Lecture Notes in Electrical Engineering 1087

Ruchika Malhotra · L. Sumalatha · S. M. Warusia Yassin · Ripon Patgiri · Naresh Babu Muppalaneni *Editors*

High Performance Computing, Smart Devices and Networks

Select Proceedings of CHSN 2022

Springer

Lecture Notes in Electrical Engineering

Volume 1087

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

Ruchika Malhotra · L. Sumalatha ·
S. M. Warusia Yassin · Ripon Patgiri ·
Naresh Babu Muppalaneni
Editors

High Performance Computing, Smart Devices and Networks

Select Proceedings of CHSN 2022

Springer

Editors
Ruchika Malhotra
Delhi Technological University
New Delhi, India

S. M. Warusia Yassin
Universiti Teknikal Malaysia Melaka
Melaka, Malaysia

Naresh Babu Muppalaneni
Department of CSE
Indian Institute of Information Technology
Design and Manufacturing
Kurnool, Andhra Pradesh, India

L. Sumalatha
Jawaharlal Nehru Technological University
Kakinada
Kakinada, Andhra Pradesh, India

Ripon Patgiri
National Institute of Technology Silchar
Silchar, Assam, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-99-6689-9 ISBN 978-981-99-6690-5 (eBook)
https://doi.org/10.1007/978-981-99-6690-5

Dedicated to

Late Prof. Allam Appa Rao
Former Vice-Chancellor, JNTU Kakinada

Late Prof. Ch. Satyanarayana
Former Registrar, JNTU Kakinada

Program Committee

Prof. (Dr.) Viranjay M. Srivastava, University of KwaZulu-Natal, Durban
Prof. D. Vivekananda Reddy, SV University
Prof. Daniel Thalmann, École polytechnique fédérale de Lausanne, Switzerland
Prof. Edara Sreenivasa Reddy, Acharya Nagarjuna University
Prof. O. P. Vyas, Indian Institute of Information Technology, Allahabad
Prof. R. Bhramaramba, GITAM University
Prof. Ruchika Malhotra, Delhi Technological University
Dr. A. S. N. Chakravarthy, JNTU Vizianagaram
Dr. A. Srinagesh, RVR and JC College of Engineering
Dr. Aaditya Lochan Sharma, Sikkim Manipal Institute of Technology
Dr. Anchana P. Belmon, Rajadhani Institute of Engineering and Technology
Dr. Anjali Sharma, CSIR-National Physics Laboratory
Dr. Ankita Jain Bansal, Netaji Subhas University of Technology
Dr. Anuradha Chug, GGSIPU
Dr. Arijit Bhattacharya, Gour Mahavidyalaya
Dr. Asif Iqbal Hajamydeen, Management and Science University, Malaysia
Dr. B. S. Chandana, VIT AP
Dr. B. N. Jagadesh, VIT AP
Dr. Badal Soni, National Institute of Technology Silchar
Dr. Biswaraj Sen, Sikkim Manipal Institute of Technology
Dr. Bukhary Ikhwan Ismail, MIMOS Berhad-National Applied R&D Centre, Malaysia
Dr. Chukhu Chunka, National Institute of Technology Silchar
Dr. D. Ramesh, JNTUH University College of Engineering Jagtial
Dr. Dahlia Asyiqin Ahmad Zainaddin, German Malaysian Institute
Dr. Dalton Meitei Thounaojam, National Institute of Technology Silchar
Dr. E. Suresh Babu, National Institute of Technology, Warangal
Dr. Eugénia Moreira Bernardino, Polytechnic of Leiria, Portugal
Dr. Fahmi Arif, Institut Teknologi Nasional Bandung, Indonesia
Dr. G. Lakshmeeswari, GITAM University
Dr. G. Lavanya Devi, Andhra University

Dr. Ganeshkumar Pugalendhi, Anna University Regional Campus, Coimbatore
Dr. Ganta Chamundeswari, Sir C. R. Reddy College of Engineering, Eluru
Dr. Gurram Sunitha, Sree Vidyanikethan Engineering College
Dr. I. Gede Pasek Suta Wijaya, University of Mataram, Indonesia
Dr. J. Harikiran, VIT AP
Dr. J. Avanija, Sree Vidyanikethan Engineering College
Dr. Janmenjoy Nayak, Maharaja Sriram Chandra Bhanja Deo University, Baripada
Dr. Jayalaxmi G. Naragund, KLE Technological University
Dr. Juhi Jain, Amity School of Engineering and Technology
Dr. K. Vivekanandan, Puducherry Technological University
Dr. K. Reddy Madhavi, Sree Vidyanikethan Engineering College
Dr. Kabir Kharade, Shivaji University
Dr. Karthikeyan Subramanian, University of Technology and Applied Sciences, Oman
Dr. D. R. Kumar Raja, Reva University
Dr. L. Venkateswara Reddy, KG Reddy College of Engineering and Technology, Hyderabad
Dr. Laiphrakpam Dolendro Singh, National Institute of Technology Silchar
Dr. Latha Parthiban, Pondicherry University Community College
Dr. M. Brindha, National Institute of Technology, Tiruchirappalli
Dr. Malaya Dutta Borah, National Institute of Technology Silchar
Dr. Megha Ummat, Sri Guru Gobind Singh College of Commerce, University of Delhi
Dr. Mohammed Abdul Qadeer, Aligarh Muslim University
Dr. Mohammed Nasser Al-Mhiqani, Keele University
Dr. Mohammed Pasha, Muffakham Jah College of Engineering and Technology
Dr. Mohd Fairuz Iskandar Othman, Universiti Teknikal Malaysia Melaka
Dr. Mohd Faizal Abdollah, Universiti Teknikal Malaysia Melaka
Dr. Mohd Faizal Abdollah, Universiti Teknikal Malaysia Melaka
Dr. Mohd Zaki Mas'ud, Universiti Teknikal Malaysia Melaka
Dr. Naresh Babu Muppalaneni, Indian Institute of Information Technology Design and Manufacturing, Kurnool
Dr. Naveen Palanichamy, Multimedia University, Malaysia
Dr. Nazrulazhar Bahaman, Universiti Teknikal Malaysia Melaka
Dr. Ng Kok Why, Multimedia University, Malaysia
Dr. Nitikarn Nimsuk, Thammasat University (Rangsit Campus), Thailand
Dr. Noor Hisham Bin Kamis, Multimedia University (MMU), Malaysia
Dr. P. Sateesh, MVGR College of Engineering
Dr. Partha Pakray, National Institute of Technology Silchar
Dr. Pascal Lorenz, University of Haute Alsace, France
Dr. Prashant Giridhar Shambharkar, Delhi Technological University
Dr. Pratik Chattopadhyay, Indian Institute of Technology (BHU), Varanasi
Dr. R. Kanesaraj Ramasamy, Multimedia University, Malaysia
Dr. R. Murugan, National Institute of Technology Silchar
Dr. R. Rajeshwara Rao, JNTU Vizianagaram

Dr. Raihana Syahirah Abdullah, Universiti Teknikal Malaysia Melaka
Dr. Rajesh Prasad, American University of Nigeria
Dr. Rajib Kumar Jha, Indian Institute of Technology Patna
Dr. Ram Bilas Pachori, Indian Institute of Technology Indore
Dr. Ramanujam E., National Institute of Technology Silchar
Dr. Rashmi Saini, G. B. Pant Institute of Engineering and Technology
Dr. Ripon Patgiri, National Institute of Technology Silchar
Dr. Robiah Binti Yusof, Universiti Teknikal Malaysia Melaka
Dr. S. M. Warusia Yassin, Universiti Teknikal Malaysia Melaka
Dr. S. K. Chaya Devi, Vasavi College of Engineering
Dr. Sanjaya Kumar Panda, National Institute of Technology, Warangal
Dr. Sansanee Auephanwiriyakul, Chiang Mai University, Thailand
Dr. Santosh Singh Rathore, ABV-IIITM Gwalior
Dr. Sasikumar Gurumoorthy, Jerusalem College of Engineering
Dr. Sharifah Sakinah Syed Ahmad, Universiti Teknikal Malaysia Melaka
Dr. Sherali Zeadally, University of Kentucky
Dr. Siti Azirah Asmai, Universiti Teknikal Malaysia Melaka
Dr. Siti Rahayu Binti Selamat, Universiti Teknikal Malaysia Melaka
Dr. Sivasutha Thanjappan, Multimedia University, Malaysia
Dr. Sonika Dahiya, Delhi Technological University
Dr. Srinivasa Chakravarthi Lade, GITAM University
Dr. Surya Kameswari Uduga, Acharya Nagarjuna University
Dr. Surya S. R., College of Engineering Perumon, Kollam
Dr. Syed Muzamil Basha, REVA University
Dr. Syed Thouheed Ahmed, REVA University
Dr. T. G. Vasista, Pallavi College Engineering
Dr. Tanasanee Phienthrakul, Mahidol University, Thailand
Dr. Tomasz Rak, Rzeszow University of Technology, Poland
Dr. Uma N. Dulhare, Muffakham Jah College of Engineering and Technology
Dr. Umashankar Subramaniam, Prince Sultan University, Riyadh, Saudi Arabia
Dr. Veenu Mangat, Panjab University
Dr. Vishal Ramesh Satpute, Visvesvaraya National Institute of Technology, Nagpur
Dr. Vishal Saraswat, Bosch Engineering and Business Solutions
Dr. Wei-Chiang Hong, Asia Eastern University of Science and Technology, Taiwan
Dr. Yogan Jaya Kumar, Universiti Teknikal Malaysia Melaka
Dr. Zulkiflee Muslim, Universiti Teknikal Malaysia Melaka
Dr. Zuraida Abal Abas, Universiti Teknikal Malaysia Melaka
Mr. Ksh Robert Singh, Mizoram University
Mr. Nageshwar Nath Pandey, Siksha 'O' Anusandhan Deemed to be University
Mr. Rohit Pratap Singh, National Institute of Technology Silchar
Mr. Sahinur Rahman Laskar, National Institute of Technology Silchar
Mr. Sanjay Patidar, Delhi Technological University
Mr. Yonten Jamtsho, Royal University of Bhutan
Mr. Zakir Hussain, National Institute of Technology Silchar
Ms. Anjali Bansal, Delhi Technological University

Ms. Deepali Jain, National Institute of Technology Silchar
Ms. K. Bavani, Kalasalingam Academy of Research and Education
Ms. Priya Singh, Delhi Technological University
Ms. Sabuzima Nayak, National Institute of Technology Silchar
Ms. Shweta Meena, Delhi Technological University
Ms. Zeba Shamsi, National Institute of Technology Silchar

Keynote Speakers

Dr. S. M. Warusia Mohamed S. M. M. Yassin
Universiti Teknikal Malaysia Melaka

Dr. I. Gede Pasek Suta Wijaya
University of Mataram, Indonesia

Dr. Worapan Kusakunniran
Mahidol University, Bangkok, Thailand

Dr. Zuraida Abal Abas
Universiti Teknikal Malaysia Melaka

Foreword

The 3rd International Conference on Computer Vision, High Performance Computing, Smart Devices and Networks (CHSN-2022) is aimed to bring researchers together working in this area to share their knowledge and experience. In this conference, topics of contemporary interested would be discussed to provide a holistic vision on latest technologies for computer science and engineering. The scope includes data science, machine learning, computer vision, deep learning, artificial intelligence, artificial neural networks, mobile applications development and Internet of Things; conference participants are expected to gain relevant knowledge and better understanding of the applications of computer science in various fields.

CHSN-2022 would be both stimulating and informative with the active participation of galaxy of keynote speakers. We would like to thank all the authors who submitted the papers, because of which the conference became a story of success. We also would like to express our gratitude to the reviewers, for their contributions to enhance the quality of the papers. We are very grateful to the keynote speakers, reviewers, session chairs and committee members who selflessly contributed to the success of CHSN-2022. We are very thankful to Jawaharlal Nehru Technological University Kakinada, Kakinada, for providing the basic requirements to host the CHSN-2022.

Last but not least, we are thankful for the enormous support of publishing partner, i.e., Springer, for supporting us in every step of our journey toward success.

Kakinada, India

Prof. A. S. N. Chakravarthy
Convener, CHSN-2022

Contents

About the Editors

Dr. Ruchika Malhotra is the Head of the Department and Professor in the Department of Software Engineering at Delhi Technological University, Delhi, India. She served as Associate Dean of Industrial Research and Development at Delhi Technological University. She was awarded the prestigious Raman Fellowship for pursuing Post-doctoral research at Indiana University Purdue University Indianapolis USA. She received her master's and a doctorate in software engineering from the University School of Information Technology, Guru Gobind Singh Indraprastha University, Delhi, India. She has published over 200 research papers in international journals and conferences. Her research interests are in software testing, improving software quality, statistical and adaptive prediction models, software metrics, and the definition and validation of software metrics.

Dr. L. Sumalatha is the Professor of Computer Science and Engineering at the University College of Engineering Kakinada, a constituent college of JNTUK, Kakinada with over 23 years of experience. She served as Head of the Department, Director of Evaluation, and Director of Industry Institute Interaction and Placements. She graduated with her B.Tech. from Acharya Nagarjuna University, M.Tech., and Doctor of Philosophy from JNTUK in CSE discipline. Her research interests are Cyber Security, Natural Language Processing and Image Processing. She has supervised nine Ph.Ds. and currently supervising 11 Ph.D. scholars. She has published 65 research papers in International Journals and Conferences. She chaired sessions and has been a keynote speaker at International Conferences. She has one patent granted and two published.

Dr. S. M. Warusia Yassin is a senior lecturer at the Faculty of Information and Communications Technology, Universiti Teknikal Malaysia Melaka (UTeM). He completes his Bachelor's Degree in Computer Science (2007), Master of Science (2010), and Ph.D. (2015) at Universiti Putra Malaysia (UPM). He received recognition from MIMOS Berhad Malaysia as the best researcher in 2015. He bagged Silver Medal in Invention, Research and Innovation Exhibition (PRPI) 2014. He has executed various R&D, Consultancy projects funded by reputed organizations.

His research interests include computer security, data mining, cloud, and quantum computing.

Dr. Ripon Patgiri received his Bachelor's Degree from the Institution of Electronics and Telecommunication Engineers, New Delhi in 2009. He received his M.Tech. degree from the Indian Institute of Technology Guwahati in 2012. He received his Doctor of Philosophy from the National Institute of Technology Silchar in 2019. After M.Tech. degree, he joined as an Assistant Professor at the Department of Computer Science and Engineering, National Institute of Technology Silchar in 2013. He has published numerous papers in reputed journals, conferences, and books. Also, he is awarded several international patents. His research interests include distributed systems, file systems, Hadoop and MapReduce, big data, bloom filters, storage systems, and data-intensive computing. He is a senior member of IEEE. He is a member of EAI, and a lifetime member of ACCS, India.

Dr. Naresh Babu Muppalaneni working as an Associate Professor in the Department of Computer Science and Engineering at the Indian Institute of Information Technology Design and Manufacturing, Kurnool. He received his M.Tech. from Andhra University and Ph.D. from Acharya Nagarjuna University. He has published more than 30 papers in different International journals, Book Chapters, conference proceedings and edited research volumes. He has published five volumes in *Springer Briefs in Forensic and Medical Bioinformatics*. He is a Fellow of IETE, a life member of CSI, a member of ISCA and a Senior Member of IEEE. He is a recipient of the Best Teacher Award from JNTU Kakinada. He has completed research projects worth 2 crore rupees from DST and DRDO. He has organized six International Conferences and four Workshops. His research interests are artificial intelligence in biomedical engineering, human and machine interaction and applications of intelligent system techniques, social network analysis computational systems biology, bioinformatics and cryptography.

Analysis of Prevalence of Flat Foot in Primary School Children

Subodh Mor, Shikha N. Khera, and G. C. Maheshwari

Abstract Flat foot is a common health condition that prevails among children as well as adults. This paper analyses the relationship between age and gender with the prevalence of flat feet among primary school children. The results are validated on 424 primary school children (254 males and 170 females) between the age of 6 years to 10 years in Delhi, India. The foot imprinter plate as well as physical and photographic assessment was used to diagnose the presence of flat feet among primary school children. The number of children diagnosed with completely flat feet was 118, and children diagnosed with partial flat feet were 176. The results show out of every five children, three children were either completely or partially flat feet (69.3%). The results showed that there was a significant association between gender and flat foot. It was concluded that assessment of flat feet should be made available to children and parents at an early age to prevent the condition to be converted into a serious health problem and hindrance in various sports activities.

Keywords Flat feet · Partial flat feet · Primary school children · Sports activities

1 Introduction

The feet are an important factor in an individual's health since it plays a key role in mobility and posture and is responsible for the well-being and quality of life of an individual. Flat feet, also known as pes planus, is a common biomechanical problem faced by many individuals from an early age. It is present in infants as a part of the development of the foot and is gradually 15–20% resolved in adulthood [9, 21].

Flat feet can be defined as a postural deformity in which the foot does not have a normal arch where the entire sole of the foot is in near or complete contact with the ground [17]. These foot deformities may cause the development of various issues such as pain, fatigue, imbalance, and uneven distribution of planer pressure [16] and also

S. Mor (✉) · S. N. Khera · G. C. Maheshwari
Delhi Technological University, Bawana Road, Shahbad Daulatpur, Delhi 110042, India
e-mail: subodhmor@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_1

be a cause of various injuries. These changes can subsequently lead to compromise in well-being and quality of life while affecting the mobility, walking speed, and stance duration of an individual [12]. Flat feet can be divided into two categories, namely flexible (partial) or rigid. Flat feet can occur due to family history, structural and musculoskeletal abnormalities, neuromuscular issues, obesity, and urban lifestyle [1, 5, 6].

Flat feet may lead to hindrances in various sports activities like running, jumping, hopping, and various other coordination activities as excessive foot pressure can lead to stress on the legs, hips, and spine [4]. It can also be a cause of various injuries in sports persons. In previous studies, it was also found that anterior knee pain was high in male young defense recruits as compared to females. It was also shown that there was a significant correlation between flat feet and anterior knee pain [15].

There have been various studies that evaluate the relationship between personal characteristics such as age, gender, and BMI with flat feet [2, 3, 7, 10, 11, 13, 18, 23]. In this work, we analyze the relationship between age, gender with flat feet. The results are validated on 424 primary school children in Delhi, India, between 6 years and 10 years of age group. The data has been collected as a part of an awareness and educational campaign in schools. The early detection of flat feet can prevent the issue from converting into a serious medical condition among children. It will also guide parents and children in the selection of appropriate sports as a future career.

The rest of the paper is organized as follows: Section 2 presents the related work, and the research background is summarized in Sect. 3. In Sect. 4, results are presented and discussed. Section 5 presents the conclusions of the work.

2 Related Work

In this section, the studies that are closely related to our work are summarized. Table 1 presents the summary of the size of participants, age, research variables, and statistical tests used in the studies. Eluwa et al. examined 1000 students of Akwa Ibom State aged between 20 and 30 and concluded that there were a higher number of females with flat feet as compared to males [10]. Chang et al. analyzed 2083 school children and concluded that males had twice more flat feet as compared to females. Further, overweight children had more flat feet as compared to the ones with normal weight [7]. Ezema et al. conducted a study with 474 primary school children [11]. They found that there was a significant association between flat feet and age and also between obesity and flat feet. In [11], it was further concluded that males were twice likely to be diagnosed with flat feet than females. Pashmdarfard et al. assessed the prevalence of flat feet among 1700 primary school children aged between 7 and 12 years [18]. There was a significant relationship found between weight and flat feet. Bhoir et al. conducted a study with students aged between 18 and 25 years and found that there was no correlation between gender and BMI with arch index [3]. Al-shenqiti et al. examined school children between 6 to 12 years and concluded

Table 1 Summary of related work

Authors	Study size	Age	Variables used	Statistical techniques
Eluwa et al. [10]	1000	20–30	Gender	Mean and standard deviation
Chang et al. [7]	2083	7–12	Age, gender, and BMI	Multivariate analysis
Ezema et al. [11]	474	6–10	Age, gender, and BMI	Chi-square test
Pashmdarfard et al. [18]	1700	7–12	Age, height, weight	Ordinal regression
Bhoir et al. [3]	80	18–25	Gender and BMI	Correlation analysis
Al-Shenqiti et al. [2]	563	6–12	Age, weight, height, and BMI	Correlation analysis

that there was no correlation between weight and BMI with flat feet, and there was a significant correlation between age, height, and gender with flat feet [2].

In this work, data from 424 primary school children in Delhi, India, is collected, and the association between age, gender, and prevalence of flat feet and partial flat feet is assessed.

3 Research Methods

In this section, the description of variables, study participants, and statistical techniques is presented.

3.1 Research Design and Variables

The study compiled a list of all primary school children between 6 and 10 years. The data for the children with completely flat feet or low arc (partial flat feet) was collected. Physical assessment, as well as discussions, was carried out so that the children with other foot deformities such as high arc or other deformities could be excluded from the study.

The flat foot issues that are of little concern at an early stage of children's upbringing may become a serious issue at a later stage when the child becomes older particularly if it is associated with mobility and quality of life issues. Further, this information can be used effectively by the children and their parents for the pursuance of sports now and in the future.

The children were first physically examined. Then, photographic analysis was obtained for further assessment (Fig. 1). A foot imprinter plate was used to analyze whether the foot was flat, partially flat (low arched), or normal. The portable foot

Fig. 1 Foot assessment

imprinter plate had 2704 calibrated sensors, 5 Hz frequency, 5% accuracy, 100% digital calibration, and −10 to +45 °C temperature. Simultaneously, a check on weight distribution on both feet was made. Finally, the obtained data was analyzed by an expert team, and based on the analysis, a recommendation folder consisting of suggested exercises and further details was given to the participant child.

3.2 Study Participants and Flat Foot Diagnosis

A total of 424 primary school children with age between 6 years and 10 years were included in the study. The data was collected from five classes (I-V). Ethical permission was obtained for the inclusion of the data. The parents were fully informed about the assessment procedure. The main aim of the assessment of primary school children was to educate children and their parents and to communicate basic information about his/her feet so that they have a better understanding of their children's needs, and better care can be provided to them. After the examination, the expert team summarized the characteristics of children including age, gender, and type of feet and prescribed appropriate exercises and measures that the child needs to learn and follow for improving the foot mechanics.

The study considered two personal characteristics, namely age, gender, and feet type with three categories (1) normal feet (2) partial flat feet, and (3) flat feet. Table 2 presents the summary of data from 424 participants with respective counts and percentages. There are 120, 82 78, 68, and 76 primary school children aged 6, 7,

Table 2 Characteristics of participants

Variable	Value	Count (percentage)
Age	6	120 (28.3%)
	7	82 (19.3%)
	8	78 (18.39%)
	9	68 (16.03%)
	10	76 (17.92%)
Gender	Male	254 (60%)
	Female	170 (40%)
Feet type	Normal feet	130 (30.66%)
	Partial flat feet	176 (41.5%)
	Flat feet	118 (27.83%)

8, 9, and 10, respectively. The percentage of female children is 40% and male children 60%. There were 30.66% of children with normal feet, 41.5% of children with partial flat feet, and 27.83% of children with flat feet.

