision diagram package release 2.7.0. available at https://github.com/ivmai/cudd

26. Stoffel D, Kunz W (2004) Equivalence checking of arithmetic circuits on the arithmetic bit level. IEEE Trans Comput Aided Des Circuits Syst 23(5):586–597
27. Temel M, Slobodová A, Hunt WA (2020) Automated and scalable verification of integer multipliers. In: Computer aided verification, pp 485–507
28. Vasudevan S, Viswanath V, Sumners RW, Abraham JA (2007) Automatic verification of arithmetic circuits in RTL using stepwise refinement of term rewriting systems. IEEE Trans Comput 56(10):1401–1414
29. Wegener I (2000) Branching Programs and Binary Decision Diagrams. SIAM
30. Wilhelm U, Ebel S, Weitzel A (2016) Functional safety of driver assistance systems and ISO 26262. In: Handbook of driver assistance systems: basic information, components and systems for active safety and comfort. Springer, pp 109–131
31. Wolf C (2015) Yosys open synthesis suit. http://www.clifford.at/yosys/
32. Zimmermann R (1997) Binary adder architectures for cell-based VLSI and their synthesis. PhD thesis, Swiss Federal Institute of Technology

Author Index

A
Aarthy, K. P., 201
Abikoye, Oluwakemi Christiana, 75
Agalya, P., 111
Ajantha Devi, V., 1
Akande, Hakeem Babalola, 75
Akande, Oluwatobi Noah, 75
Ali, Wajid, 307
Anci Manon Mary, A., 65
Aron, Rajni, 213
Ashvathe, S., 111
Ayoola, Joyce, 75

B
Babu, Kunda Suresh, 175
Babysruthi, S., 139
Balogun, Abdullateef, 75
Banerjee, Tathagat, 29
Bedi, Pradeep, 251
Bhandari, Sheetal U., 421
Bhatt, Shri. C. M., 201
Bisoy, Sukant Kishoro, 237
Bohara, Mohammed Husain, 151
Boni, Pritam Khan, 43

C
Charvi, K., 29
Chitrakala, S., 111, 201

D
Das, Siddharth, 387
Dhamale, Triveni D., 421
Dharavath, Ramesh, 17
Dheeraja, Karthikeyan, 267
Dhyani, Kshitij, 123
Drechsler, Rolf, 457

E
Etier, I., 65

G
Gaffar, Abdul, 349
Gandhi, Gaurav, 97
Gangadevi, E., 1
Gill, Simran, 319
Goel, Parth, 151
Goyal, S. B., 251
Gupta, Aakansha, 435

H
Hanuma Reddy, Pernamitta, 139
Harida, P. K., 387
Harpale, Varsha K., 421
Harsini, R., 111
Hussain, Ishtiaq, 307
Hussan, Muzamil, 57

I
Islam, Md. Rafiqul, 43

J
Jadav Badavath, Hemika, 17

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

N. Chaki et al. (eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Lecture Notes on Data Engineering and Communications Technologies 99, https://doi.org/10.1007/978-981-16-7182-1

Jan, Aiman, 57
Jena, P. M., 223
Joshi, Anand B., 349, 403

K
Kannan, Nithiyananthan, 65
Karahroudi, Nasrin Rabiei, 251
Karthikeyan, S., 29
Katarya, Rahul, 435
Kaur, Mandeep, 213
Kondaveeti, Hari Kishan, 377
Kowta, Alekya Sai Laxmi, 387
Kumar, Dhanesh, 403
Kumar, Jugnesh, 251
Kumar, Shailesh, 251
Kumar, S. Saravana, 361
Kumbhar, Harshawardhan S., 421

M
Mahzoon, Alireza, 457
Mallikharjuna Rao, K., 175
Mathews, Lincy, 447
Mehta, Vishesh, 89
Mishra, Jahnavi, 123
Mishra, Ram Krishn, 89

N
Narasimha Swamy, S., 159
Naved, Mohd, 1
Nayyar, Anand, 1
Nnaemeka, Enemuo Stephen, 75

P
Padma Priya, R., 267
Paladhi, Subhra, 123
Panigrahi, Prabin Kumar, 237
Parah, Shabir A., 57
Parida, S. K., 223
Patel, Priyang, 151
Pathwar, Prathmesh, 319
Phaneendra Kumar, B. L. N., 285
Prasanna Venkatesh, Vishnu Priya, 201
Priya, V., 387
Purohit, Gaurav, 159
PushpaLakshmi, R., 191

Q
Qureshi, G. J., 57

R
Raju, Kota Solomon, 159
Ramaguru, R., 295
Ramani, R. Geetha, 337
Raman, Sruti, 29
Ramesh, Kiruthya, 201
Rampriya, R. S., 139
Ritika, Thakur, 267

S
Sabarinathan, 139
Sajja, Ratna Sravya, 285
Sarkar, Sneha, 17
Satapathy, Santosh, 377
Seetha, Hari, 447
Sethumadhavan, M., 295
Shanthamalar, J. Jeslin, 337
Sharma, Amandeep, 29
Sharma, Sharad, 97
Shivanarayanamurthy, P., 159
Singh, Sonali, 403
Sivadanam, Yaswanth Lingam, 295
Sudhan, H. V. Madhu, 361
Suganya, R., 139
Sumaiya Thaseen, I., 123

T
Tripathi, Madhav, 151
Tripathy, B. K., 223
Tripathy, Hrudaya Kumar, 237

V
Venkatraman, Shruti Varsha, 387

W
Weingarten, Lennart, 457

Y
Yamini, Pitchuka, 139
Yang, Miin-Shen, 307

View publication stats