3.3 Statistical Analysis

In this work, chi-square, a statistical test, is used to find the association between the personal characteristics (age and gender) and the presence of flat feet or partial flat feet of a child. The chi-square test is a non-parametric test that is used to check whether there exists a significant difference between expected frequency and observed frequency [22]. In this test, data samples are drawn randomly from the population. This test works on categorical data or we can say that this test is used when the data is ordinal or nominal. The level of significance was 0.01. The hypothesis of the work is that there is an association between age, gender, and prevalence of flat feet or partial flat feet.

4 Analysis of Results

In this section, presented here is the analysis and discussion of the results of the statistical analysis carried out in this work. A total of 424 primary school children were included in the analysis. The association between age and gender with the prevalence of flat foot was analyzed using the chi-square test as given in Sect. 3.4.

4.1 Association of Personal Characteristics with Flat Foot

Table 3 presents the summary of the count of the presence of flat feet in school children with respect to age, between 6 and 10 years. The percentage of children with flat feet and partial flat feet at various age levels is depicted in the bar chart shown in Fig. 2. The prevalence of flat feet was found to be highest in 10-year-old children (36.84%) and lowest in 7-year-old children (21.95%). The prevalence of partial flat foot was found in about 43% of children aged 6, 7, 8, 10, and lowest in 9-year-old children (30.88%). There was no significant association found between age and flat foot at a 0.01 significance level (chi-square statistic: 11.134, p-value: 0.194).

Table 4 presents the prevalence of flat feet in primary school children with respect to gender. Figure 3 depicts the percentage of males and females with normal, partial flat, and flat feet. There are 14% of females and 37% of females and males with flat feet. There are 52% of females and 34% of males with partially flat feet. The association between the gender and prevalence of flat foot is found to be significant (chi-square statistics: 27.974, *p*-value: 0.000) at a 0.01 significance level.

Table 5 shows the prevalence of flat feet among male and female primary school children with respect to various levels of age. It can be seen flat feet are highest in 6-year-old and 10-year-old male children.

Table 3 Prevalence of flat feet with respect to age

	6 years	7 years	8 years	9 years	10 years
Normal feet	35	28	26	26	15
Partial flat feet	52	36	34	21	33
Flat feet	33	18	18	21	28

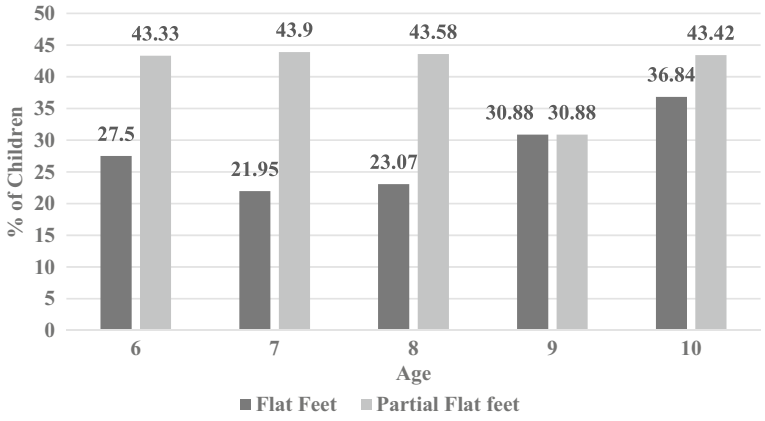


Fig. 2 Percentage of primary school children with flat and partial flat foot

Table 4 Prevalence of flat foot with respect to gender

	Male	Female
Normal feet	73	57
Partial flat feet	87	89
Flat feet	94	24

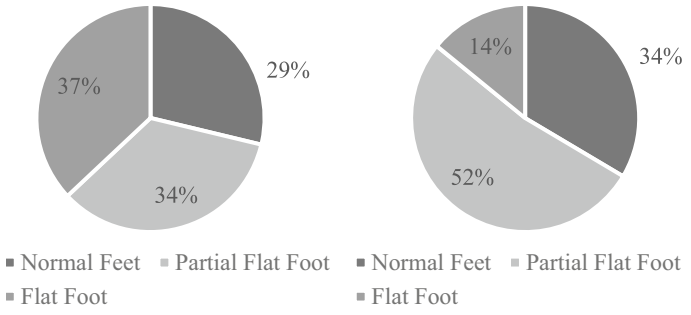


Fig. 3 Percentage of **a** males **b** females with normal, partial flat, and flat foot

Table 5 Prevalence of flat foot with respect to age and gender

Age	6 years		7 years		8 years		9 years		10 years	
Gender	M	F	M	F	M	F	M	F	M	F
Normal feet	16	19	19	9	12	14	18	8	8	7
Partial flat feet	26	26	13	23	15	19	14	7	19	14
Flat feet	24	9	16	2	16	2	16	5	22	6

4.2 Discussion

It can be seen that there were 21.95–36.84% of primary school children with flat feet. This implies that at least one in every five primary school children had a flat foot. There were 43% of primary school children with partial flat feet which means that more than two in every five primary school children have a partial flat foot. Thus, more than 60% of primary school children have either flat feet or partial flat feet. In other words, three in every five primary school children have either a flat foot or a partial flat foot.

The study reveals that with age flat foot increases and 10-year-old children have the highest number of flat feet. The presence of partial flat foot was consistent among all age children except 9-year-old children. This implies corrective actions and changes should to taken at an early age so that the children with partial flat feet do not develop complete flat feet with age. However, the results show that there is no significant association between age and the prevalence of flat foot or partial flat foot. Chang et al. [7] and Ezema et al. [11] observed that the prevalence of flat foot among

primary school children decreased with age. Thus, further study is required to prove this hypothesis.

The findings of this study demonstrate that male primary school children with flat feet were more than twice as compared to their female counterparts. But, the prevalence of partial flat feet in female children was equal as compared to male children which means that the number of female children with flat feet may increase with growing age. There was a significant association detected between gender and flat or partial flat feet among children. The results are consistent with the other studies. Ezema et al. examined 474 primary school children aged between 6 to 10 years and found that about twice male children with flat feet were more as compared to females [11]. Chang et al. also observed that there were twice more female children with flat feet as compared to male children [7]. Pashmdarfard et al. found that there was no significant effect of gender on the prevalence of flat foot among 7 to 12 years aged children [18]. In [10], the results show that the female occurrences of flat feet were more as compared to male children. This may be because the study examined 20–30 years old participants.

4.3 Recommendations

The results of this study show that there is 27.83% and 41.5% of primary school children aged between 6 and 10 years with flat feet and partial flat feet, respectively. Thus, 294 out of 424 (69.3%) primary school children have either flat foot or partial flat foot which is a matter of concern. The initial sign of a flat foot may become an issue of concern at a later age as the child will get older. It may lead to a serious health issue or may affect the quality of life of the child by affecting physical mobility (walking, running, and balance), causing pain and other complications [8, 19].

The results by Martin et al. showed the prevalence of flat feet led to a decrease in quality of life and an increase in disability and pain in the feet [12]. The results of the study conducted by López-López et al. of Spanish patients showed that individuals with foot pathologies had worse quality of life as compared with individuals with the normal foot. The study emphasized improving foot health.

Given the fact that footwork, speed, agility, and balance are key components of many sports (basketball, tennis, and running), the prevalence of flat feet among children may affect the career of a child in sports in the future [4]. A flat foot increases the stress and pressure on the inside of the foot and ankles. The presence of flat feet may increase the risk of injuries in athletes during training sessions and sports competitions [14]. When the physical activity increases and the overwork is done by an individual with a flat foot, it can result in foot pain, muscle spasms, calf fatigue, hip or back pain, and so on. This may hinder the sports activities of an individual and their performance in various sports. Many times, an individual is not aware of the presence of a flat foot until some issue arises while playing a particular sport. Sharma and Upadhyaya emphasized that a flat foot affects the running performance of an athlete due to a decrease in ankle muscle strength [20].

The solution for the flat foot is relatively simple at a younger age. This study is a part of an awareness program conducted in schools with primary school children at an early age so that the problem of foot deformities and other issues can be addressed at an early age. The children and their parents can be educated about foot-related issues and simple exercises and techniques for the protection of the foot structure at a younger age. Thus, the flat foot issue is significantly critical and must be prevented and controlled at an early age before it is converted into a serious medical condition at an older age. Medical camps and awareness programs in schools can be very useful for preventing and controlling foot deformities in children at a very early age. Further, an early assessment of flat feet can provide the basis for parents to select appropriate sports for their child.

5 Conclusion

In this work, we analyzed the association between age and gender with the prevalence of flat or partially flat feet. The results are validated on the dataset collected from 424 primary school children aged between 6 and 10 years. The results showed.

that there are 41.5% with partial flat feet and 27.83% with completely flat feet. Hence, in Indian children aged between 6 and 10 years the prevalence of flat feet or partial flat feet is high, almost 3 in every five children. There was no association found between age and the presence of flat feet. However, there was a significant association found between gender and the prevalence of flat feet. Hence, male children require close monitoring and are at higher risk of flat feet at an early age and females have partial flat feet at an early age, and if precautions and minor corrections are taken, it will prevent converting the partial flat feet into complete flat feet at an older age.

The study suggests an early assessment of the prevalence of flat feet in children so that preventive or corrective measures can be taken at an initial age of a child. This will not only improve the well-being and quality of life of an individual but also will help in making decisions about participation in appropriate sports.

In the future, further analysis of children aged between 10 and 15 years will be made to further gain insight into the prevalence of flat feet among these children.

References

1. Abdel Fattah MM, Hassanin MM, Felembane FA, Nassaane MT (2006) Flat foot among Saudi Arabian army recruits: prevalence and risk factors. EMHJ-Eastern Med Heal J 12(1–2):211–217
2. Al-Shenqiti AM, Eweda RS, Emara HA, Khaled OA, Ibrahim SR, Ahmed MS, Mohamed MI, El-gohary TM (2020) Prevalence of flat foot in Saudi Arabian primary school children in relation to age, gender, height and obesity: a cross-sectional study. Med Sci 24(101):235–242
3. Bhoir T, Anap D, Diwate A (2014) Prevalence of flat foot among 18–25 years old physiotherapy students: cross sectional study. Indian J Basic Appl Med Res 3(4):272–278

4. Bhosale N, Nandala P (2021) Prevalance of flexible flat foot in athletes. Kesari Mahratta Trust 1(1)
5. Cappello T, Song KM (1998) Determining treatment of flatfeet in children. Curr Opin Pediatr 10(1):77–81
6. Cappello T, Song KM (1998) Foot deformities in infants and children. Pediatr Clin North Am 33:14411–14427
7. Chang JH, Wang SH, Kuo CL, Shen HC, Hong YW, Lin LC (2010) Prevalence of flexible flatfoot in Taiwanese school-aged children in relation to obesity, gender, and age. Eur J Pediatr 169(4):447–452
8. Dabholkar T, Agarwal A (2020) Quality of life in adult population with flat feet. Int J Heal Sci Res 10(8)
9. Dunn JE, Link CL, Felson DT, Crincoli MG, Keysor JJ, McKinlay JB (2004) Prevalence of foot and ankle conditions in a multiethnic community sample of older adults. Am J Epidemiol 159(5):491–498
10. Eluwa MA, Omini RB, Kpela T, Ekanem TB, Akpantah AO (2009) The incidence of pes planus amongst Akwa Ibom State students in the University of Calabar. Int J Forensic Sci 3(2):1–5
11. Ezema CI, Abaraogu UO, Okafor GO (2014) Flat foot and associated factors among primary school children: a cross-sectional study. Hong Kong Physiother J 32(1):13–20
12. Gonzalez-Martin C, Pita-Fernandez S, Pertega-Diaz S (2018) Quality of life and functionality in patients with flatfoot. Update in management of foot and ankle disorders. London: Intech Open, 73–90
13. Gross KD, Felson DT, Niu J, Hunter DJ, Guermazi A, Roemer FW, Dufour AB, Gensure RH, Hannan MT (2011) Association of flat feet with knee pain and cartilage damage in older adults. Arthritis Care Res 63(7):937–944
14. Korkmaz MF, Acak M, Duz S (2020) The effect of sports shoes on flat foot. Pedagogy Phys Cult Sports 24(2):64–71
15. Lakstein D, Fridman T, Ziv YB, Kosashvili Y (2010) Prevalence of anterior knee pain and pes planus in Israel defense force recruits. Mil Med 175(11):855–857
16. López-López D, Pérez-Ríos M, Ruano-Ravina A, Losa-Iglesias ME, Becerro-de-Bengoa-Vallejo R, Romero-Morales C, Calvo-Lobo C, Navarro-Flores E (2021) Impact of quality of life related to foot problems: a case–control study. Scient Reports 11(1):1–6
17. Lovett HW, Dane J (1896) The affections of the arch of the foot commonly classified as flat-foot. J Bone Joint Surg 1(1):78–92
18. Pashmdarfard M, Amini M, Sanei SH, Shirdareh ML, Marzbali KH, Namazi NG, Ostadzadeh A (2019) Prevalence of flat foot in primary school students aged 7–12 years, in Zanjan City, Iran. J Modern Rehab 13(4):207–214
19. Pita-Fernandez S, Gonzalez-Martin C, Alonso-Tajes F, Seoane-Pillado T, Pertega-Diaz S, Perez-Garcia S, Seijo-Bestilleiro R, Balboa-Barreiro V (2017) Flat foot in a random population and its impact on quality of life and functionality. JCDR 11(4):LC22
20. Sharma J, Upadhyaya P (2016) Effect of flat foot on the running ability of an athlete. Indian J Orthopaedics Surg 2(1):119–123
21. Staheli LT (1992) Fundamentals of pediatric orthopedics. Raven Press, New York
22. Weaver KF, Morales VC, Dunn SL, Godde K, Weaver PF (2017) An introduction to statistical analysis in research: with applications in the biological and life sciences. Wiley
23. Xu L, Gu H, Zhang Y, Sun T, Yu J (2022) Risk factors of flatfoot in children: a systematic review and meta-analysis. Int J Environ Res Public Health 19(14):8247

Evolutionary, Protein–Protein Interaction (PPI), and Domain–Domain Analyses in Huntington's Disease

Sai Gopala Swamy Gadde, Kudipudi Pravallika, and Kudipudi Srinivas

Abstract Mutations play a vital role in causing human neurodegenerative diseases such as Huntington's, Alzheimer's, and Parkinson's. A DNA section known as a CAG trinucleotide repeat is involved in the Huntingtin (HTT) mutation that leads to Huntington's disease (HD). Evolutionary, protein–protein interaction (PPI), and domain–domain interaction analysis provide new perceptions into neurodegenerative disorders and enable the prediction of the new causing proteins and their functionality associated with neurodegenerative diseases. The main objective of this study is to identify the top and associated proteins for HTT. This study is divided into three phases. In the first phase implemented the evolutionary analysis to identify the similarities at the cellular and molecular levels of Huntington's disease-causing proteins. In the second phase applied the PPI analysis to build an HTT network which is used to identify existing associations and predict the new associations for this HTT. Finally, domain–domain analysis has been used to analyze the domains of similar proteins with newly identified associated proteins of HTT. This study reveals that brain-derived neurotrophic factor (BDNF), α-Adaptin, and Butyrylcholinesterase (BChE) are the new proteins directly and indirectly associated with Huntington's disease. Also, this integrated analysis finally leads to personalized and precision medicine for other chronic diseases.

Keywords Mutations · Evolutionary analysis · Huntington's disease · Protein–protein interaction analysis · α-Adaptin · Domain-domain analysis · HTT · BDNF · BChE

S. G. S. Gadde · K. Srinivas (✉)
VR Siddhartha Engineering College, Vijayawada, India
e-mail: vrdrks@gmail.com

K. Pravallika
Sir C. R. Reddy Engineering College, Eluru, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_2

1 Introduction

Several hundred neurodegenerative disorders are thought to exist at this time, and many of them seem to overlap both clinically and pathologically, making it difficult to classify them practically. The fact that different combinations of lesions might result in distinct clinical presentations in disorders like multisystem atrophy, which affects many brain regions, further complicates the situation. Additionally, multiple parts of the brain may be affected by the same neurodegenerative process, particularly in the beginning, making a particular illness look very distinct from a symptomatic perspective. Despite these challenges, the primary clinical characteristic, or the topography of the prominent lesion—or sometimes a combination of both—remains the basis for categorizing neurodegenerative diseases [1].

CAG repeats more than 36 in at least one Huntingtin (HTT) allele seems to increase the likelihood of developing Huntington's disease (HD), a degenerative neurological disorder (HD). For brain development and embryogenesis, HTT gene functioning is vital. Memory loss, worsening paralysis, and premature grave all seem to be indications of neurotoxicity triggered by HTT proteins that have an N-terminal stretch of Q repeats containing more than 36 glutamine residues. This condition has been identified as PolyQ syndrome and is spurred on by highQ-Htt proteins [2]. The disease strikes people after 40 repetitions, and the age at which the condition first manifests is negatively correlated with the number of repetitions. The chance of developing Huntington's disease is higher in older adults above the age of 60 [3].

The expanded-Q Htt (mHtt) protein's N-terminal region, which is generated by proteolytic cleavage, is primarily responsible for the expanded-Q Htt protein's toxic gain of function. Increased quantities of polyQ aggregates cause cellular stress, which would in turn triggers a persistent unfolded protein response (UPR) and eventually neuronal death. The complexity of Huntington's disease can be linked to mHtt's propensity to interact improperly with numerous proteins that ordinarily either interact or do not interact with the wild-type (wt) Htt protein. This is made worse by the Htt protein's widespread subcellular distribution, where it is thought to interact in several signaling cascades and/or associate with many other protein partners during the course of its normal function [3]. Vesicle transport, transcriptional activity, mitochondrial activities, synaptic transmission, and more recently chromatin condensation are a few of the crucial molecular and cellular processes that are impacted [4].

Huntington's protein contains approximately 3150 amino acids (which are the fundamental components of all proteins). Longer chains are associated with worse symptoms before the disease the mutant protein in Huntington's disease patients contains an overly long string of a single amino acid repeat. The impact of other, neighboring amino acids in this massive protein, particularly biochemical modifications to those amino acids, has finally returned under researchers' observation [5].

Not just a few proteins seem responsible for the progression of Huntington's disease (HD). It connects with several directly and indirectly contextualizing HTT gene proteins. It is feasible that a modest increase in an unidentified protein caused certain cases of Huntington's disease [5]. Understanding the anonymous protein that greatly affects the disease makes it easier to understand the syndrome. Protein–protein interactions (PPI) can be used to identify these disease-causing proteins [6].

PPIs are a critical step in the understanding of how proteins operate in the advancement of the cell cycle, DNA replication, and signaling. More PPI data have been gathered because of the advent of high-throughput biological technologies, such as mass spectrometry, tandem affinity purification tagging (TAP), and protein chip technology. Multiple built-in databases have been used to store PPI datasets, including Gene cards, Uniport, and the National Center for Biotechnology Information (NCBI). Experimentation requires a lot of time and effort, though. Only a small part of the overall PPI network's PPIs is validated using these techniques. Applying the string tool, an online tool that discovers relationships between proteins and builds networks based on identified proteins has been taken into consideration.

The diseases like Alzheimer's, Parkinson's, Huntington's, and many more are related to many factors like age, gender, genetics, etc. To get a clear idea of the causes of these diseases, we need to have a new and advanced approach to this study which includes protein level analysis [7].

The term phylogeny is a combination of two Greek words. A phylogenetic tree helps to find similar proteins among the list of proteins that are considered, it gives an idea to find the change from the evolution of the species from the past years [8]. The PPI analysis gives us a clear idea of how the proteins interact among themselves, directly or indirectly. A domain–domain analysis is used to discover a specific function or interaction that contributes to a protein's overall role which gives a conformational idea of the interaction between the proteins that cause the disease by identifying the domains of every protein that has an involvement with the disease-causing protein [9].

The main objective of this study is to identify the new proteins that cause Huntington's disease by influencing the HTT protein. In this study, the phylogenetic tree is constructed to identify similar proteins and construct the PPI network for the identified similar proteins to know the relation among the proteins. Applying the domain–domain analysis gives us a confirmational idea that the proteins are the cause of the disease by comparing the functionalities with the HTT.

2 Literature Survey

Liu et al. [9] discussed the overview of the prediction protocol, graph-pheno performance evaluation, the identification of genes linked to abnormalities in the mitochondrial respiratory chain, and the identification of predicted and known phenotype-associated genes with related biological properties. Using graph-pheno, they inferred

995 potential DAMRC, which connected genes. They anticipated the DAMRC-associated genes, which are connected to the respiratory electron transport chain, mitochondrial protein complex, and inner membrane of the mitochondria. To determine the analysis of the protein and forecast the conclusion, they built a PPI network.

Calabrese et al. [10] discussed the analysis of the six proteins that are prone to aggregation with the help of the PPI network. By building a network, they represented the proteins that exhibit interaction with the PPI evidence on BioGRID. They looked at Huntington's, Alzheimer's, and Parkinson's disease and using a network and the primary proteins that cause the disorders discovered connections between all of the diseases.

Proskura et al. [11] discussed about normal HTT interactions with a specific collection of proteins help to preserve synaptic transmission effectiveness over time, especially in the hippocampus CA1 field. This paper's primary goal was to conceptually assess the relevance of HTT to alterations in synaptic plasticity by fusing the available experimental data. They hypothesized that polyQ-HTT may compete with proteins that control cytoskeleton remodeling and postsynaptic density. Their suppositional results and closures provide a way that helps in developing novel techniques for the detection, prevention, and treatment of the associated pathologies as well as for analytically confirming incongruous data in a directed manner.

Rao et al. [12] discussed about the diffusion algorithm that seeks to identify the nodes most pertinent to the initial set by using a bundle of nodes and the whole interaction network. Each node in the set receives heat from diffusion, which then allows the heat to diffuse through the connecting edges to other nodes. The list of nodes is then ordered according to the heat that is gathered. An isolated node will often have a low rank, whereas a higher ranking was given to the nodes with many connections. The collection of selected nodes serves as the heat source for diffusion, with initial heat being applied to all nodes equally. After pre-processing, pruned proteins are used as the input for the diffusion procedure. To determine network propagation, this diffusion approach transmits the full network of nodes to a Web-based REST API. The top proteins with the same sequence are found by using the sequence analyzer, which was employed to identify the dataset's most frequently occurring sequence.

Rao et al. [13] discussed about PPI databases, sorting PPI detection techniques, methods for PPI comparison, PPI network computational analysis, and PPI networks' function in proteomics. Their main focus is on the PPIs as they relate to the proteins listed. The relationships between proteins change throughout time, one protein may have several different jobs, and rarely, two proteins having distinct functions will link. This method overcomes all of the pre-existing methods by having a high accuracy rate.

Heinz et al. [14] discussed about the tight regulation of protein expression from particular mRNA molecules based on the cell type, as indicated. They discussed about MID1, which stimulates the translation of target mRNAs by proteins. This study examines the domains and genes of the MID1 protein and how it interacts with Huntington's disease structure.

Wanker et al. [15] discussed about the altered mutant HTT PPIs, which are a hallmark of Huntington's disease pathobiology. They talked about the pathobiological and mechanism-based processes that lead to neuronal malfunction, toxicity, and the formation of phenotypes in the minds of those suffering from Huntington's disease. Their findings indicate that striatal BDNF restoration efforts may have therapeutic benefits on HD since they show that diminished striatal BDNF is a critical factor in HD aetiology. They discovered, at last, that YAC128 mice also had lower striatal levels of mature BDNF.

Prasanthi et al. [16] discussed about the ID3 decision tree classification algorithm. Their study examined the ID3 a robust decision tree algorithm and the applications thereof across a wide range of manufacturing industries, including fitness, medicine, pedagogy, and systematization. Their performance of ID3 has shown good performance across all the domains. They also talked about the use of ID3 in treating diabetes. Each cluster's decision tree generates a set of adaption rules that can be used to identify patients with diabetes.

Fodale et al. [17] discussed about comparing the HTT protein, which is the protein that causes Huntington's disease, and the five standard HTT proteins N573Q22, N573Q45, N573Q72, FLQ17, and FLQ46, and one may determine the expression of mutant and full-length HTT in Huntington's disease. By contrasting all the proteins along with the HTT protein, primarily deals with, and concentrates on the sickness. This has made them evident how similar the proteins are to the HTT.

Rao et al. [18] discussed about phylogenetic tree analysis of BChE to show the relationships between butyrylcholinesterase (BChE) and other proteins. They used PPI analysis on proteins that are related to one another by building a network to do domain analysis and validate the domain analysis. The authors revealed that there is a connection between BDNF and BChE and demonstrated how BChE contributes to the selective gradation of transient and aberrant proteins in stressed cells.

Sridhar et al. [19] discussed how BChE and type 2 diabetes are related. They considered 35 sequences from the NCBI database for the study, and their relationships were established by creating a phylogenetic tree using the distance technique and the maximum parsimony approach (MP).

3 Methodology

This framework is divided into three phases, such as evolutionary analysis, PPI analysis, and domain–domain analysis. In evolutionary analysis, Huntington's disease-causing proteins collected from the Uni-Port (https://www.uniprot.org/), NCBI (https://www.ncbi.nlm.nih.gov/), and Gene-card (https://www.genecards.org/). The other proteins that influence Huntington's disease are identified after applying evolutionary analysis, PPI analysis, and domain–domain analysis. Figure 1 describes the methodology of this study.

Steps Used:

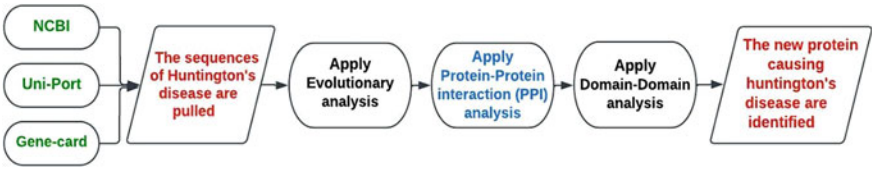


Fig. 1 Block diagram of evolutionary, PPI, and domain–domain analyses in Huntington's disease

Step 1: By classifying the Homo sapiens species, use various proteins from Uni-Port (https://www.uniprot.org/), NCBI (https://www.ncbi.nlm.nih.gov/), and Gene-card (https://www.genecards.org/) as input.

Step 2: Using the FAST-All (FASTA), create a phylogenetic tree out of the proteins, you acquired from NCBI, Gene-cards, and Uni-Port using the cluster omega (https://www.ebi.ac.uk/Tools/msa/clustalo/).

Step 3: Utilize evolutionary analysis to determine the most prevalent proteins from the phylogenetic tree that was created.

Step 4: Create a PPI network by using string (https://stringdb.org/cgi/input?sessionId=bY6X0Yf5DpXN&input_page_show_search=on) and read the most prevalent proteins that the phylogenetic tree has revealed.

Step 5: An isolated node will typically have a low rank, while a very connected node would often hold a high position.

Step 6: The protein interaction network is then created by removing the low-ranking nodes.

Step 7: Determine the domains by using Interpro (https://www.ebi.ac.uk/interpro/) and compare them.

Step 8: A sequence analyzer is used to examine the resulting protein dataset in further detail.

Step 9: The proteins with the highest relation with HTT are the results.

3.1 Data Collection

The sequences are pulled from Uni-Port, NCBI, and Gene-card being connected directly or indirectly to the process producing Huntington's disease to analyze individuals or identify differences. A total of 28 sequences some of which were evaluated, and for the study, domain units with known functions were employed. The following proteins have been identified as being associated with Huntington's disease: Huntingtin (P42858), nucleolar protein 14 (P78316), transcription elongation regulator 1 (O14776), SH3 domain-binding protein (P78314), Huntingtin-associated

protein 1 (P54257), histone-lysine N-methyltransferase SETDB1 (Q15047), Cdc42-interacting protein (Q15642), Capucin (A6NDD5), Antithrombin-III (P01008), Huntingtin-interacting protein 1 (O00291), histone-lysine N-methyltransferase SETD2 (Q9BYW2), fibroblast growth factor receptor 3 (P22607), prothrombin (P00734), Alpha-1-antitrypsin (P01009), major prion protein (P04156), palmitoyl-transferase ZDHHC17 (Q8IUH5), Caspase recruitment domain-containing protein (Q5EG05), Pre-mRNA-processing factor 40 homolog B (Q6NWY9), Heparin cofactor 2 (P05546), Complexin-2 (Q6PUV4), Alpha-adducin (P35611), RING finger protein 4 (P78317), Junctophilin-3 (Q8WXH2), Endophilin 3(Q99963), α-Adaptin (O94973), Huntingtin-associated protein 40(P236 10), brain-derived neurotrophic factor (P23560), and butyrylcholine esterase (P06276).

3.2 Evolutionary Analysis

FASTA (FAST-All) of 28 sequences are placed into the cluster omega tool to investigate how the program for multiple-sequence alignment can accurately and efficiently align 28 sequences. Physiologically relevant multiple-sequence alignments connect dissimilar sequences. Phylograms and cladograms are useful for identifying evolutionary connections. Create a phylogenetic tree based on the similarities between the most widespread parameter. The length between the proteins indicates how similar they are to one another [20–22]. Phylogenetic tree helps to identify the most recent ancestors to know their protein evolution.

3.3 Protein–Protein Interaction (PPI) Analysis

While many proteins in the network in PPI have fascinating connections to one another and other proteins, the PPI does not interact at all [23]. Understanding which protein causes Huntington's disease is made possible by the PPI analysis of the condition. The top comparable proteins that are generated from the evolutionary study have been employed by the string tool to discover the PPI [24]. The PPI network has been created after taking into account the proteins, namely butyrylcholine esterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein1 (HAP1), brain-derived neurotrophic factor (BDNF), and Junctophilin 3 (JPH3) along with the main HTT protein [10, 11].

3.4 Domain–Domain Analysis

The interactions between the nine proteins along with HTT are based on the PPI study. The proteins butyrylcholine esterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein1 (HAP1), brain-derived neurotrophic factor (BDNF), Junctophilin 3 (JPH3), and HTT have been noticed. Along with the proteins taken to construct the network, the top proteins having high interactions in the network are also considered. Here, the relationship between the proteins which may or may not be a direct or indirect relationship was discovered. Here, the HTT protein has been considered for the analysis for the comparison with the proteins, namely butyrylcholine esterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein1 (HAP1), brain-derived neurotrophic factor (BDNF), Junctophilin 3 (JPH3), and the high interaction proteins identified from the PPI network. First, the domain and domain architecture contents of the HTT and butyrylcholine esterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein1 (HAP1), brain-derived neurotrophic factor (BDNF), and Junctophilin 3 (JPH3) proteins along with the other high interacted proteins are compared individually, and domains and domain architectures that are only present in those proteins are identified. Second, public repositories are searched for details on specific pathways that proteins with promiscuous domains, such as butyrylcholine esterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein1 (HAP1), brain-derived neurotrophic factor (BDNF), Junctophilin 3 (JPH3), and other high interacted proteins which may be involved. To compare and identify the domains, the Interpro has been used which deals with the insertion of the FASTA of the nine proteins and compares [16, 18].

4 Result

The result includes all the three analyses, namely the evolutionary analysis, PPI analysis, and domain–domain analysis which finds the new protein that causes Huntington's disease.

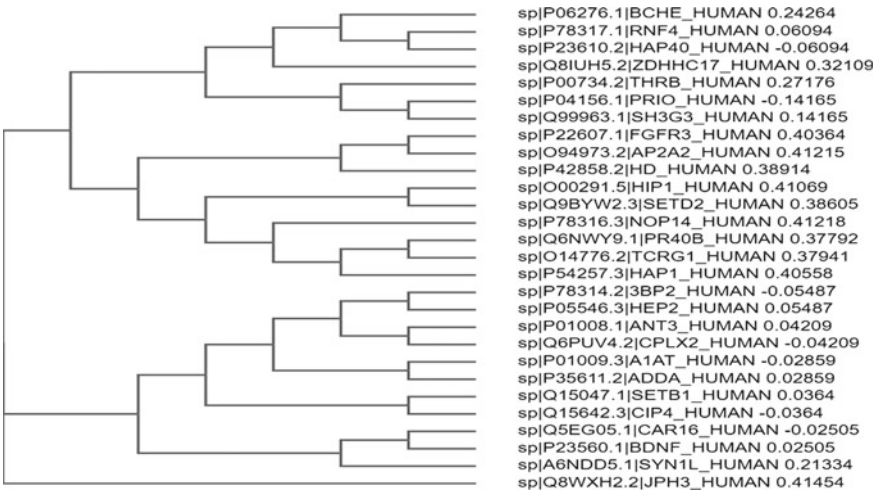


Fig. 2 Evolutionary analysis of 28 Huntington's disease-causing proteins

4.1 Evolutionary Analysis

The topologies were analyzed using the neighbor-joining technique used for phylogenetic inference. Figure 2 depicts the phylogenetic analysis of Huntington's disease with 28 proteins. Branch lengths are a sign of genetic divergence; the longer the branch, the more genetic divergence has taken place. This aids in identifying similarities among the 28 proteins used. The nine proteins with a lot of similarities with the HTT protein out of the 28 total proteins were discovered. The nine proteins are butyrylcholine esterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein1 (HAP1), brain-derived neurotrophic factor (BDNF), and Junctophilin 3 (JPH3). The nine proteins share 0.24264, 0.32109, 0.40364, 0.41215, 0.41069, 0.38605, 0.40558, 0.02505, and 0.41454 percent similarity in their amino acid sequences with HTT.

4.2 Protein–Protein Interaction (PPI) Analysis

Construct the PPI network using the Search Tool for the Retrieval of Interacting Genes/Proteins tool (STRING) for the nine proteins obtained from the previous evolutionary analysis. Figure 3 depicts the PPI network for Huntington's disease-causing proteins. All the proteins except BChE, FGFR3, and AP2A2 have a direct relationship with HTT. Whereas the BChE has an indirect relation with HTT by connecting with BDNF, FGFR3 has an indirect relation with HTT by interacting

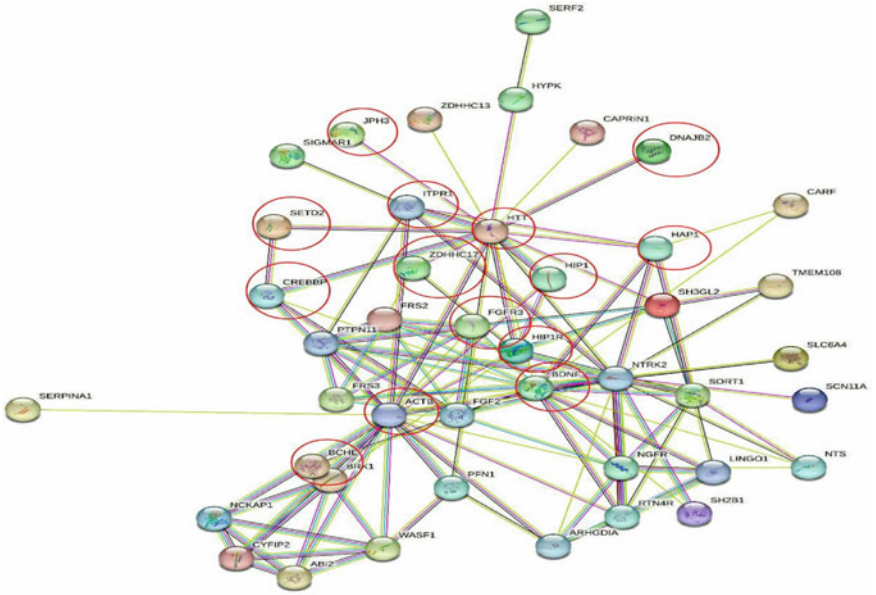


Fig. 3 PPI analysis of similar Huntington's disease-causing proteins

with BDNF, and AP2A2 has an indirect connection with HTT by interacting with ACTB.

4.3 Domain–Domain Analysis

The domain–domain analysis is applied using the Interpro tool for the 14 proteins such as butyrylcholinesterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein 1 (HAP1), brain-derived neurotrophic factor (BDNF), Junctophilin 3 (JPH3) proteins, high interacted proteins the Actin Beta (ACTB), Inositol 1,4,5-trisphosphate receptor type 1 (ITPR1), CREB binding protein (CREBBP), DnaJ heat shock protein family (Hsp40) member B2 (DNAJB2), and Huntingtin-interacting protein 1-related (HIP1R). Figures 4 and 5 show the domains and families of HTT and α-Adaptin proteins. In domain analysis, we observed that α-Adaptin and the HTT proteins share some common domains and functionalities such as ARMtype fold and ARM-like.

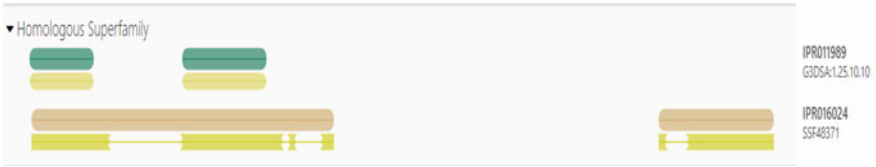


Fig. 4 HTT protein domains and families

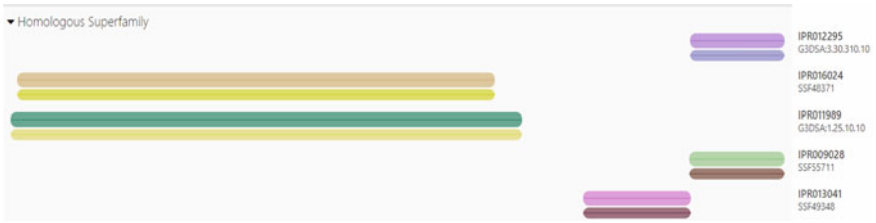


Fig. 5 α-Adaptin protein domains and families

Moreover, to that, it has been identified that there is a relationship between HTT and BDNF from the PPI analysis. The relation is also proven in many papers that BDNF can influence HTT [15, 25, 26]. The BChE and BDNF are interconnected, and in the PPI network, the BChE is related to HTT by an indirect connection with BDNF. So, it is believed that BChE also influences HTT.

5 Conclusion

In this study, the evolutionary, PPI, and domain–domain interaction analysis has been implemented on HTT protein which causes Huntington's disease. PubMed, Uniport, Gene-Cards, and literature survey initially considered the 28 Huntington's disease (HD) proteins. Through the evolutionary analysis, examining these 28 proteins and identified nine similar proteins associated with HTT which are butyryl-cholinesterase (BChE), ZDHHC17, fibroblast growth factor receptor 3 (FGFR3), α-Adaptin, Huntingtin-interacting protein 1 (HIP1), SET domain-containing 2 (SETD2), Huntingtin-associated protein 1 (HAP1), brain-derived neurotrophic factor (BDNF), and Junctophilin 3 (JPH3). PPI was applied to these nine proteins and constructed the PPI networks to check the direct or indirect relations between these proteins and identified 5 new proteins which include Actin Beta (ACTB), Inositol 1,4,5-trisphosphate receptor type 1 (ITPR1), CREB binding protein (CREBBP), DnaJ heat shock protein family (Hsp40) member B2 (DNAJB2), Huntingtin-interacting protein 1 related (HIP1R). Finally applied domain–domain analysis on these 14 proteins. It revealed that α-Adaptin, BDNF, and BChE are the new causing

proteins for HTT. From this study, it is identified that diabetes could also cause Huntington's disease (HD) because BChE is the one of the causing proteins for diabetes. The future work includes applying the structure level analysis and doing the drug target interaction network on the newly found three proteins (α-Adaptin, BDNF, and BChE) to know the detailed interactions with Huntington's disease (HD).

References

1. Przedborski S, Vila M, Jackson-Lewis V (2003) Neurodegeneration: what is it and where are we? J Clin Invest 111:3–10
2. Nopoulos PC (2016) Huntington disease: a single-gene degenerative disorder of the striatum. AICH—Servier Research Group 2016 18(1):91–98
3. Gatto EM, Rojas NG, Persi G, Etcheverry JL, Cesarini ME, Perandones C (2020) Huntington disease: advances in the understanding of its mechanisms. Clin Parkin & Related Disord 3:100056
4. Roos RAC (2010) Huntington's disease: a clinical review. RoosOrphanet J Rare Diseases 5:40
5. McColgan P, Tabrizi SJ (2018) Huntington's disease: a clinical review. European J Neurol 25(1)
6. Anne B (2003) Young, huntingtin in health and disease. J Clin Invest 111(3):299–302
7. Gitler AD, Dhillon P, Shorter J (2017) Neurodegenerative disease: models, mechanisms, and a new hope. Disease Models Mechan 10:499–502
8. McLennan DA (2010) How to read a phylogenetic tree. Evo Edu Outreach 3:506–519
9. Liu Y, He R, Qu Y, Zhu Y, Li D, Ling X, Xia S, Li Z, Li D (2022) Integration of human protein sequence and protein-protein interaction data by graph autoencoder to identify novel protein-abnormal phenotype associations. Cells 11:2485
10. Calabrese G, Molzahn C, Mayor T (2020) Protein interaction networks in neurodegenerative diseases: from physiological function to aggregation. JBC Rev 298(7)
11. Proskura AL, Vechkapova SO, Zapara TA, Ratushniak AS (2017) Protein-protein interactions of huntingtin in the hippocampus. Molekulyarnaya Biologiya 51(4):734–742
12. Rao BM, Rao VS, Rao KS (2019) Analysis of pruned protein protein interaction network using diffusion method. IJRECE 7(1)
13. Rao VS, Srinivas K, Sujini GN, Kumar GNS (2014) Protein-protein interaction detection: methods and analysis. Inter J Prot
14. Heinz A, Schilling J, van Roon-Mom W, Krau S (2021) The MID1 protein: a promising therapeutic target in huntington's disease. Front Genet, Sec. RNA
15. Wanker EE, Anne A, Schindler F, Trepte P, Schnoegl S (2019) The pathobiology of perturbed mutant huntingtin protein–protein interactions in huntington's disease. J Neurochem 151:507–519
16. Prasanthi LS, Kumar RK (2015) ID3 and its applications in generation of decision trees across various domains—survey. Inter J Comp Sci Info Tech 6(6):5353–5357
17. Fodale V, Pintauro R, Daldin M, Altobelli R, Spiezia MC, Bisbocci M, Macdonald D, Bresciani A (2020) Analysis of mutant and total huntingtin expression in Huntington's disease murine models. Sci Rep 10(1):22137. https://doi.org/10.1038/s41598-020-78790-5. PMID: 33335120; PMCID: PMC7746729
18. Rao AA, Srinivas K, Rajender R, Das UN (2008) Prediction of Butyrylcholinesterase function through domain analysis and protein-protein interaction. Current Nutrition Food Sci 4:176–184
19. Sridhar GR, Nirmala G, Apparao A, Madhavi AS, Sreelatha S, Rani JS, Vijayalakshmi P (2005) Serum butyrylcholinesterase in type 2 diabetes mellitus: a biochemical and bioinformatics approach. Lipids Health Dis 8(4):18

20. Wang Y, Zhang H, Zhong H, Xue Z (2021) Protein domain identification methods and online resources. Comp Struct Biotech J 19
21. Tartari M, Gissi C, Lo Sardo V, Zuccato C, Picardi E, Pesole G, Cattaneo E (2008) Phylogenetic comparison of huntingtin homologs reveals the appearance of a primitive polyQ in sea urchin. Mol Biol Evol 25(2):330–338
22. Hussain B, Khalid H, Nadeem S, Sultana T, Aslam S (2012) Phylogenetic and chronological analysis of proteins causing Alzheimer's, Parkinson's and Huntington's diseases. Int J Bioauto 16(3):165–178
23. Johnson SL, Tsou W-L, Prifti MV, Harris AL, Todi SV (2022) A survey of protein interactions and posttranslational modifications that influence the polyglutamine diseases. Front Mol Neurosci, 14 September. Molecular Signalling and Pathways.
24. Podvin S, Rosenthal SB, Poon W, Wei E, Hook KMFV (2022) Mutant huntingtin protein interaction map implicates dysregulation of multiple cellular pathways in neurodegeneration of huntington's disease. J Huntington's Disease 11:243–267
25. Kerkis I, Silva JM, Wenceslau CV, Mambelli-Lisboa NC, Frare EO (2020) Brain-derived neurotrophic factor and stem cell-based technologies in huntington's disease therapy. In: Neurodegenerative diseases—molecular mechanisms and current therapeutic approaches. IntechOpen, London, UK
26. Ji H, Dai D, Wang Y, Jiang D, Zhou X, Lin P, Ji X, Li J, Zhang Y, Yin H, Chen R, Zhang L, Xu M, Duan S, Wang Q (2015) Association of BDNF and BChE with Alzheimer's disease: meta-analysis based on 56 genetic case-control studies of 12,563 cases and 12,622 controls. Exp Ther Med 9(5):1831–1840

A Novel Res + LSTM Classifier-Based Tomato Plant Leaf Disease Detection Model with Artificial Bee Colony Algorithm

Alampally Sreedevi and Manike Chiranjeevi

Abstract World economy is mainly depending on agriculture. In recent days, many of the plants have been affected by various types of diseases, which will affect the economic growth of the country. Early detection of plant diseases may help to increase the profit rate for farmers. Tomatoes are the most consumable vegetable around the world. Catastrophic influence on food production safety is caused by plant diseases, and it decreases the quantum and eminence of agricultural products. The tomato plants are affected with different types of leaf spot diseases like grey spot, mosaic, leaf spot, brown spot, Alternaria, and also the rust and pests that may affect the growth of tomato plants at their various growth stages. Hence, the identification of leaf spot diseases that occur in the tomato plant is important to increase the survival rate of the tomato plants. Manual inspections of tomato leaf diseases are secured more time, and it mainly depends on the expert's knowledge. To address this problem, a novel hybrid deep learning technique is developed for the detection of leaf spots in the tomato plant. The tomato leaf images with normal and affected tomato leaves are collected from real world and traditional benchmark datasets. After the tomato leaves are segmented using the U-net model, the resultant segmented images are subject to the classification phase. Finally, leaf spots in tomato leaves are classified with the help of residual network and long short-term memory (ResNet-LSTM), and the parameter in the Resnet and LSTM is tuned by utilizing artificial bee colony (ABC) algorithm to offer effective leaf spot classification rate. Finally, the severity level is computed. The experimental results demonstrate the effectiveness of the proposed model in detecting the leaf spot in the tomato plant.

Keywords Tomato plant leaf spot detection · Residual network · Long short-term memory · U-net based-segmentation · Artificial bee colony algorithm

A. Sreedevi · M. Chiranjeevi (✉)
Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh 522502, India
e-mail: chiranjeevi@klh.edu.in

A. Sreedevi
e-mail: sreedevi.a@klh.edu.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_3

1 Introduction

Tomato is the most commonly used vegetable in the world, and it can be affected due to various kinds of diseases based on fungal, bacterial infections, and several environmental factors. The tomato gray leaf spot is the type of disease that will occur in the tomato plant [3]. This will destroy the photosynthesis process of the leaves by damaging the leaves. Tomato spot diseases have affected the growth of the plants, and they will spread easily through the entire plant [6]. It is difficult to prevent tomato leaf spot diseases at the final stage [7]. Hence, the initial stage of leaf spot disease detection is very important at the early stage [8]. The infection period of the leaf spot pathogen is categorized into four periods that are onset, latency, contact, and invasion period [9]. The pathogens are in contact with the other host plants during the contact period [10]. Pathogens invading the host and establishing a parasitic relationship between them occurred during the invasion period [11]. The pathogens started to showcase the obvious symptoms in the latency period [14]. The propagation and spreading through the entire host by observing all the nutrients encountered in the onset period [12]. Therefore, automatic identification of tomato leaf spot diseases is essential for identifying the diseases at the initial stage to increase the survival rate of the tomato plants [13].

The traditional methods require large-scale planting and provide very less diagnosis efficiency [15]. Hence, several machine learning-based approaches are deployed for providing incomparable results over the traditional approaches [16]. Moreover, these techniques require a limited number of sample images and effectively identify the rigorous patterns in the collected sample images [17]. But, these methods are needed large network training for the classifiers to produce highly accurate results, and it is more time-consuming for the detection process [18]. In addition, the generalization ability of tomato leaf spot detection is very low. To resolve the different drawbacks that occur in the detection of tomato leaf spot diseases using newly developed deep network-based architecture [19]. Because, this deep network-based architecture provides promising results in semantic segmentation, image classification, and object detection. The major contributions of the newly investigated tomato leaf spot disease detection methodology are explained below.

- To present an efficient tomato leaf spot disease detection model for detecting the leaf spot disease at the initial stage and take preventive measures to control the effect and spreading of diseases through the entire plant.
- To utilize an ABC algorithm for optimizing the parameters in the ResNet and LSTM networks to enhance the detection outcome by maximizing the classification accuracy.
- To design and ResNet-LSTM-based detection model for the accurate detection of tomato leaf spot diseases. Here, the hidden neurons and the activation functions such as ReLu, linear, softmax, sigmoid, and tanh functions are optimized in the ResNet structure, and the hidden neurons as well as the epochs are optimized in the LSTM structure to improve the detection efficiency. Finally, the level of severity was also examined.

- The efficiency of the developed tomato leaf spot disease detection approach is verified by taking a comparative analysis of the developed model and the conventional tomato leaf detection methodologies in terms of distinct estimation measures.

The remaining parts of the developed tomato leaf spot disease detection model are structured as follows. Part II describes the conventional deep learning-based tomato leaf spot disease detection model and its merits as well as demerits. Part III explains the architectural representation of the developed tomato leaf spot disease detection approach and the collection of the dataset description. Part IV explains the segmentation and the classification approach utilized for the developed model. Part V conveys the results and discussions of the developed tomato leaf spot disease detection model. Finally, Part VI gives the conclusion of the developed model.

2 Literature Survey

2.1 Related Works

In 2022, Kodamana et al. [1] introduced a deep learning-assisted model for the detection of tomato begomovirus infections in tomato plants. The early detection of leaf diseases was possible in the developed CNN-based architecture. Here, the classification model has been built with Visual Geometric Group 16 (VGG 16) for identifying the begomovirus infections in tomato plants. Quite promising results were achieved in the developed model by comparing the performance measures in terms of accuracy. In 2022, Pan et al. [2] adopted a deep neuro-fuzzy based classification approach for classifying the diseases that occur in tomato leaves. The complex features were effectively identified in two layers like pooling and interference layer in the fuzzy network. Thus, the extracted features were subjected to the fully connected layer for the final classification. Further, eight different kinds of unaffected and affected tomato leaves were utilized for the implementation, and three evaluation indexes were considered for the performance validation.

In 2022, Ahmed et al. [3] employed a lightweight deep neural architecture for the classification of diseases present in tomato leaves. Initially, the illumination of the images was corrected for enhancing the effectiveness of the classification process. Then, the pertained network has been utilized for retrieving features from the images using the MobileNetV2 model for classifying the diseases. In 2020, Chen et al. [4] implemented a self-supervised collaborative multi-network for fine-grained visual categorization of tomato diseases. This proposed structure consisted of three networks, namely location, feedback, and classification network named as LFC-Net for effectively detecting the informative regions in the tomato leaves. The informative regions were detected by the location network, optimization was applied in the feedback network, and the final classification has been performed in the classification network. In 2021, Zhou et al. [5] demonstrated a deep residual dense network (DRDN) for categorizing tomato leaf diseases very effectively. The traditional RDN

network has been utilized to obtain the image super-resolution, and then, this network has been restructured for the classification process. High satisfactory performance has been accomplished by using this developed classification model.

2.2 Problem Statement

To increase the overall profit of the stakeholders and also ensure global food security, the effective detection of tomato leaf diseases within the appropriate time is very important. The tomato leaf diseases are identified using the artificial recognition method very effectively, but these are more time-consuming and highly subjective to laborious experiments. Moreover, the small discriminative feature determination is very difficult in the artificial recognition approach. Hence, machine learning and deep learning-assisted structures are demonstrated to identify leaf diseases within the proper time. The advantages while using these deep learning-based methodologies and the disadvantages that occur after the implementation of these deep learning-based methodologies are illustrated in Table 1. VGG16 [1] does not want manual inception for finding the diseased leaves. In addition, it requires low memory space and less computational power. But, it has been accomplished with low detection accuracy. Further, the reliability of the system is very low due to human error. Deep neuro-fuzzy [2] reduces the impact of noises present in the acquired images. Moreover, it effectively solves the gradient dispersion function. Yet, it has tackled high overfitting, high data leakage, and high data imbalance problems. Therefore, it is hugely expensive for designing the architecture of DNN [3] to improve the detection speed, and hence, the diseases are detected earlier. Furthermore, the sensitivity of the system is very high. Nonetheless, it is infeasible for low-end devices. Moreover, it limits the system usability and scalability in user-grained applications, and hence, the efficiency of the system is very low. LFC-Net [4] sensitively discards the complex samples that restrict the generalization capability. Furthermore, it requires low cost for the entire implementation. Nevertheless, it is a labor-intensive method, and it needs further improvements in the system's scalability and reliability. RDN [5] reduces the tedious manual monitoring tasks and minimizes the human effort requirement. But, large dimensional data are not effectively handled in this approach. In addition, it provides low performance in keeping model size because a big model size increases the computational cost and complexity. Therefore, these difficulties are resolved by using the given detection of tomato leaf disease using deep learning-based strategies.

Table 1 Features and challenges of the familiarly used deep learning-related tomato leaf disease classification methodologies

Author [citation]	Methodology	Features	Challenges
Kodamana et al. [1]	VGG16	• It does not want manual inception for finding diseased leaves • It requires low memory space and less computational power	• It has been accomplished with low detection accuracy • The reliability of the system is very low due to human error
Pan et al. [2]	Deep Neuro-fuzzy	• It reduces the impact of noises present in the acquired images • It effectively solves the gradient dispersion function	• It has tackled high overfitting, high data leakage, and high data imbalance problems • It is hugely expensive for designing the architecture
Ahmed et al. [3]	DNN	• It improves the detection speed, and hence, the diseases are detected earlier • The sensitivity of the system is very high	• It is infeasible for low-end devices • The efficiency of the system is very low • It limits the system's usability and scalability in user-grained applications
Chen et al. [4]	LFC-Net	• It sensitively discards the complex samples that restrict the generalization capability • It requires low cost for the entire implementation	• It is a labor-intensive method • It needs further improvements in the system's scalability and reliability
Zhou et al. [5]	DRDN	• It reduces the tedious manual monitoring tasks and minimizes the human effort requirement	• Large dimensional data are not effectively handled in this approach • It provides low performance in keeping model size because a big model size increases the computational cost and complexity

3 Description of Introduced Tomato Leaf Spot Disease Detection Model

3.1 Proposed Model and Description

Agriculture is a significant contributor to obtaining national income for many countries. Plant pathogens like viruses, bacteria, and fungi are the main reasons for plant diseases. In agriculture, there is a need for discovering diseases in the early stage and providing appropriate timing control to minimize the production cost, reduce

damage, and increase income. The intelligent tomato leaf spot detection approaches like regression, artificial intelligence, and deep learning approaches are helpful to identify the illness very effectively, and these models give more reliable results. In addition, the mortality rate has been effectively reduced, and consistent results regarding affected diseases were obtained using this disease detection model. The deep learning-based structures are used to decrease the effect of the continuous human monitoring and intensifying of labor works. Therefore, to provide better results in the detection of tomato leaf spots, a deep network-based architecture is assisted, and the architectural demonstration of the developed tomato leaf spot disease detection method is given in the following Fig. 1.

In this newly recommended deep network-based leaf spot disease detection model, various types of leaf spot diseases are detected at the early stage and provide preventive control measures to decrease the spread rate of the leaf spot diseases throughout the entire plant. The required sample leaf images with affected and unaffected images are collected from the traditional benchmark datasets and real-world scenarios. Firstly, the collected images are applied to the U-net-based image segmentation approach, and the resultant segmented tomato leaf images are given to the detection phase. The ResNet and the LSTM classifier network are utilized for the detection of various types of tomato leaf spot diseases in the tomato leaves. Here, the hidden neurons and the activation function like sigmoid, softmax, tanh, linear, and ReLu are optimized with the usage of the newly developed ABC algorithm. Furthermore, the

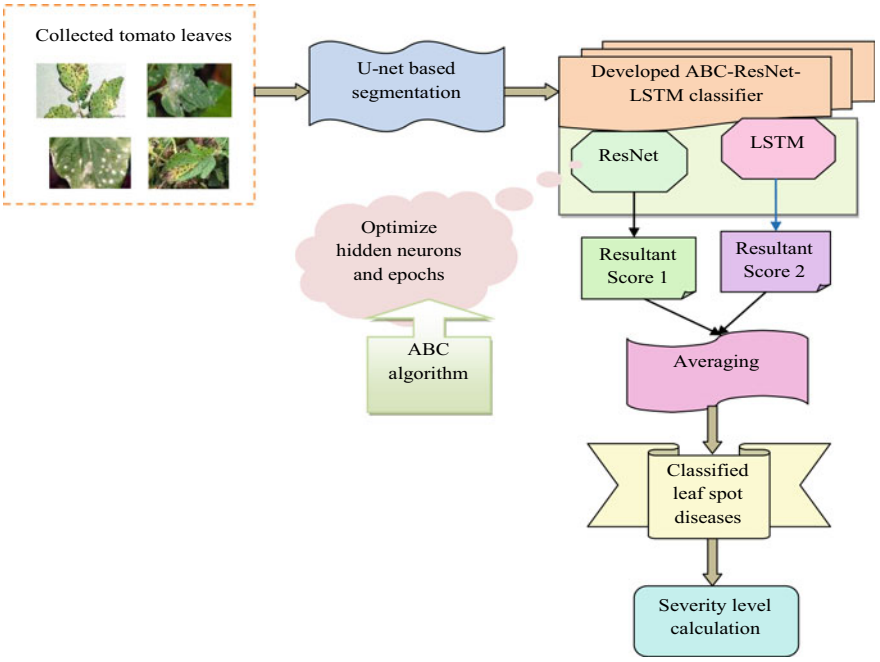


Fig. 1 Architectural description of the suggested tomato leaf spot image detection model

hidden neurons and the epochs in the LSTM are optimized with the same developed ABC algorithm. Thus, the output from the ResNet classifier and the output from the LSTM network is subjected to an averaging process to obtain the final classification results. This optimization of parameters in the developed tomato leaf spot disease detection model should increase the accuracy very effectively. At last, the efficiency of the developed model is guaranteed by comparing the computation measures among the proposed tomato leaf spot disease model and other traditional detection models.

3.2 Dataset Information

There are two types of datasets are utilized for the detection of tomato leaf spot diseases are explained below.

Dataset 1 (Tomato leaf disease detection): The required tomato leaf images obtained from the standard source are "https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf: access date 2022-09-10". The dataset presents in the format of train, val, cnn_train.py. The total available size of the dataset is 186.22 MB. There are various types of leaf disease sample images present in the dataset, namely early blight, bacterial spot, target spot, leaf mold, late blight, and tomato mosaic virus. There are six sample sets of images are chosen for each disease.

Dataset 2 (Mendeley data): The collected sample images from the online source are "https://data.mendeley.com/datasets/ngdgg79rzb: access date 2022-09-30". The file information is Taiwan.7z, where 7z is the file extension type, and the available file size is 46 MB. Various types of leaf diseases collected in this dataset such as Fungal-leaf_Early_blight, Fungal-leaf_Gray_mold, Pest-damage-Tuta_absoluta, Pest-Leaf_serpentine_miner, and Pest-tobacco caterpillar leaf damage. There are four sample images are chosen for each disease.

The collected input tomato leaf images are indicated by $\mathrm{TSD}_y^{\mathrm{sam}}$, where $y = 1, 2, 3, \ldots Y$. Here, the term Y represents the total amount of images to be collected.

4 Leaf Segmentation and Classification Using Deep Learning Architecture

4.1 Segmentation of Leaf Spot Using UNET

In this developed model, the input to be applied for the recommended leaf spot disease detection model is the collected $\mathrm{TSD}_y^{\mathrm{sam}}$ sample image. The U-net [25] is built up with two basic architectures that are the contracting path and the expanding path. The unpadded convolutions are involved in the contracting path, and it mainly consists of 2×2 convolutions. These are applied repeatedly after the 2×2 maximum pooling operation and ReLu while considering the down-sampling value stride2. In

the expanding path, the upsampling with the feature map is presented with output =stride2, and the 2 × 2 convolution operation is carried out in the feature map that halves the total number of feature channels in the U-net structure. Finally, the feature concatenation with cropped feature map is accomplished based on 3 × 3 convolutions with the utilization of the ReLU. Totally, 23 layers are employed in this U-net structure. Finally, the segmented output is obtained at the final stage with high resolution.

4.2 Utilization of ABC Algorithm

The ABC algorithm is demonstrated in this developed tomato leaf spot disease detection model because it provides high generalization ability in the search space and it has better balancing availability in both the exploration mechanism and the exploitation algorithm. Moreover, the computation complexity of this ABC algorithm is very low.

ABC [20] is constructed based on the behavior of the honey bee swarms by handling the unconstrained benchmark heuristic operation. To obtain the best optimal solution, two mechanisms have functioned that is the exploitation mechanism and the exploration mechanism. The initial population solutions were indicated by $z = 1, 2, 3,\ldots,B_{\text{em}}$, where the term B_{em} defines the total amount of employed bees in the search space. All the solutions S are involved with N dimensional vectors for optimizing [26, 7] the parameters. After initializing the parameters in the search space, the position of the population is given to the repeated cycle for the search process that is indicated by cle $= 1, 2, 3,\ldots,\text{M}_{\text{cn}}$, where the term cle denotes the repeated cycle, and the term M_{cn} is the maximum cycle number. The food selection probability is estimated using Eq. (1).

$$\text{PL}_z = \frac{\text{Fit}_z}{\sum_{z=1} B_{\text{em}}(z)} \tag{1}$$

Here, the term Fit_z denotes the fitness value of the current solution, z, and the total number of food source positions is indicated $B_{\text{em}}(z)$. The neighbor food source position is represented as $z + 1$ and this position is used to find the food source.

The scout in the search space produces a new food source position and the updated position is illustrated in Eq. (2).

$$S_z^{y(\text{New})} = \min S_z^y + \varphi\left(\max S_z^y - \min S_y^z\right) \tag{2}$$

The term φ gives the randomly generated parameter in the range between $[-1, 1]$. This position applies to all the k parameters. Hence, the execution process is stopped until obtaining the best solution based on the fitness function.

The neighboring food source position of the bees is determined by using the randomly created parameters. The other parameters used in the system are kept constant. The expression related to finding the neighborhood food source is given in Eq. (3).

$$S_{zy}^{\text{New}} = S_{zy}^{\text{Old}} + \varphi\left(S_{zy}^{\text{Old}} - S_{yz}\right) \tag{3}$$

Here, the term φ is used to perform a multiplier function taken in the interval between $[-1, 1]$ and the variables $y \neq z$ and that are defined in the range between $k = 1, 2, 3, \ldots S$. The kth parameter in the Sth solution is represented by the term S_{zy}.

4.3 Proposed ABC-ResNet-LSTM Based Classifier

The developed classifier for tomato leaf spot disease detection is mainly helpful for detecting leaf spot diseases in the earlier stage. The hidden neurons and the epochs in the LSTM classifier are effectively optimized by the ABC algorithm and also the activation functions like ReLu, linear, softmax, sigmoid, and tanh as well as the hidden neurons in the ResNet are optimized with the same ABC algorithm for the maximization of accuracy. The ResNet and LSTM classifiers are selected in the developed tomato leaf spot disease detection model because it effectively classifies the diseases at the initial stage and may be helpful for farmers to increase the economic rate.

ResNet [22]: The ResNet structure effectively identifies the important features required for the detection of tomato leaf spot diseases. It consists of a batch normalization layer, activation layer, softmax layer, maximum pooling layer, fully connected layer, flattens layer, and the average pooling layer. The segmented output is given to the convolution layer of the ResNet architecture. The simplest ResNet architecture involved four stages for classifying the tomato leaf spot. All the stages consist of multiple identity mapping modules and the sampling module to provide higher effective detection results. The final stage output is given to the average pooling layer and then the flattened layer and the fully connected layer. In the flattened layer, the multi-dimensional features are converted into one-dimensional features. Then, the softmax layer is introduced to check whether the internal characteristics of the network have obtained the optimal level in the residual network. This mapping of the softmax layer is used to reduce the use of network parameters and reduce computational complexity.

LSTM [23]: The general LSTM block comprises three interactive gates, namely input, output, and forget, and also one memory unit. The previous state output is stored in the memory cell. The amount of data to be stored in the memory cell is defined in the input gate at the time s,and the forget gate decides whether the data will refuse or pass to the input gate at a time $s - 1$. The classified output is obtained from the output gate. The functions of all gates are given in below Eqs. (4–9), respectively.

$$N_s = \sigma(M_{NZs} + P_s H_{s-1} + \mathrm{BI}_s) \tag{4}$$

$$R_s = \sigma(M_{\mathrm{RZ}s} + P_R H_{s-1} + \mathrm{BI}_R) \tag{5}$$

$$\tilde{D}_s = \tanh(M_D Z_s + P_D H_{s-1} + \mathrm{BID}) \tag{6}$$

$$D_s = R_s L D_{s-1} + N_s L \tilde{D}_s \tag{7}$$

$$\mathrm{Op}_s = \sigma\left(M_{\mathrm{Op}} Z_s + P_{\mathrm{Op}} H_{s-1} + \mathrm{BI}_{\mathrm{Op}}\right) \tag{8}$$

$$H_s = \mathrm{Op}_s L \tan h(D_s) \tag{9}$$

The input at a time s is indicated by Z_s and the weight matrices are defined by the terms P_* and M_* respectively. The bias matrix is indicated by BI_*, the hidden state is represented by H_* and the memory state is denoted by D_s. The input, output and forget states are described by the terms Z_s, Op_s and R_s respectively.

4.4 Objective Function of the Developed ABC-ResNet-LSTM-Based Classifier

The developed ABC-ResNet-LSTM-based tomato leaf spot disease detection method is mainly useful to detect diseases at the initial stage. The parameters present within the developed classifier are optimized by using the ABC algorithm. In ResNet, the hidden neurons are tuned in the interval of [5, 255], and the activation functions like ReLu, softmax, sigmoid, linear, and tanh functions are optimized. In LSTM, the hidden neurons and the epochs are optimized in the interval of [5, 255], and the number of epochs is optimized in the range between [50, 100]. This optimization process improves the accuracy of the developed tomato leaf spot disease detection model. Thus, the score 1 output obtained from the ResNet and the score 2 output obtained from the LSTM are subjected to an averaging process to obtain the tomato leaf spot classified output. The objective function is represented in below Eq. (10).

$$\mathrm{FT} = \underset{\left\{\mathrm{ND}_{w*}^{\mathrm{Res}}, \mathrm{AN}_{l*}^{\mathrm{Res}}, H_{s*}^{\mathrm{LSTM}}, \mathrm{Epoh}_{m*}^{\mathrm{LSTM}}\right\}}{\arg\min} \left(\frac{1}{\mathrm{Acry}}\right) \tag{10}$$

Here, the term $\mathrm{ND}_{w*}^{\mathrm{Res}}$ gives the optimized hidden neurons in ResNet, $\mathrm{AN}_{l*}^{\mathrm{Res}}$ is the optimized activation function in the ResNet, H_{s*}^{LSTM} is the optimized hidden neurons in LSTM, and $\mathrm{Epoh}_{m*}^{\mathrm{LSTM}}$ is the optimized epochs in the LSTM classifier. Moreover, the term Acry is the accuracy, and it is calculated based on the positive and negative

observation values that are given in below Eq. (11).

$$\text{Acry} = \frac{A_{\text{PS}} + A_{\text{NG}}}{A_{\text{PS}} + A_{\text{NG}} + B_{\text{PS}} + B_{\text{NG}}} \tag{11}$$

Here, the true positive and the false positive are indicated by the terms A_{PS} and B_{PS}, respectively. Moreover, the true negative and the false negative are represented by the terms A_{NG} and B_{NG}, respectively.

The resultant output score obtained from the ResNet architecture is represented by Sc_1, and the resultant classified outcome achieved from the LSTM network is indicated by Sc_2. These two score values are averaged for getting the promising classified outcome based on tomato leaf spot diseases, which are illustrated below in Eq. (12).

$$\text{Classified outcome} = \frac{\text{Sc}_1 + \text{Sc}_2}{2} \tag{12}$$

The architectural representation of the developed tomato leaf spot disease detection approach is shown in Fig. 2.

5 Results and Discussions

5.1 Experimental Setup

The newly presented tomato leaf spot disease detection method was implemented in Python, and the experiments were conducted for ensuring the efficiency of the developed model when analyzing divergent benchmark leaf spot detection approaches while regarding the positive and negative estimation measures. The total population taken for performing this experiment was 10. The chromosome length to be taken for validating the performance over different baseline approaches was considered to be 4, and also the maximum number of iterations should be taken as 25 for executing the experiments to find the best solution. The baseline tomato leaf spot disease detection methodologies to be taken for performing efficiency comparison between them was convolutional neural network [21], ResNet [22], LSTM [23], and ResNet-LSTM [24] by determining the performance measures such as Mathews correlation coefficient (MCC), precision, accuracy, specificity, sensitivity, negative predictive value (NPV), false negative rate (FNR), false positive rate (FPR), and false discovery rate (FDR).

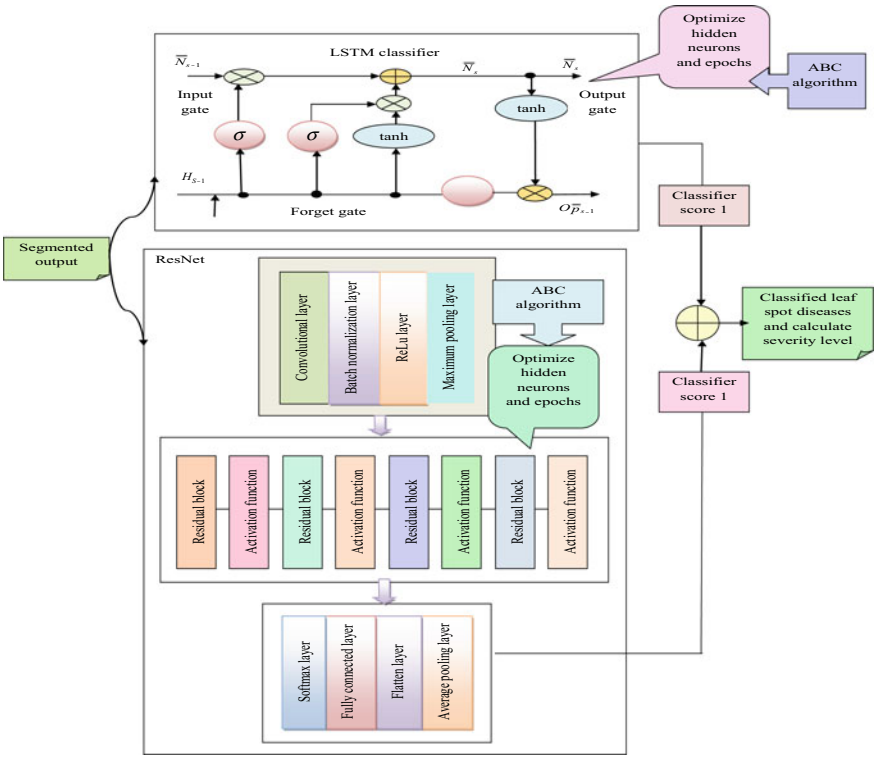


Fig. 2 ABC-ResNet-LSTM-based classifier for the developed tomato leaf spot disease detection model

5.2 Computation Metrics

(a) NPV: This is the type of negative measure; it is used to evaluate the performance of the developed model; it is calculated based on the true and false negative observations that are described in Eq. (13).

$$N_{\mathrm{PV}} = \frac{A_{\mathrm{NG}}}{A_{\mathrm{NG}} + B_{\mathrm{NG}}} \tag{13}$$

(b) Precision: The precision value is estimated by taking the ratio of the true negative rate to the sum of the true and false positive rate which is explained in below Eq. (14).

$$\mathrm{Precn} = \frac{A_{\mathrm{NG}}}{A_{\mathrm{PS}} + B_{\mathrm{PS}}} \tag{14}$$

(c) FNR: This negative measure is computed by calculating the ratio of false negative measure to the sum of true negative and positive observation that is expressed in Eq. (15).

$$F_{\mathrm{NR}} = \frac{B_{\mathrm{NG}}}{A_{\mathrm{NG}} + A_{\mathrm{PS}}} \tag{15}$$

(d) F1-score: The most familiarly used performance metric is the F1-score which is computed based on the true and false observations that are given in Eq. (16).

$$F1_S = \frac{2 \times A_{\mathrm{PS}}}{2A_{\mathrm{PS}} + A_{\mathrm{NG}} + B_{\mathrm{NG}}} \tag{16}$$

(e) MCC: It is evaluated based on the following Eq. (17).

$$\mathrm{MCC} = \frac{A_{\mathrm{PS}} \times A_{\mathrm{NG}} - B_{\mathrm{PS}} \times B_{\mathrm{NG}}}{\sqrt{(A_{\mathrm{PS}} + B_{\mathrm{PS}})(A_{\mathrm{PS}} + B_{\mathrm{NG}})(A_{\mathrm{NG}} + A_{\mathrm{PS}}B)(A_{NG} + B_{\mathrm{NG}})}} \tag{17}$$

(f) Sensitivity: It is the type of positive measure, and it is calculated using below Eq. (18).

$$\mathrm{SNT} = \frac{A_{\mathrm{PS}}}{A_{\mathrm{PS}} + B_{\mathrm{PS}}} \tag{18}$$

(g) Specificity: The general formula used to evaluate the specificity measure is illustrated in the following Eq. (19).

$$\mathrm{SPT} = \frac{A_{\mathrm{NG}}}{A_{\mathrm{NG}} + B_{\mathrm{PS}}} \tag{19}$$

(h) FDR: The FDR rate is calculated using the below formula that is mathematically expressed in Eq. (20).

$$\mathrm{FD} = \frac{B_{\mathrm{PS}}}{B_{\mathrm{PS}} + A_{\mathrm{PS}}} \tag{20}$$

(i) FPR: The FPR value is computed using Eq. (21).

$$\mathrm{FP} = \frac{B_{\mathrm{PS}}}{B_{\mathrm{PS}} + A_{\mathrm{NG}}} \tag{21}$$

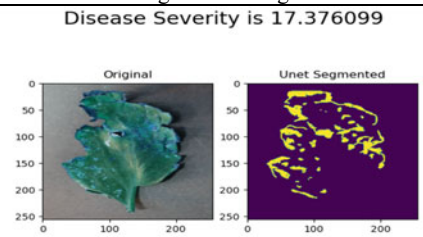
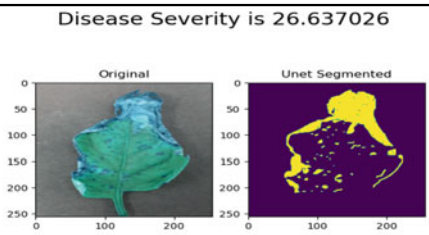
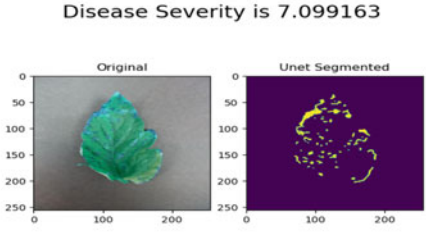
Set of images	Segmented images
Image 1	
Image 2	
Image 3	

Fig. 3 Figure resultant U-net-based segmented images for leaf spot disease detection model

5.3 Resultant Segmented Images

The output image obtained from the U-net-based image segmentation process is given in below Fig. 3.

5.4 Performance Validation Using Dataset 1 and Dataset 2

The following Figs. 4 and 5 give the performance computation of the developed tomato leaf spot disease detection model among recently utilized tomato leaf disease detection approaches using dataset 1 and dataset 2. From this computation results, the developed model ensured with 6.74% improved specificity, 5.61% improved

sensitivity, 23.07% improved MCC, 33.92% improved precision, 21.42% improved F1-score, and 4.39% improved accuracy than CNN model while taking the learning percentage value of 35 among dataset 1. The overall performance of the developed model is high when comparing all the performance measures among the developed and the existing methodologies.

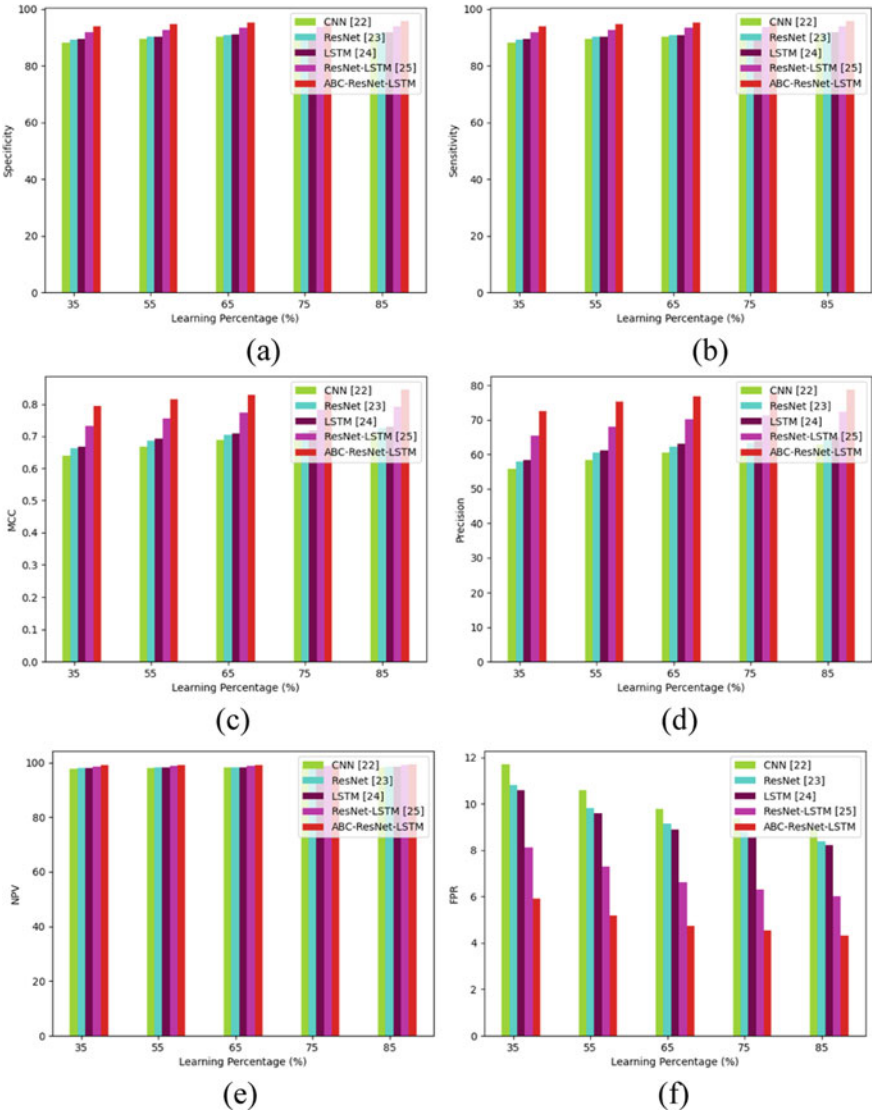


Fig. 4 Examination of performance on developed tomato leaf spot disease detection model using dataset 1 while considering with **a** specificity **b** sensitivity **c** MCC **d** precision **e** NPV **f** FPR **g** F1-score **h** FDR **i** FNR **j** accuracy among different existing detection approaches

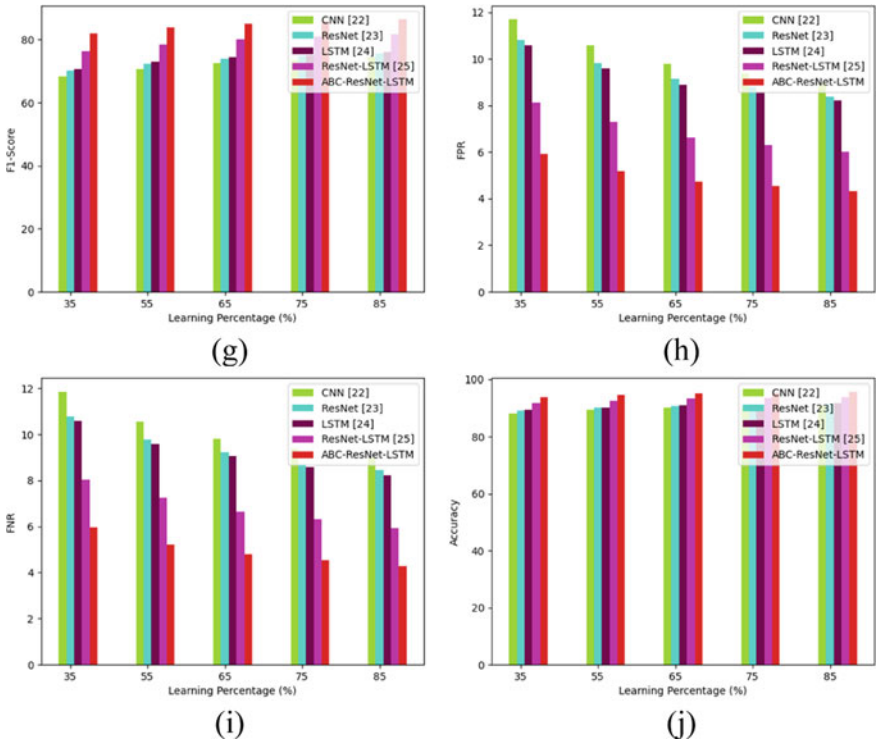


Fig. 4 (continued)

5.5 *Efficiency Computation on the Developed Model*

The efficiency analysis is taken by comparing the performance measures over the developed model, and the recently utilized tomato leaf disease detection approaches using dataset 1 and dataset 2 are illustrated in Tables 2 and 3 respectively. From this tabulation results, the developed model accomplished 5.15%, 4.46%, 4.23%, and 1.79% improved accuracy than CNN, ResNet, LSTM, and ResNet-LSTM. The performance of the developed model produced incomparable results than the existing tomato leaf disease detection approaches.

5.6 *Estimation of Computation Time for the Designed Method*

The estimation of computation time for the designed tomato leaf spot disease detection method is shown in Table 4 for dataset 1 and 2. Thus, the given designed method confirmed that it achieves minimum computation time for all datasets.

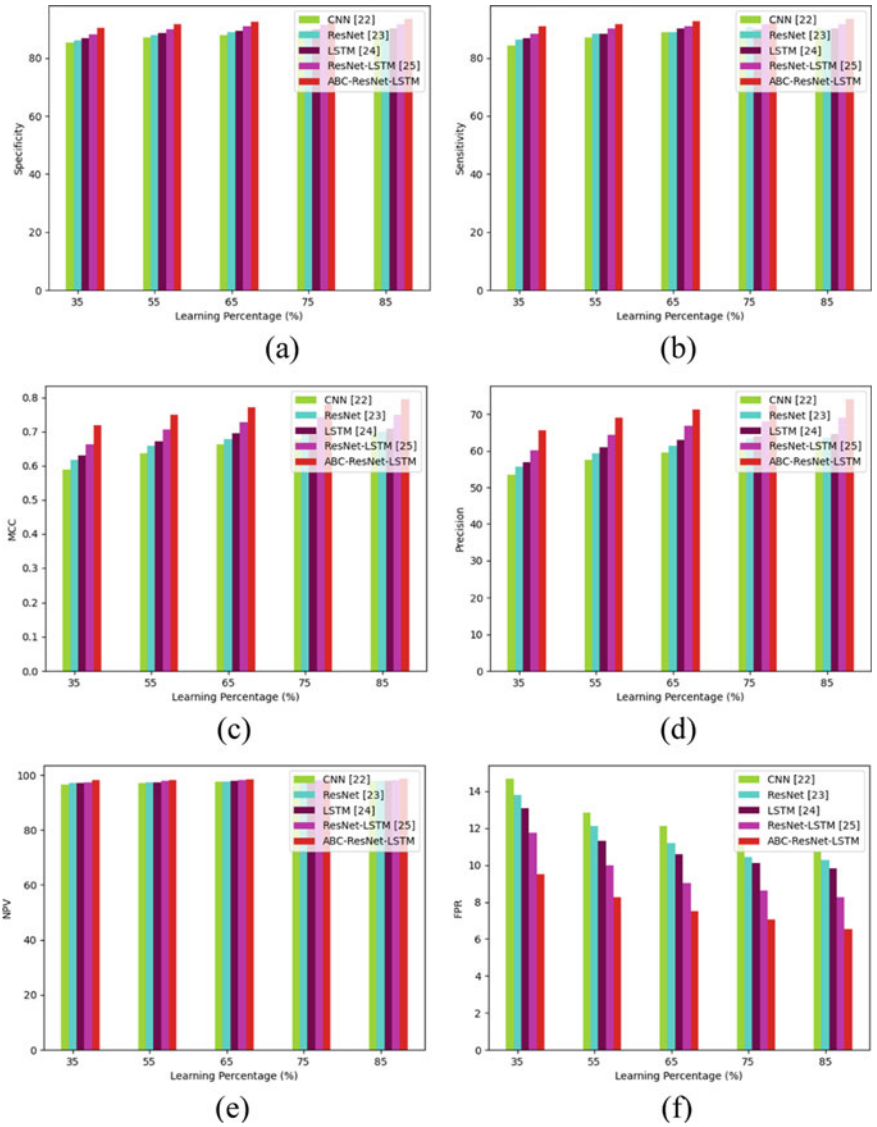


Fig. 5 Examination of performance on developed tomato leaf spot disease detection model using dataset 2 while considering with **a** specificity **b** sensitivity **c** MCC **d** precision **e** NPV **f** FPR **g** F1-score **h** FDR **i** FNR **j** accuracy among different existing detection approaches

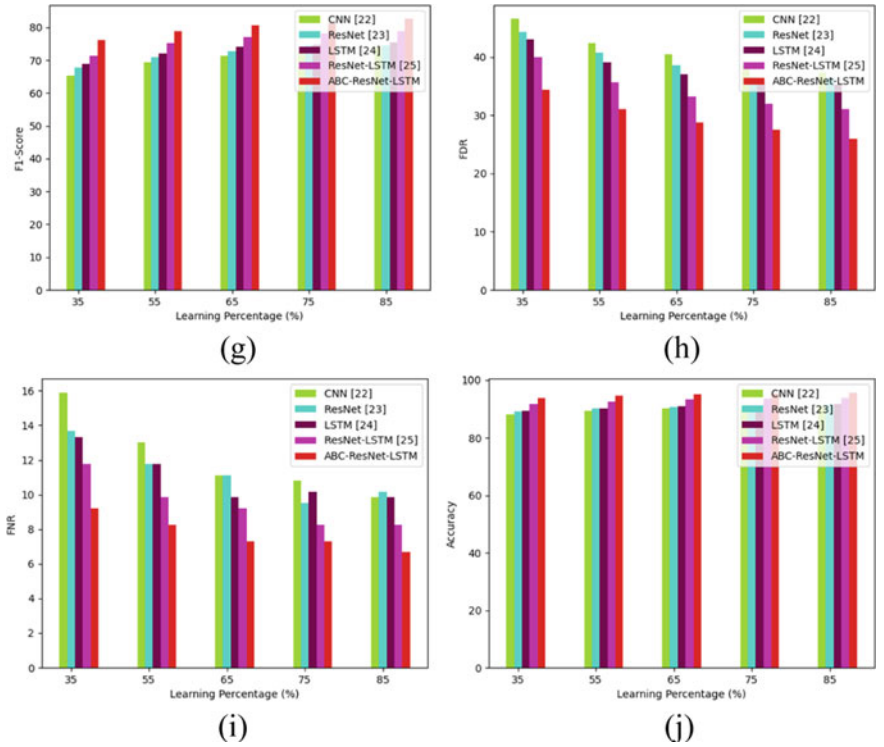


Fig. 5 (continued)

6 Conclusion

The newly presented deep structure-based model was used to effectively detect the tomato leaf spot diseases at the initial stage to increase the survival rate of the tomato plants. The collected sample images were subjected to the U-net-based segmentation, and the resultant images were fed to the ResNet and LSTM-based disease detection stage, where the hidden neurons and epochs were optimized in the LSTM network and the activation functions and hidden neurons were optimized in the ResNet structure to maximize the accuracy of the developed model. The developed ABC-ResNet-LSTM-based tomato leaf spot disease detection approach accomplished with 3.52%, 3.88%, 3.52%, and 1.72% improved sensitivity than the CNN-ABC-ResNet-LSTM, ResNet-ABC-ResNet-LSTM, and LSTM-ABC-ResNet-LSTM conventional models. The effectiveness of the system is very high when compared to the existing tomato leaf spot disease detection methodologies.

Table 2 Comparative analysis of presented tomato leaf spot disease detection method among various detection approaches using dataset 1

Computation measures	CNN-ABC-ResNet-LSTM [21]	RESNET-ABC-ResNet-LSTM [22]	LSTM-ABC-ResNet-LSTM [23]	RESNET-LSTM-ABC-ResNet-LSTM [24]	ResNet-ABC-ResNet-LSTM
F1-score	74.3931	75.725	76.1525	81.7559	86.4134
Sensitivity	91.0481	91.5682	91.773	94.0741	95.7289
MCC	70.9255	72.4198	72.9044	79.2123	84.4519
FNR	08.9519	8.4318	8.227	5.9259	4.2711
NPV	98.3877	98.4893	98.5282	98.9601	99.2616
Precision	62.8892	64.5556	65.076	72.2902	78.7502
Specificity	91.0454	91.6207	91.7914	93.99	95.6948
FPR	8.9546	8.3793	8.2086	6.01	4.3052
Accuracy	91.0458	91.6132	91.7888	94.002	95.6997
FDR	37.1108	35.4444	34.924	27.7098	21.2498

Table 3 Comparative analysis of presented tomato leaf spot disease detection method among various detection approaches using dataset 2

Computation measures	CNN-ABC-ResNet-LSTM [21]	RESNET-ABC-ResNet-LSTM [22]	LSTM-ABC-ResNet-LSTM [23]	RESNET-LSTM-ABC-ResNet-LSTM [24]	ResNet-ABC-ResNet-LSTM
F1-score	73.9583	74.4737	75.3316	78.7466	82.5843
Sensitivity	90.1587	89.8413	90.1587	91.746	93.3333
MCC	69.3418	69.8799	70.8826	74.9079	79.4069
FNR	9.8413	10.1587	9.8413	8.254	6.6667
NPV	97.8427	97.7855	97.8635	98.2325	98.5934
Precision	62.6932	63.5955	64.6925	68.9737	74.0554
Specificity	89.2698	89.7143	90.1587	91.746	93.4603
FPR	10.7302	10.2857	9.8413	8.254	6.5397
Accuracy	89.418	89.7354	90.1587	91.746	93.4392
FDR	37.3068	36.4045	35.3075	31.0263	25.9446

Table 4 Estimation of computation time for the designed tomato leaf spot disease detection method

Classifiers	CNN-ABC-ResNet-LSTM [21]	RESNET-ABC-ResNet-LSTM [22]	LSTM-ABC-ResNet-LSTM [23]	RESNET-LSTM-ABC-ResNet-LSTM [24]	ResNet-ABC-ResNet-LSTM
Dataset 1					
Time (sec)	58.140	52.319	58.441	63.140	49.578
Dataset 2					
Time (sec)	23.185	24.685	21.373	28.335	16.515

Acknowledgements I acknowledge the "KLEF–2012030001" Full-Time fellowship awarded for the successful completion of the work.

References

1. Kodamana H, Chakraborty S, Chakraborty S (2022) Deep learning aided automatic and reliable detection of tomato begomovirus infections in plants. J Plant Biochem Biotechnol 31:573–580
2. Pan J, Tian X, Meng X, Wu Q, Chen Y (2022) Identification of tomato leaf diseases based on a deep neuro-fuzzy network. J Inst Eng (India): Series 103:695–706
3. Ahmed S, Hasan MB, Ahmed T, Sony MRK, Kabir MH (2022) Less is more: lighter and faster deep neural architecture for tomato leaf disease classification. IEEE Access 10:68868–68884
4. Chen G, Yang G, He Y, Yan Z, Guo Y, Ding J (2020) Self-supervised collaborative multi-network for fine-grained visual categorization of tomato diseases. IEEE Access 8:211912–211923
5. Zhou C, Zhou S, Xing J, Song J (2021) Tomato leaf disease identification by restructured deep residual dense network. IEEE Access 9:28822–28831
6. Indu VT, Priyadharsini SS (2022) Crossover-based wind-driven optimized convolutional neural network model for tomato leaf disease classification. J Plant Dis Prot 129:559–578
7. Basavaiah J, Anthony AA (2020) Tomato leaf disease classification using multiple feature extraction techniques. Wireless Pers Commun 115:633–651
8. Kaur P, Harnal S, Gautam V, Singh MP, Singh SP (2022) A novel transfer deep learning method for detection and classification of plant leaf disease. J Ambient Intell Humanized Comp
9. Moussafir M, Chaibi H, Saadane R, Chehri A, El Rharras A, Jeon G (2022) Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks. Plant Soil
10. Zhang Y, Chen M (2022) An IoT-enabled energy-efficient approach for the detection of leaf curl disease in tomato crops. Wireless Net
11. Djimeli-Tsajio AB, Thierry N, Jean-Pierre LT, Kapche TF, Nagabhushan P (2022) Improved detection and identification approach in tomato leaf disease using transformation and combination of transfer learning features. J Plant Dis Prot 129:665–674
12. Russel NS, Selvaraj A (2022) Leaf species and disease classification using multiscale parallel deep CNN architecture. Neural Comp Appl
13. Gadekallu TR, Rajput DS, Reddy MPK, Lakshmanna K, Bhattacharya S, Singh S, Jolfaei A, Alazab M (2021) A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. J Real-Time Image Proc 18:1383–1396
14. Lu J, Zhou M, Gao Y, Jiang H (2018) Using hyperspectral imaging to discriminate yellow leaf curl disease in tomato leaves. Precision Agricult 19:379–394
15. Gajjar R, Gajjar N, Thakor VJ, Patel NP, Ruparelia S (2022) Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. Vis Comput 38:2923–2938
16. Xie C, Shao Y, Li X, He Y (2015) Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging. Scient Reports 5:16564
17. Nandhini S, Ashokkumar K (2021) Improved crossover based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network. Multimed Tools Appl 80:18583–18610
18. Thangaraj R, Anandamurugan S, Kaliappan VK (2021) Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. J Plant Dis Prot 128:73–86
19. Wang H, Sanchez-Molina JA, Li M, Berenguel M (2020) Development of an empirical tomato crop disease model: a case study on gray leaf spot. Eur J Plant Pathol 156:477–490

20. Karaboga D, Basturk B (2008) On the performance of Artificial Bee Colony (ABC) algorithm. Appl Soft Comput 8(1):687–697
21. Kannan EN, Kaushik M, Prakash P, Ajay R, Veni S (2020) Tomato leaf disease detection using convolutional neural network with data augmentation. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES)
22. Kaur M, Bhatia R (2019) Development of an improved tomato leaf disease detection and classification method. In: 2019 IEEE Conference on Information and Communication Technology
23. Naeem H, Bin-Salem A (2021) A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images. Appl Soft Comp 113:107918
24. Rana S, Rahman R, Sagor MK, Hasan M (2021) Tomato leaf disease detection using customized transfer learning architectures and LSTM. AI Research Project
25. Chowdhury MEH, Rahman T, Khandakar A, Ayari MA, Khan AU, Khan MS, Al-Emadi N, Reaz MBI, Islam MT, Ali SHM (2021) Automatic and reliable leaf disease detection using deep learning techniques. Agri Engineering 3(2):294–312
26. Tabjula JL, Kanakambaran S, Kalyani S, Rajagopal P, Srinivasan B (2021) Outlier analysis for defect detection using sparse sampling in guided wave structural health monitoring. Struct Cont Heal Monit 28(3)
27. Tabjula J, Kalyani S, Rajagopal P, Srinivasan B (2022) Statistics-based baseline-free approach for rapid inspection of delamination in composite structures using ultrasonic guided waves. Struct Heal Monit 21(6)

The Development of Advanced Deep Learning-Based EoR Signal Separation Techniques

S. Pradeep, C. V. P. R. Prasad, and Ch Ruchitha

Abstract The weak EoR signal is submerged in the strong foreground radiation interference. The classic the foreground removal methods assume that the foreground spectrum is smooth, but the complex instrumental effect will affect the spectral structure, resulting in the failure to accurately detect the signal. In this paper, a deep learning network called CNN-LSTM model is proposed to separate the EoR signal to improve computing resource utilization for massive observational data. Based on the simulation data of SKA1-LOW, a CNN-LSTM fusion model is constructed to reconstruct the EoR signal. Experimental results show that compared with the traditional methods including polynomial fitting and continuous wavelet transform, the EoR signals detected by the proposed deep learning model have better quantitative evaluation indexes of SNR and Pearson correlation coefficient. This property provides a new way to explore the research field of EoR.

Keywords OR signal · Deep learning · Signal separation algorithm

1 Introduction

The 21-cm neutral hydrogen line records the evolution stage of the universe from neutral hydrogen filled to ionized hydrogen. Studying the cosmic EoR signals of great significance for exploring the epoch of EoR (EoR) and the formation of the first generation of celestial bodies in the evolution of the universe. The radiation intensity of EoR signal is lower than 10 mK, while the brightness temperature of foreground radiation signal is about 4–5 orders of magnitude higher, and the power SNR reaches

S. Pradeep (✉) · C. V. P. R. Prasad
Department of CSE, Malla Reddy Engineering College for Women (UGC AUTONOMOUS), Maisammaguda, Hyderabad, Telangana, India
e-mail: pradeep.sunkari87@gmail.com

C. Ruchitha
Department of IT, Malla Reddy Engineering College for Women (UGC AUTONOMOUS), Maisammaguda, Hyderabad, Telangana, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_4

about -50 dB. The weak EoR signal is submerged in the strong foreground radiation interference, containing a large amount of noise caused by instrument effect. It indicates that accurate identification and separation of foreground interference is the key to solve the problem of detecting EoR signal in image space and power spectrum space.

At present, some methods can be used to detect the EoR signal. The traditional EoR signal separation methods mainly include foreground subtraction and foreground avoidance. The foreground subtraction method is to separate the EoR information from the strong foreground disturbance in image space or Fourier space by using the different characteristics of foreground contamination and EoR information displayed on the spectrum. Based on such ideas, V. Jellić et al. constructed a parametric model to describe the spectral variation of the foreground, so as to fit and deduct the pollution in the simulated radio sky map, which has a significant effect on removing the large-scale foreground interference [1]. After that, Adrian Liu used weighted polynomials to fit the information content of each pixel. On the basis of maintaining the original effect, Adrian Liu processed the small-scale foreground components more effectively [2]. Anna Bonaldi applied the correlation component analysis (CCA) method to the simulated data of square kilometer array, effectively clearing the foreground and noise in the observation data within the exploration frequency range (100–200 MHz) [3]. However, the selection of fitting parameters makes the experimental results fluctuate greatly, and both over fitting and under fitting will cause the real features of the foreground to be unable to be accurately captured, thus affecting the detection effect of the EoR signal [3]. The Wp smoothing method proposed by Harker [4] avoids the fitting residuals being polluted by the power leakage in the foreground and makes full use of spectral features to achieve separation. Gu uses continuous wavelet transform to greatly reduce the calculation amount on the basis of ensuring the same reconstruction effect as Wp smoothing method [5].

Such methods require a high degree of smoothness in the foreground radiation spectrum to clearly distinguish the reionized signal. In many cases, the foreground deduction method may mistake some large-scale EoR signals as pollution and deduct them, which makes the fitted signals differ greatly from the input signals. Ian H assumed that the foreground interference existed in a region of the two-dimensional power spectrum and processed its power spectrum to avoid the foreground contaminated area to extract EoR signals. This foreground avoidance method reduced the deviation of the foreground deduction method, but it would cause the information of the observed data outside the EoR window to be ignored in the processing [6]. The work of Patil et al. discussed the influence of various factors on the EoR power spectrum by calibrating residual sidelobe noise, nonlinear effects, and gain errors caused by noise and ionosphere at the baseline [7].

In addition, the processing of massive data collected by large interferometric arrays such as SKA1-LOW is also a key bottleneck [8]. With high sensitivity and resolution, the equipment can detect signals with complex morphology and extremely high radiation intensity range, which provides more observation information for the detection of EoR signals.

Literature [9–11] shows its outstanding performance in galaxy cluster identification and classification, pulsar search, and other aspects. These applications indicate that deep learning has the potential to extract weak signals from highly time-varying data and can be used to solve the problem of EoR signal separation.

In this paper, with the aid of OSKAR [https://github.com/OxfordSKA/OSKAR] and WSClean [https://sourceforge.net/projects/wsclean/] simulation with instrument effect data radio sky, the deep learning algorithm is used to construct the CNN-LSTM fusion model to separate the reionized signal from the complex foreground pollution, and it is compared with the traditional polynomial fitting and continuous wavelet transform method, which provides a new idea for detecting the reionized signal.

1.1 EoR Signal Separation Approach

At present, a number of methods have been proposed to try to separate EoR signals from foreground interference, including the traditional foreground deduction method, foreground avoidance method, and the newly developed crossover study of deep learning and EoR signal. In the study of EoR signals, the most classical methods are polynomial fitting [12] and continuous wavelet transform [5]. In fact, the separation algorithm has different treatment effects on the pollution components with different physical sources and morphological characteristics.

1.2 Foreground Removal Method

Di Matteo first uses the angle fluctuation of the 21-cm spectral line to separate the foreground contamination, but the effective signal is submerged by various foreground radiation and difficult to separate. Later, Zaldarriaga et al. shifted their focus to the frequency correlation of foreground and used the mutual correlation as deduction, which evolved into the line of sight (LOS) method [13], that is, in image space or UV space (Fourier space). Based on the distinct spectral features of foreground radiation and EoR signal, smooth foreground components are identified and subtracted for each pixel along the frequency dimension (i.e., line of sight direction). In the following research, Chapman et al. [14] divided it into the method that adopts functional form (parameterized) for foreground signal and the method that slightly liberates this form (non-parameterized), aiming to find the form of smooth foreground function along frequency for each line of sight and subtract it from the total signal. Leaving behind noise, fitting errors, and residuals of the cosmological signal.

2 Polynomial Fitting

Early parametric methods used polynomials to fit directly to the measured data in the frequency or logarithmic frequency domain. The usual way of polynomial fitting is to fit the total observed spectrum along the line of sight with a smooth function, such as a polynomial of order N:

$$\log T_{b,\mathrm{fg}}(v) = a_0 + \sum_{i=1}^{n} a^i \log v^i \tag{1}$$

The order n is set as 2, and Jelic adjusts the order n to 3 and discusses the influence of polynomial order on the fitting effect. If the order of polynomial is too small, the foreground fitting will be insufficient, and the fitting residuals will affect the EoR signal. If the order of the polynomial is too large, the EoR signal is over fitted.

After stripping the radio source with very large brightness temperature in the frequency domain space, the foreground interference is deducted along the line of sight according to the characteristic that the foreground component has smoother spectrum than the EoR signal. In other words, the logarithm of the total signal intensity I_{total} of the observed frequency received by pixel V_i in the line of sight direction gives the function.

$$Y_i = \lg(I_{\mathrm{Total}}(V_i)) \tag{2}$$

The intensity of foreground component was simulated by multi-order polynomial fitting to the logarithmic signal I_{fore},

$$I_{\mathrm{fore}}(V_i) \approx a_0 + a_1 \lg(v_i) + a_2 \lg^2(v_i) \tag{3}$$

EoR signal radiation and instrument noise can be expressed as the difference between the overall radiation intensity and the fitted polynomial signal intensity.

According to the different angular power spectrum of the signal, the reconstructed signal is obtained by subtracting the instrument noise $I_{\mathrm{EoR}\prime}$

$$I_g(I_{\mathrm{Total}}(V_i)) \approx a_0 + a_1 \lg(v_i) + a_2 \lg^2(v_i) + I_{\mathrm{EOR}}\left(v^i\right) \tag{4}$$

$$I_{\mathrm{EOR}\prime}(v_i) \approx I_{\mathrm{Total}}(v_i) - 10^{a_0+a1} \lg(v_i) + a_2 \lg^2(v_i) \tag{5}$$

Fitting the foreground signal with various curve functions will have different influences on the extraction of EoR signal, such as exponential function, Fourier function, Gaussian function, and power exponential function. Chapman et al. showed in their research work that the EoR signal extracted by linear fitting the foreground signal is closer to the simulated signal [14]. In the work of Bonaldi [15]. and Brown, the idea of

polynomial fitting was introduced into Fourier domain correlation component analysis, and second-order statistics were used to estimate the spectrum of foreground components from data [13]. For this study, we will adopt the conclusion of Jelic et al. and use third-order polynomials to fit in log space.

2.1 Continuous Wavelet Transform

Non-parametric methods do not directly assume that foreground spectra should conform to a particular parametric model, or any spatial structure about them, but make full use of the different spectral characteristics of foreground radiation and EoR signals to achieve the separation of the two. For example, typical methods are Wp smoothing, independent component analysis, ICA, generalized morphological component analysis (GMCA), and continuous wavelet transform (CWT).

In this paper, we use the continuous wavelet transform method outlined by Harker et al., and according to the inverse CWT definition proposed by Daubechies [16],

$$h(t) = \frac{2}{C_\psi} \int_0^\infty \left[\int_{-\infty}^{\infty} W_{x,\psi}(\tau, s) \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-\tau}{s} \right) \mathrm{d}\tau \right] \frac{\mathrm{d}s}{s^2} \tag{6}$$

where $h(t)$ represents the real space signal to be transformed, $\psi(t)$ called the mother wavelet function, and τ, s represents the real space and scale exponent of the wavelet coefficient $W_{x,\psi}$. Prospect in the study of EoR, radiation, and EoR signal distribution of significant coefficients is different, and the significant coefficients of smooth prospects mainly depend on the data boundary discontinuity,

$$\psi_s(t) = \frac{\psi\left(\frac{t}{s}\right)}{\sqrt{|s|}} \tag{7}$$

where $T_d(v)$ represents the wavelet coefficient of the total radiation signal, δ_c is the Dirac delta function, and $v_{\min}$ represents the lower frequency limit. The wavelet coefficients of the total signal are filtered through a mask. Since the wavelet transform is actually the cross correlation between $\psi_s(t)$ and the real space signal $h(t)$, according to the cross correlation theorem, the real space signal can be effectively calculated in the Fourier space, and this equation is used to reconstruct the filtered EoR signal.

$$h(t) = \frac{2}{C_\psi} \int_0^\infty F^{-1}\left\{ F\left\{ W_{x,\psi} \right\} \cdot \psi_s \right\}(t, s) \frac{\mathrm{d}s}{s^2}. \tag{8}$$

Harker introduced the Wp smoothing method into the EoR detection experiment and used the physical expected values of foreground components for separation. This method uses a smoothing function to fit a set of observations of a frequency channel. Machler first studied the boundary value problem to give a smooth function formula;

Gu et al. further studied the problem and gave a new algorithm. Although a general smoothness of the foreground is assumed to make the method a good application, the method includes a smoothness parameter that allows the user to control the harshness of this smoothness condition to allow for deviations from the prior smoothness.

In addition, Chapman et al. applied FastICA to EoR simulations as an independent component analysis technique, which assumes that foreground components are statistically independent in order to model them. Emma Chapman et al., using the component separation method, generalized morphological component analysis (GMCA) [14] has successfully conducted separation experiments on simulated LOFAR data.

2.2 Foreground Avoidance Method

The advantage of foreground deduction method is that it can retain all the information of EoR signal, but the disadvantage is that it may not be able to deduct foreground pollution accurately or some large-scale EoR signals are mistaken for foreground and deducted, resulting in a certain degree of deviation in the results. In order to effectively avoid the possible deviation caused by foreground pollution on EoR detection results, foreground avoidance method has been paid more attention and studied in recent years.

In the study of Chapman et al. [17], it is shown that by deducting the foreground by modeling or by analyzing the specific part of the EoR delay power spectrum that is not affected by the foreground, there is an "EoR window" in the two-dimensional power spectrum theoretically because most of the instrumental color effect is confined to the wedge. The effectiveness of this method mainly depends on the instrument response and the smoothness of the observed sky [18].

Liu et al. proposed a mathematical formalism for describing wedges [19] that allows maximizing the range of the EoR window. Several methods have also been developed to estimate the covariance of prospects, which can then be incorporated into the power spectrum estimator. However, these foreground avoidance or suppression methods have the disadvantage of substantially reducing the sensitivity of the instrument, as they reduce the number of detectable patterns. Moreover, Rajesh Mondal et al. [20] Future observation studies using the upcoming SKA-LOW measure the prospect of EoR signals and in this work quantitatively addresses the impact of prospects on map delectability predictions by avoiding signals contained within the planar foreground wedge.

2.3 Deep Learning Method

In recent years, some literatures have explored the idea of foreground removal based on machine learning, and several attempts have been made to improve the performance and accuracy of neural networks for the task of detecting EoR signals, with

varying degrees of success. Samuel Gagnon-Hartman et al. described and tested the implementation of a U-Net architecture [21] with the aim of using information from Fourier patterns to identify ionized regions in wedge-filtered images and to be able to reconstruct their shape, size, and location in the image. In the work of Li et al. [22].

In addition, Shimabukuro et al. [23] introduced artificial neural networks (ANNs) to constrain astrophysical parameters from the 21-cm power spectrum to rigorously estimate the 21-cm power spectrum from the interferogram while mitigating foreground contamination, thus improving sensitivity. In La Plante's work, the convolution neural network (CNN) is used to further constrain the cosmological parameters during the EoR period. By using several supervised learning methods to improve prediction accuracy, including neural networks, kernel regression, or ridge regression, this work compares the performance of these methods using a large training set of SKA simulated 21-cm power spectra.

In contrast, in our work, in order to study the EoR signal separation problem, we treat foreground radiation as strong noise and EoR signal as effective signal, so the problem is transformed into the identification problem of EoR signal: By removing the noise (i.e., foreground radiation) in the total radiation signal (i.e., the sum of EoR signal and foreground radiation), the signal (i.e., EoR signal) is obtained, and the separation of EoR signal and foreground radiation is realized.

3 Methods

3.1 CNN BiLSTM Network Architecture

We built a neural network to eliminate the foreground radiation seen in the instrument beam pattern and separate the weak EoR signal. The effect will be demonstrated with the help of a radio sky simulating SKA observations. According to the astrophysical theory, we preprocessed the OSKAR simulated visibility data to exclude the influence of instrument effect as much as possible and then standardized the data.

3.2 Classic Convolutional Neural Network

The essence of the problem of detecting the EoR signal is to separate the foreground contamination and extract the EoR signal from the total radiation signal. One-dimensional convolutional neural network (CNN) can extract a variety of abstract features from data and has a good performance effect in the separation of mono sound sources. The core of neural network is to extract features layer by layer through convolution operation,

$$X_j^{(l)} = f\left(\sum_{i \in M_j} X_j^{(l-1)} * K_{ij}^{(b)} + b_j^{(b)}\right) \tag{9}$$

where l the number of is convolutional layers; $X_j^{(l)}$ is the first feature map of the layer, $K_{ij}^{(b)}$ is the convolution kernel, and b is the bias term. $*$ stands for convolution operator.

3.2.1 Long Short-Term Memory

The bidirectional long short-term memory network (LSTM) derived from recurrent neural network (RNN) has a strong ability to extract weak features from highly time-varying data. This model mainly adds three gating structures of hidden layers to the original recurrent neural network. The calculation method is as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{10}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{11}$$

$$\widetilde{C}_t = \tan h\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \tag{12}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \tag{13}$$

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{14}$$

$$h_t = o_t * \tan h(C_t) \tag{15}$$

The parameters in the above formula are updated through the internal unit structure of LSTM, where x_t represents the input value at the current unit time, $h_{t\text{-}1}$ represents the output value at the previous unit time, $\widetilde{C}$ represents the state information of candidate unit, and C_t represents the internal state information of current unit. The input gate i determines the part of the input information that can be reserved for C_t; the forgetting gate f decides to save the part of the unit state C_{t-1} at the previous moment according to the data feature information of the attention feature; the output gate o determines the part of the hidden state at the moment $t-1$ that can be transferred to the current state. Its unit structure is shown in Fig. 1.

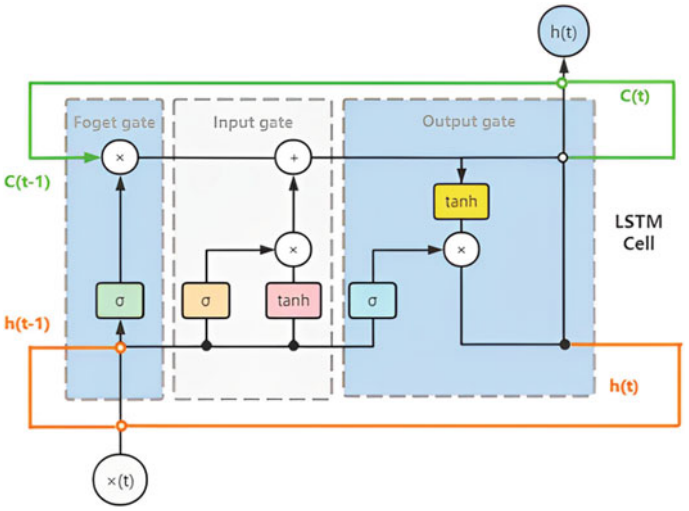


Fig. 1 LSTM internal structure diagram

3.2.2 Fusion Module

The analysis can predict that the two networks have the ability to process astronomical data and separate EoR signals. Since the signal-to-noise ratio of the signal to be measured is very low, we take advantage of the characteristics of CNN and LSTM and combine the advantages of the two frameworks to construct a new CNN-BiLSTM network.

In the test phase, we constructed two kinds of fusion deep neural networks, which were vertically and horizontally combined with CNN and LSTM, respectively, and finally selected the vertical cascade structure with better performance as the model basis. In this model, LSTM receives the output of CNN and extracts features in a deeper way. The architecture of the CNN-BiLSTM model is shown in Fig. 2, followed by a detailed overview of the framework.

The local characteristics of the input data used in this paper, the model could reflect the key information of phase spectrum change point, convolution neural network (CNN) with multiple local feature extraction of convolution layer to the data, because each convolution filter layer contains a fixed size, and quantity greatly reduced, in the use of the limited weight of back propagation algorithm to update the parameters, After data preprocessing, each pixel in the observation area is listed in the dimension of observation frequency as the input of three one-dimensional convolutional layers, in which the first convolutional layer contains 32 filters of size 3×1, and the second and third convolutional layers contain 16 filters of size 3×1. The eigenvector is obtained by operation according to the equation v_i,

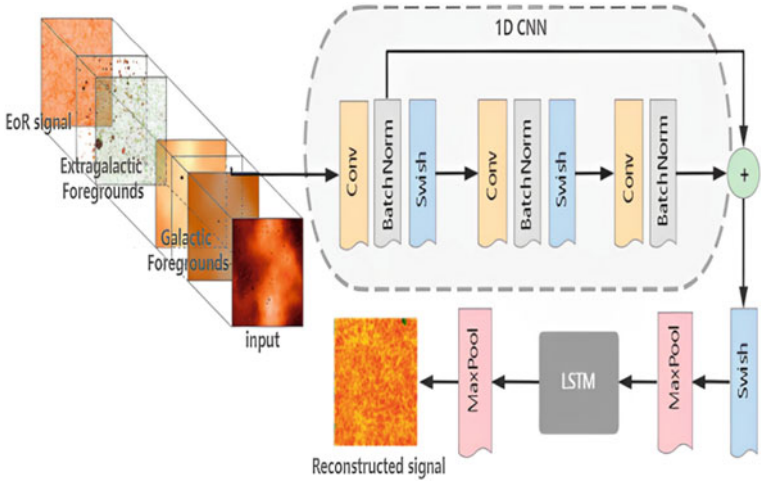


Fig. 2 Network architecture diagram

$$v_i^{(l)} = \phi^{(l)}\left(\sum_{j=1}^{m_{l-1}} v_j^{(l-1)} * W_i^{(l)} + b_i^{(l)}\right), \quad i = 1, 2, \ldots, m_1 \tag{16}$$

where l represents the sequence number of convolutional layer, m_1 represents the total number of filters in layer l, φ represents the activation function of this layer, $W_i^{(l)}$ and $b_i^{(l)}$, respectively, represent the weight parameter and bias term of layer l, and set $\left\{v_i^{(l)}\right\}$; $i = 1, 2, \ldots, m_1$ represents the sum of feature vectors obtained from all filters in layer l.

3.2.3 Activation Function

By comparing and testing the effects of different activation functions, Swish function is finally determined as the activation function of network convolutional layer.

$$f(x) = x * \sigma(\beta x) \tag{17}$$

where σ stands for sigmoid function. Sigmoid function has saturation property, and introducing β can train parameters to avoid gradient disappearance. When β takes different values, the function image is as follows (Fig. 3).

Swish function is essentially a smooth function between linear function and ReLU function. Since the sparse processing forced by ReLU will reduce the effective capacity of the model, when the gradient is 0 when $x < 0$, it often leads to the problem of neuron death. The Swish function can correct this problem and adjust

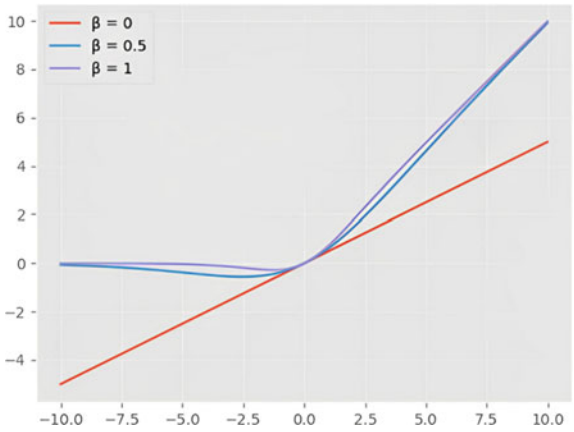


Fig. 3 Swish function curve

the training parameters suitable for the model so that the network can achieve higher performance than the standard ReLU.

After maximum pooling, the feature vectors are passed through the LSTM layer containing hidden units, and the state activation function of TANH and the gate activation function of sigmoid are used. In addition, we add an appropriate dropout layer to the CNN model to discard 20% of random features and weaken the joint adaptability between neuronal nodes, thus avoiding overtraining of LSTM sequence data. At the same time, Adam algorithm is used to optimize the objective function to further improve the performance of the model.

3.3 Model Training

The CNN-LSTM network architecture is trained and tested. Firstly, uniform initialization was used to set all parameters of the network to random values. The initial learning rate was set as 0.003, and the batch size was set as 125. The total radiation spectrum data of pixels were input into the network model one by one for convolution operation. After 100 rounds of iterative training, EoR signals were separated from the network model. The loss function used to guide parameter tuning has been tested and decided to quantify the difference between the EoR signal obtained from the model and the EoR signal input to the network using cross-entropy, which is defined as

$$C = -\frac{1}{n}\sum_{x}[y \ln a + (1-y)\ln(1-a)] \tag{18}$$

In formula (19), x represents the sample, y represents the actual label, a represents the predicted output, and n represents the total number of samples. This loss function

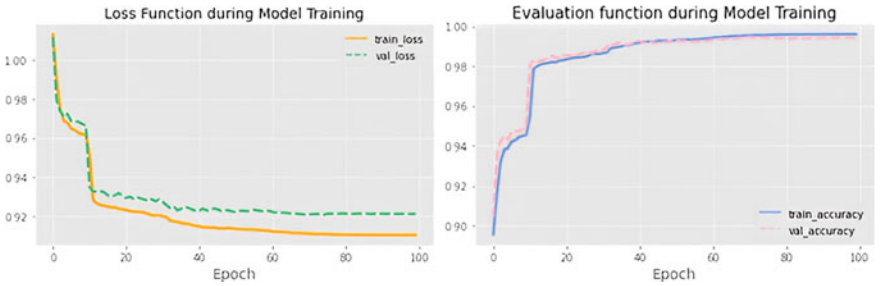


Fig. 4 Model training process: curve of loss function (left panel) and Evaluation function (right panel)

avoids the problem that the weight of the square loss function is updated too slowly. When the error is large, the weight update speed is improved, and when the error is small, the weight update speed is slowed down, so that the parameter update strategy can be adjusted in time. At the same time, the evaluation function is set, and the accuracy is calculated to evaluate the performance of the current training model.

Figure 4 shows the changes of the loss function and evaluation function with the training process, taking the training period as the variable. The final network accuracy is 0.972, indicating that the network has good performance.

4 Experiments

4.1 Datasets

The observed data used in the experiment are from the OSKAR simulation data based on the SKA1-LOW site layout and the EoR simulation data published by the Evolution of 21 cm Structure project. Taking (R.A.,Dec.) = (3°,−27°) as the pointing center, the first image cube containing foreground and EoR signals is simulated in the frequency band range of 154–162 MHz. For the EoR signal, the data should be standardized after eliminating the outliers. We divided 70% of the data points in the first set of image cubes into training sets, and the remaining pixels were used as validation. All the data in the second image cube are used as the test set to test the network performance and reconstruct the complete image of EoR signal. Figure 5 shows the entire framework of the preprocessing process.

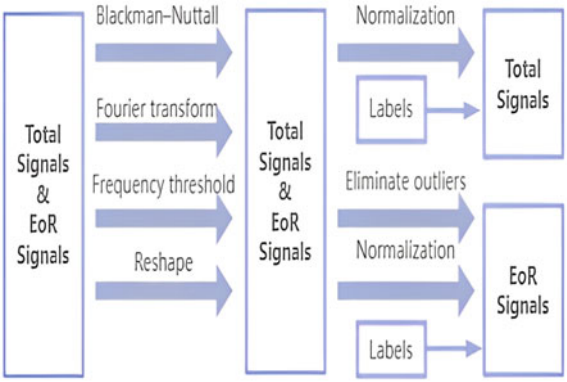


Fig. 5 Data preprocessing pipeline

4.2 Model Performance

In the study of EoR, the CLEAN algorithm has limitations on the processing effect of faint diffuse radiation, so we took advantage of the "Faintgalaxies" Lightcone image cube to generate sky maps by WSClean. As shown in Fig. 6. The EoR image shows the density distribution corresponding to the red shift, while in the superposed low-frequency sky, the EoR signal with relatively low bright temperature is completely covered by the intense foreground radiation.

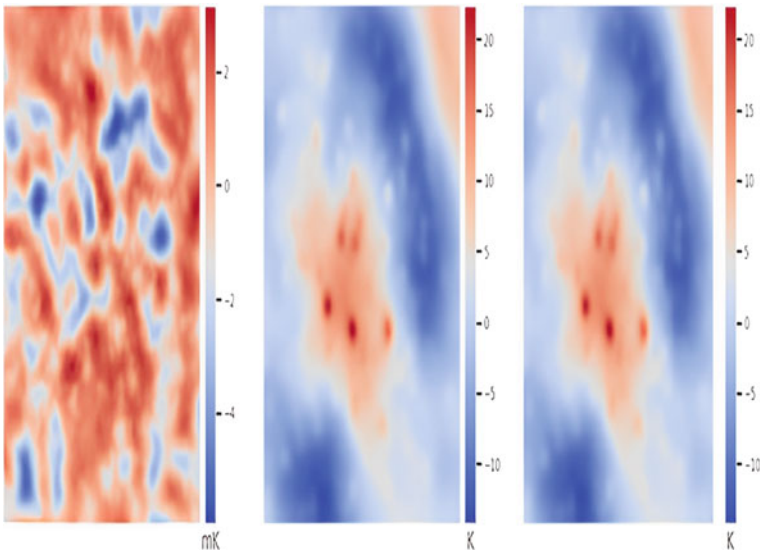


Fig. 6 Simulated images of the EoR signal (left panel), the foreground emission (middle panel), and sky map (right panel) at 160 MHz

The EoR signal after model processing was imaged, and the difference of EoR signal before and after reconstruction was compared. In Fig. 7, for detecting signals and reconstruction, overall space structure and the outline is almost the same, choose the same color display standardized after radiation intensity distribution. This is because there are fine small-scale ripples structures (small ripples) in the foreground Fig. 8, there are narrow bands in the power spectrum, making the fitting effect deviation, optimize the model to get a more accurate fitting effect is our next improvement direction.

Figure 9 shows the variation curve of the magnetic field intensity of the received foreground radiation signal, EoR signal, and total signal simulated by a randomly selected sky pixel as a function of frequency. It can be seen that the EoR signal presents a sawtooth spectral structure, but there is a gap of 4–5 orders of magnitude

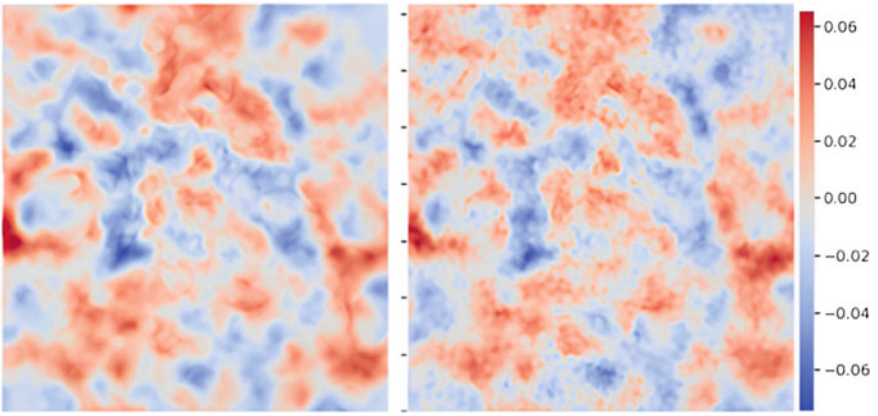


Fig. 7 Comparison between reconstructed EoR signal (left panel) and reconstructed EoR signal (right panel)

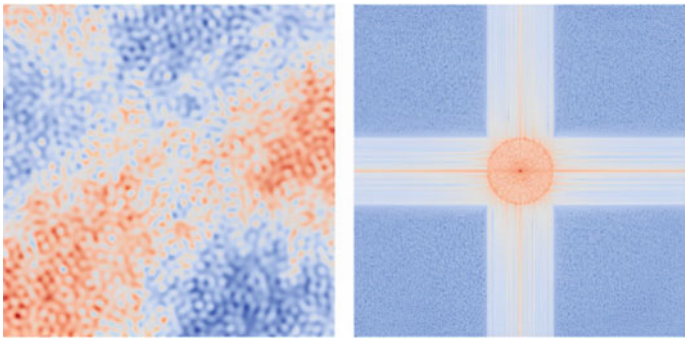


Fig. 8 Small ripples of original foreground signal (left panel) and bright stripe in the 2D power spectrum (right panel)

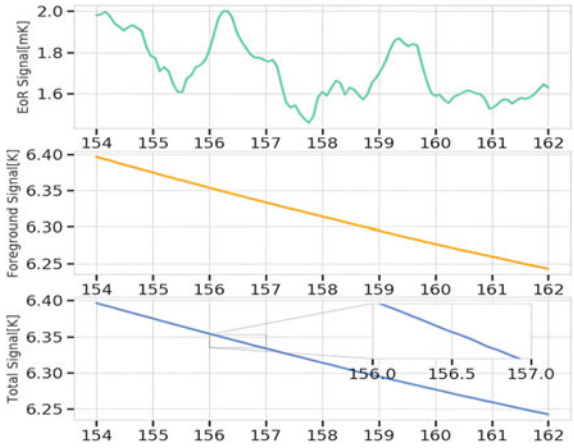


Fig. 9 EoR radiation signal foreground radiation signaling on total radiation signal

between its brightness temperature and that of foreground component. The EoR signal cannot be visually distinguished in the total signal.

4.3 Result Comparison

In order to verify the difference between the proposed method and the traditional method, qualitative and quantitative comparisons of the effects of the two evaluation methods were used to compare the separation of EoR signals.

A. *Spectral Analysis*

In the observed frequency band, the spectrum characteristics of the reionized signal obtained by the continuous wavelet transform method, polynomial fitting, and deep learning model are compared with the original signal. In order to emphasize the change effect, the radiation signal is normalized. Can be seen from the Figs. 10, 11, and 12 model is presented by this paper get the signal is compared with the other two traditional methods to get more close to the input signals of ionization signal, again shows that under the complex beam effect can still treat signal fluctuation of smaller amplitude and scale of reconstruction, get more complete ionization signal spectrum structure again.

B. *Quantitative Analysis*

In order to quantify the similarity between the reconstructed reionized signal and the input signal, Pearson correlation coefficient and the signal-to-noise ratio (R_{sn}) of the separated signal and the original signal are introduced as quantitative analysis indicators to measure the reconstruction effect, so as to better reflect the numerical differences of the results obtained by different methods. Combined with the understanding of the signal magnetic field strength, the signal-to-noise ratio index (R_{sn})

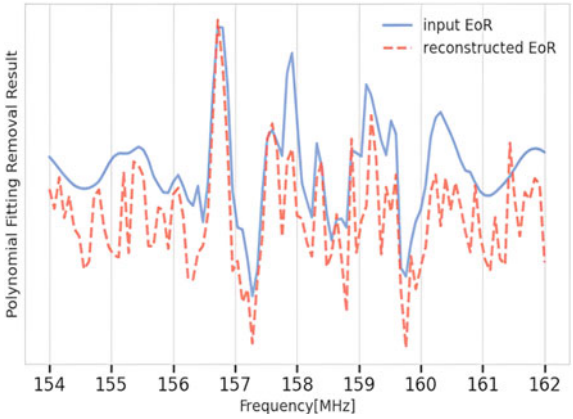


Fig. 10 Result of polynomial fitting method

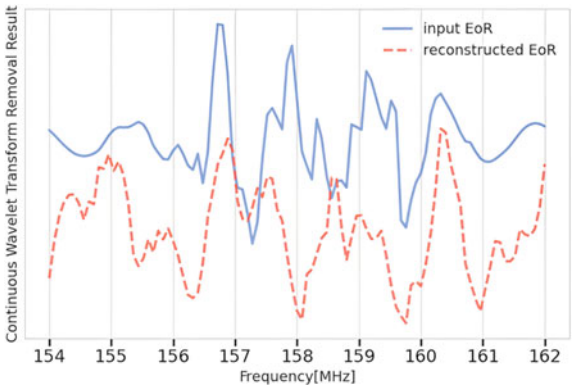


Fig. 11 Result of continuous wavelet transform method

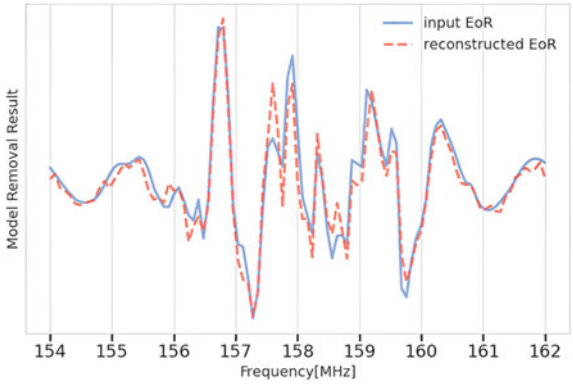


Fig. 12 Result of CNN-LSTM model

Table 1 Comparison of the performance of EoR signal reconstruction between traditional methods and deep learning method

Index	Rsn/dB	Pearson
Polynomial fitting method	9.352	0.477
Continuous wavelet transform method	13.340	0.316
CNN-LSTM model	17.231	0.813

and Pearson correlation coefficient were defined as follows

$$R_{\mathrm{sn}} = 10 \lg \frac{E(|r_{\mathrm{eor}}|)^2}{E(|x_{\mathrm{eor}} - r_{\mathrm{eor}}|)^2}, \tag{19}$$

$$(\mathbf{r}_{\mathrm{eor}}, \mathbf{x}_{\mathrm{eor}}) = \frac{\sum_{j=1}^{n} \left(r_{\mathrm{eor},j} - \overline{r}_{\mathrm{eor}}\right)\left(x_{\mathrm{eor},j} - \overline{x}_{\mathrm{eor}}\right)}{\sqrt{\sum_{j=1}^{n} \left(r_{\mathrm{eor},j} - \overline{r}_{\mathrm{eor}}\right)^2 \sum_{j=1}^{n} \left(x_{\mathrm{eor},j} - \overline{x}_{\mathrm{eor}}\right)^2}}. \tag{20}$$

where r_{eor} is reconstruction EoR signal, x_{eor} is original input EoR signal, and n is the length of the signal. Pearson's correlation coefficient is close to 1, indicating that the more accurate the reconstructed EoR signal is, the better the performance of the processing method is. Table 1 shows the statistical results of the SNR separation performance index, reflecting the different performance effects of polynomial fitting method with order 4 and continuous wavelet transform method and deep learning model for the same test data.

These results fully demonstrate that the trained deep learning model can effectively separate the spectrum of foreground radiation and EoR signals, so as to accurately detect EoR signals.

5 Conclusions

The existing EoR signal separation algorithms are not good for signal reconstruction when dealing with the foreground components with complex morphological changes in massive observation data. This paper first introduces the traditional polynomial fitting and wavelet transform method and then discusses the deep learning model from the theoretical aspect. After that, it is verified by experiments that the CNN-LSTM model is better than the traditional EoR signal separation algorithm in removing foreground components, and the SNR and Pearson correlation coefficient are used as the evaluation indexes of quantitative analysis to evaluate the separation results of different methods. The research can provide a new reference idea for EoR signal separation, which can save computing resources such as memory and time, and reduce the requirement of hardware. For further improving the reconstruction effect, there are still many key problems to be solved, such as the complex instrument effect

introduced in the observation and the size of the imaging sky area, which will affect the detection. In the next work, we will study the influence of wide-field imaging of SKA1-LOW on small-scale radiation celestial signals and explore the dispersion degree and magnetic field intensity changes of point signals at different phases far from the zenith. Since the experiment in this paper is conducted on simulated data, the processing effect of observation data from other telescopes needs to be further explored.

References

1. Jelić V, Zaroubi S, Labropoulos P et al (2010) Foreground simulations for the LOFAR—epoch of EoR Experiment. Mon Not R Astron Soc 389(3):1319–1335
2. Liu A, Tegmark ME, Bowman J et al (2009) An improved method for 21cm foreground removal. Mon Not R Astron Soc 398(1):401–406
3. Anna B, Brown ML (2015) Foreground removal for square kilometre array observations of the epoch of EoR with the correlated component analysis. Mon Not R Astron Soc 2:1973–1983
4. Harker G, Zaroubi S, Bernardi G et al (2009) Non-parametric foreground subtraction for 21-cm epoch of EoR experiments. Mon Not R Astron Soc 397(2):1138–1152
5. Gu J, Xu H, Wang J et al (2013) The application of continuous wavelet transform based foreground subtraction method in 21 cm sky surveys. Astrophys J 773(1):773–779
6. Ian H, Emma C, Pritchard JR et al (2020) Comparing foreground removal techniques for recovery of the LOFAR-EoR 21cm power spectrum. Mon Not R Astron Soc 2:2264–2277
7. Patil AH, Yatawatta S, Koopmans L et al (2017) Upper limits on the 21 cm epoch of EoR power spectrum from one night with LOFAR. Astrophys J 838(1):65–81
8. Eloy DLA, Trott CM, Wayth RB et al (2018) Spectral performance of SKALA antennas I: mitigating spectral artefacts in SKA1-LOW 21-cm cosmology experiments. Mon Not R Astron Soc 3:2662–2671
9. Lukic V, Brüggen M (2017) Galaxy classifications with deep learning. Proc Inter Astronom Union 12(S325):217–220
10. Burger B, Mattia V, Matthew P et al (2021) CNN architecture comparison for radio galaxy classification. Mon Not R Astron Soc 503(2):1828–1846
11. Lin H, Li X, Zeng Q (2020) Pulsar candidate sifting using multi-input convolution neural networks. Astrophys J 899(2):104–116
12. Wang X, Tegmark M, Santos MG et al (2006) 21 cm tomography with foregrounds. Astrophys J 650(2):529–539
13. Loeb A, Zaldarriaga M (2005) The small-scale power spectrum of cold dark matter. Physical Rev D 71(10):467–470
14. Chapman E, Abdalla FB, Harker G et al (2012) Foreground removal using fastica: a showcase of LOFAR-EoR. Mon Not R Astron Soc 423(3):2518–2532
15. Bonaldi A, Battye RA, Brown ML (2014) Cosmological constraints from Sunyaev-Zeldovich cluster counts: an approach to account for missing redshifts. Astrophys J 786(2):86–93
16. Ingrid D et al (1992) Sets of matrices all infinite products of which converge. Linear Alge Appl 161(15):227–263
17. Chapman BE, Harker A et al (2015) Cosmic dawn and epoch of EoR foreground removal with the SKA. Physics 9525(12):9009–9014
18. Parsons AR, Backer DC, Bradley RF et al (2010) The precision array for probing the epoch of re-ionization: eight station results. Astron J 139:1468–1480
19. Liu A, Parsons AR, Trott CM (2014) Epoch of EoR window. I. Mathematical formalism. Physical Rev, D. Particles, Fields, Gravitation, Cosmo 90(2):18–23

20. Mondal R, Bharadwaj S, Majumdar S (2016) Statistics of the epoch of EoR 21-cm signal–I. Power spectrum error-covariance. Monthly Notices Royal Astronomical Soc 456(2):1936–1947
21. Gagnon-Hartman S, Cui Y, Liu A et al (2021) Recovering the lost wedge modes in 21-cm foregrounds. arXiv e-prints, arXiv: 2102.08382
22. Li W, Xu H, Ma Z et al (2019) Separating the EoR signal with a convolutional denoising autoencoder: a deep-learning-based method. Mon Not R Astron Soc 485(2):2628–2637
23. Shimabukuro H, Ichiki K, Inoue S et al (2014) Probing small-scale cosmological fluctuations with the 21 cm forest: effects of neutrino mass, running spectral index and warm dark matter. Physical Rev D 90(8):83–86

Pediatric Pneumonia Diagnosis Using Cost-Sensitive Attention Models

J. Arun Prakash, C. R. Asswin, K. S. Dharshan Kumar, Avinash Dora, V. Sowmya, and Vinayakumar Ravi

Abstract Pediatric pneumonia is a medical condition in which air sacs of the lungs get filled with fluid. In recent years, chest X-rays have proved to be a better alternative to traditional diagnosis methods. Medical experts examine the chest X-ray images to detect the presence of pneumonia; however, the low radiation levels of X-rays in children have made the identification process more challenging leading to human-prone errors. The increasing use of computer-aided diagnosis in the medical field, especially deep learning architectures like Convolutional Neural Networks (CNNs) for images, helped tackle this issue. Our work proposes a Convolutional Block Attention Module (CBAM) attached to the end of pretrained ResNet152V2 and VGG19 with cost-sensitive learning. The weighted average ensemble uses weights which are calculated as a function of the precision, recall, f1-score, and AUC of each model. These values are concatenated as a vector and passed through a Tanh activation function. The sum of elements in this vector forms the weights. These weights when used in the weighted average classifier results in an accuracy of 96.79%, precision of 96.48%, recall of 98.46%, F1-score of 97.46%, and an AUC curve of 96.24% on the pediatric pneumonia dataset. The proposed architecture outperforms existing deep CNN models when trained with and without cost-sensitive training for the task at hand. We expect our proposed architecture to assist in real-time pediatric pneumonia diagnosis.

Keywords Pediatric pneumonia · Chest X-rays · Deep learning · Attention · Cost-sensitive learning · Weighted ensemble classifier

J. A. Prakash (✉) · C. R. Asswin · K. S. D. Kumar · A. Dora · V. Sowmya
Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: arun.jayakanthan@gmail.com

V. Sowmya
e-mail: v_sowmya@cb.amrita.edu

V. Ravi
Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia
e-mail: vravi@pmu.edu.sa

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_5

1 Introduction

Pneumonia is a lung infection that fills the air sacs in the lungs with liquids such as pus which can result in the infected person being unable to circulate enough oxygen into the bloodstream. The World Health Organization (WHO) stated in 2013 that pneumonia is one of the key contributing factors that led to the death of 6.3 million children under the age of 5. Globally, around 150 million cases of pneumonia in children below five years occur every year, of which 11–20 million require hospitalization. Pneumonia in children is facilitated by the short distance from the trachea to the bronchi, and the bronchiole increases the risk of transmitting the pathogens. Factors such as lack of awareness about the severity of the disease, low birth weight, lack of breastfeeding, malnutrition, and many other factors lead to such fatal infection of the respiratory tract in children. Child mortality and impairment are detrimental to the country's future economic growth. The primary reason for the prevalence of pneumonia in infants is the inaccessibility of quick and affordable pre-diagnostic tests. Chest X-Rays (CXRs) are some of the most cost-efficient ways of diagnosing pneumonia; nevertheless, detecting pneumonia in CXRs is still laborious. The loss of information due to usage of low radiation levels in children leads to false diagnosis.

The need for computer-aided diagnosis with the assistance of deep learning and attention mechanism is thus required, which will help improve robust pediatric pneumonia detection. Computer-aided diagnostic models have gained popularity in similar biomedical domains such as pneumonia [1–3], Alzheimer's [4, 5], tuberculosis [6], retinal disease [7], and hypoxia detection [8] with the advent of deep learning. This study discusses the impact of using an ensemble-based pipeline with spatial attention and channel attention module with cost-sensitive learning for pediatric pneumonia diagnosis. The pipeline for the proposed model is illustrated in Fig. 1.

2 Related Works

Deep learning architectures have replaced the manual feature extraction techniques that previously required specific filters. The majority of the initial research on pattern classification used MLP techniques. The disadvantage of this strategy was its inability to gather local knowledge. To address this problem, Convolutional Neural Networks

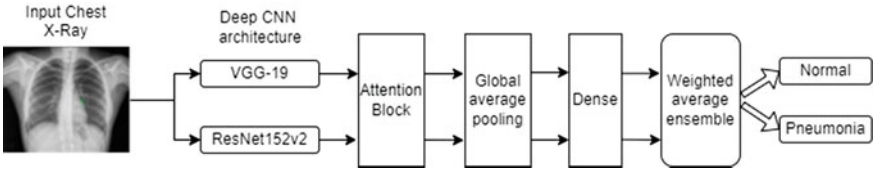


Fig. 1 Proposed pipeline for the classification of chest X-rays into normal and pneumonia

(CNNs) were developed. Kermany et al. [9] introduced the pediatric pneumonia dataset and proposed a transfer learning approach using neural networks. This trend of using transfer learning, which proved advantageous in better performance and faster training, led to researchers experimenting the use of transfer learning in various other deep CNN architectures. Several studies focused on the performance of MLP and CNN in the absence of transfer learning. A comparative study on MLP and CNN-based approaches concluded the advantages of using CNN over MLP [10]. However, the proposed model was trained from scratch, requiring much training time.

Several novel CNN architectures are proposed in [11, 12] but with the disadvantage of longer training time and limited performance. Transfer learning-based approaches showed better performance, still their performance was limited. The limitation was due to loss of spatial information with increasing convolutional layers in deeper CNN architectures. As a solution to this, Gaobo et al. [2] addressed using transfer learning and residual connections to prevent spatial information loss in feature extraction to improve performance.

Predefined weights are a significant determinant of a model's performance in transfer learning. Researchers from Stanford introduced ChexNet [13], a DenseNet121 architecture which outperformed the average radiologist for the task of lung disease classification using CXRs. Since the performance of any model depends on the pretrained weights, several studies used the ChexNet weights for transfer learning. Mahajan et al. [14] studied the effects of ChexNet weights, ImageNet weights, and random weights for this task, but the performance was limited due to the imbalanced dataset.

Class imbalance is a necessity that needs to be addressed in machine learning. Machine learning is heavily dependent on a balanced dataset for unbiased training. Nahida et al. [15] reported the effects of using random undersampling, random oversampling, and SMOTE for data balancing. The downside to such data sampling techniques is the increase in training time and the lack of reliability of the newly generated data [16]. While random oversampling contributes to new data, data generated using SMOTE is not guaranteed to capture the required features of pediatric pneumonia. Ensemble-based classification is one of the many ways to deal with class imbalanced classification. A weighted average ensemble technique with weights assignment based on precision, recall, F1-score, and AUC was proposed to deal with class imbalanced classification [17].

In order to improve feature extraction and classification utilizing the appropriate features, attention networks determine the importance of different input parts. It assists in concentrating on the necessary lesions rather than unnecessary aspects when treating pediatric pneumonia. Convolutional Block Attention Module (CBAM) efficiently introduced spatial and channel attention for attention-aware features [18]. Our work proposes a transfer learning-based approach with spatial and channel attention incorporated via CBAM. Cost-sensitive learning is used for balanced training. The proposed models' predictions are ensembled using a weighted average, as demonstrated in [17].

3 Background

3.1 CBAM

The attention mechanism developed in LSTMs to deal with long-term dependencies extended to pattern recognition with the introduction of Residual Attention Networks (RANs) [19]. However, RANs have a huge number of parameters leading to longer training times. The Convolutional Block Attention Module (CBAM) [18] was thus introduced to integrate attention efficiently.

CBAM introduced spatial and channel attention by sequentially applying the channel attention module followed by the spatial attention module. The channel attention module decomposes the input tensor of dimension $c \times h \times w$ into two vectors of dimension $c \times 1 \times 1$: produced by global average pooling and global max pooling. Each of these vectors is sent to the multi-layer perceptron network, where the output vectors are added element-wise and then passed to the sigmoid activation layer, which will give us the channel weights. These weights are then multiplied element-wise to corresponding input feature maps as shown in Fig. 2.

The channel-aware features are then sent to the spatial attention module. The spatial attention module has several operations in it, the first part where max pooling and average pooling are applied across each of the channels and are concatenated. This output is then given as an input to a convolutional layer with sigmoid activation, which will give a feature map of dimension $(1 \times h \times w)$, and this spatial attention mask is then applied to the input feature maps by element-wise multiplication as shown in Fig. 3.

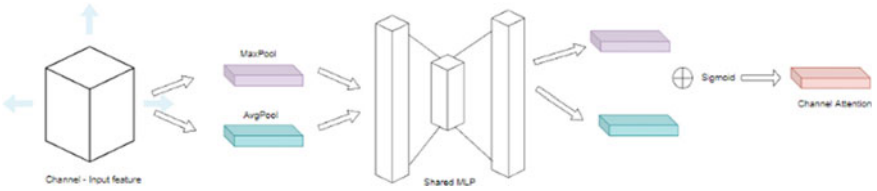


Fig. 2 Spatial attention module

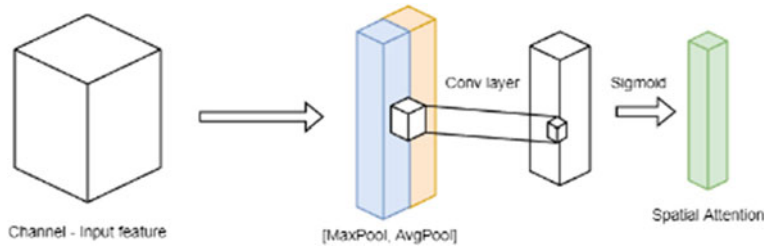


Fig. 3 Channel attention module

Table 1 Distribution of dataset used in the study

Category	Train	Test	Validation
Normal	1349	234	970
Pneumonia	3883	390	970
Total	5232	624	1940

3.2 Cost-Sensitive Learning

Learning from an imbalanced dataset is a common challenge leading to skewed predictions. This problem can be overcome by using weights to penalize each class. The use of weights automates balanced learning in the backward pass while learning. The loss function for cost-sensitive learning is given as follows:

$$L = -\sum_{s \in \text{samples}} \sum_{n} t^n \log\left(\text{pred}^n\right) * C[\text{class}(s), n] \tag{1}$$

4 Dataset and Experimental Design

4.1 Dataset Description

The Kermany et al. [9] dataset containing 5856 CXRs has been used as the dataset to perform all the experiments in this paper. These CXRs are from children between 1 and 5 years from the Guangzhou Women and Children's Medical Centre. The dataset contains two classes: normal and pneumonia, of which 1349 CXRs belong to normal, and 3883 CXRs belong to pneumonia. The dataset shows a clear skew to the pneumonia class. Random geometrical augmented images from the training set were used for validation, as given in Table 1.

4.2 Proposed Methodology

The proposed pipeline is inspired from CBAM to make robust predictions and uses a weighted average ensemble using a Tanh weights assignment [17]. The model takes in an input image of size 224×224, with each image normalized by the image data generator and on-the-fly augmentations with a sheer value of 0.2, zoom value of 0.2, and horizontal flip. The features from each of VGG19 and Resnet152V2 are passed through the CBAM block. The resulting attention-aware features capture both spatial and channel attention. The compression ratio used in this study was decided based on extensive experimentations. The attention-aware feature calibration happens in

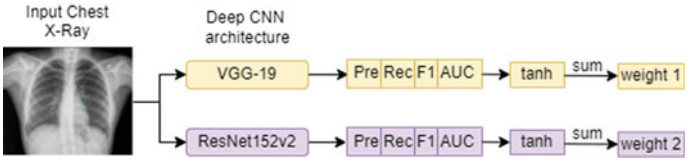


Fig. 4 Detailed illustration of weights assignment using in the proposed pipeline

an end-to-end fashion, and these features are then sent as input to a global average pooling layer, followed by the use of a dense layer with 512 neurons. The final classification is obtained using sigmoid activation.

In general, weighing based on accuracy does not consider false positives and false negatives, and this is important to reduce the misclassification rates. Therefore, in our proposed technique, each model's precision, recall, F1-score, and AUC are stacked into a 4×1 dimensional vector. This vector is passed through Tanh activation and added to give the weight for a given model. These weights are multiplied with the probability output of each model's prediction and averaged. The resulting probability is categorized as pneumonia if greater than or equal to 0.5 and normal if less than 0.5. This assignment strategy ensures that the weights used will improve class imbalanced classification. The process is shown in Fig. 4.

4.2.1 VGG19

The VGG network is an extension of the AlexNet architecture with a major emphasis on the depth of the network. The depth of the network is increased by adding more convolutional layers with a kernel size of 3×3. The convolution filters use a stride of 1 and max pooling over a widow size of 2×2 with a stride of 2. Increasing convolution filters and max pooling layers result in improved feature learning compared to AlexNet. Our study proposes using VGG19 for feature extraction as it is computationally lightweight, which prevents the model from overfitting.

4.2.2 ResNet152V2

The introduction of very deep convolutional architectures in VGGs has improved performance on the ImageNet dataset. However, experiments to further increase the number of convolutional layers for improved performance started to saturate. The main obstacle to this was not the vanishing gradient but rather the degradation of features with increasing convolution layers. This problem was elegantly tackled by using a skip connection. The skip connection adds the output from the previous layer to the layer ahead; thus, ensuring features are retained with increasing convolutions.

The ResNet152v2 with 152 convolutional layers was used in our proposed methodology to improve feature learning by leveraging more filters and simultaneously preventing spatial degradation.

5 Performance Metrics

The ability to differentiate various models' performances is a crucial component of evaluation measures. Accuracy, precision, recall, F1-score, and the AUC value are the main metrics our study uses to assess our models. The model's predictions are illustrated by the confusion matrix in Fig. 5.

True Positive (TP)—number of pneumonia X-rays correctly predicted as pneumonia.

False Negative (FN)—number of pneumonia X-rays wrongly predicted as normal.

True Negative (TN)—number of normal X-rays correctly predicted as normal.

False Positive (FP)—number of normal X-rays predicted wrongly as pneumonia.

The accuracy of a model is calculated as the ratio between correct predictions and total predictions as given in Eq. (2). Equation (3) shown the precision calculated as the ratio between true positives and predicted positives. The recall of a model is calculated as the ratio between true positive and the actual positives as given in Eq. (4). Equation (5) shows the F1-score calculated as the harmonic mean of precision and recall.

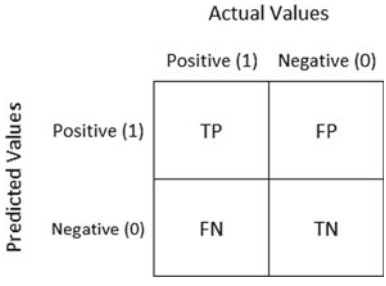
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$F1 \ - \text{score} = 2\left(\frac{\text{Precision} \ * \ \text{Recall}}{\text{Precision} + \ \text{Recall}}\right) \tag{5}$$

Fig. 5 Confusion matrix



6 Results and Discussion

Several deep-CNN models were trained on the 5232 X-ray images for 30 epochs and evaluated on the test data consisting of 624 images to determine the model best suited for feature extraction. Two best-performing models are selected which result in good performance when trained on a class imbalanced dataset. These models are then trained using class weights for unbiased training. The proposed architecture outperforms several existing deep CNN models, as given in Table 2. All are trained on Google Colab for 30 epochs with Adam as the optimizer, 32 as the batch size, and a fixed learning rate of 0.00001.

The best-performing deep CNN models are VGG19 and ResNet152V2, with an accuracy of 93.59% and 95.19%, respectively. Both these models have an AUC score greater than 92%. The results show that all models have high recall values due to the dataset's imbalanced nature. VGG19 and ResNet152V2 perform well even when trained on an imbalanced dataset due to the respective advantages of low computational cost and prevention of feature loss in VGG19 and ResNet152V2. ResNet152V2 outperforms all deep CNN architectures due its capability to learn more features without compromising on the feature loss. Modified VGG19 and Resnet152V2 with attention and cost-sensitive learning are compared in Table 3.

The class weight used in this study is 1.939 for normal and 0.674 for pneumonia. These weights were selected after extensive experimentations. The proposed CS VGG19 Attn and CS ResNet152V2 Attn outperform their counterparts in accuracy, precision, F1-score, and AUC value. The proposed VGG19 model with cost-sensitive and attention block shows an increase in precision compared to the existing VGG19, but the reduction in the recall value suggests the limitation of the model even with the presence of cost-sensitive learning. But in contrast to this, we can notice an increase in recall value of the proposed ResNet152V2 when compared to the existing ResNet152V2. This is due to the residual connections that prevent

Table 2 Comparison of performance of the proposed model with several deep CNN architectures

Model	Accuracy	Precision	Recall	F1-score	AUC
VGG16	88.79	84.78	100.0	91.76	85.04
VGG19	93.59	92.07	98.21	95.04	92.05
DenseNet121	89.10	85.46	99.49	91.94	85.64
DenseNet169	86.70	82.59	99.74	90.36	82.35
DenseNet201	87.34	83.30	99.74	90.78	83.21
InceptionV3	89.90	86.25	99.74	92.51	86.62
Xception	92.79	90.78	98.46	94.46	90.90
ResNet152V2	95.19	97.62	94.62	96.09	95.38
MobileNetV2	85.90	81.86	99.49	89.81	81.37
EfficientNetB0	58.97	96.53	35.64	52.06	66.75
Proposed pipeline	**96.79**	**96.48**	**98.46**	**97.46**	**96.24**

Table 3 Performance comparison between proposed model and existing deep CNN model

Model	Accuracy	Precision	Recall	F1-score	AUC	Training time (minutes)	Feature extraction time (second)
VGG19	93.59	92.07	98.21	95.04	92.05	89	7.00
CS VGG19 Attn	95.83	95.96	97.44	96.69	95.3	91	7.57
ResNet152V2	95.19	97.62	94.62	96.09	95.38	91	10.22
CS ResNet152V2 Attn	96.63	96.01	98.72	97.35	95.94	92	11.37
Weighted average ensemble	**96.79**	**96.48**	**98.46**	**97.46**	**96.24**	**180**	**21.59**

feature loss and help in efficient training. The false negatives may be reduced with this improved feature flow in model resulting in improved recall which is crucial in field of biomedical image analysis. However, we can still find a slight decrease in precision value from the proposed ResNet152V2 from its baseline which may be due to the randomness while training the models. Overall, both the proposed models show improved AUC values which clearly defines the number of correctly classified normal and pneumonia CXRs.

The computation time is slightly higher in the proposed models when compared to their counterparts. However, this increase in both feature extraction and training time is countered with the improved performance. The weighted average ensemble of the predictions with weights assignment based on the precision, recall, F1-score, and AUC value of both the models results in an accuracy of 96.79% and 96.24% AUC. The confusion matrix of test data is shown in Fig. 6; the model results in 14 false positives and 6 false negatives. The validation loss and accuracy plots in Fig. 7 show that the model exhibits proper training with no signs of overfitting.

The features extracted from each proposed model's penultimate layer are visualized using the t-SNE plot in Fig. 8. The t-SNE plot shows cluster formation with minor overlaps and outliers.

The proposed model is compared with recent works on the Kermany et al. [9] dataset in Table 4. The proposed model shows competing performances with recent works with remarkable precision of 96.48%, F1-score of 97.46%, AUC of 96.24%, and accuracy of 96.79%.

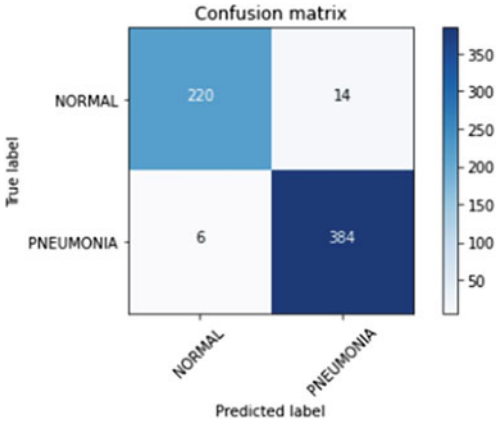


Fig. 6 Confusion matrix for proposed pipeline on the test data

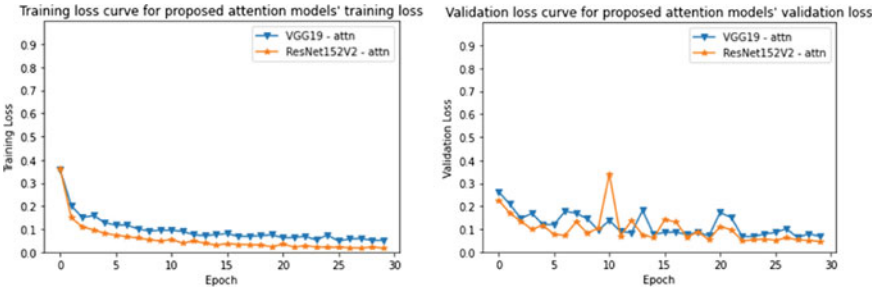


Fig. 7 Training and validation accuracy—loss history of the proposed model

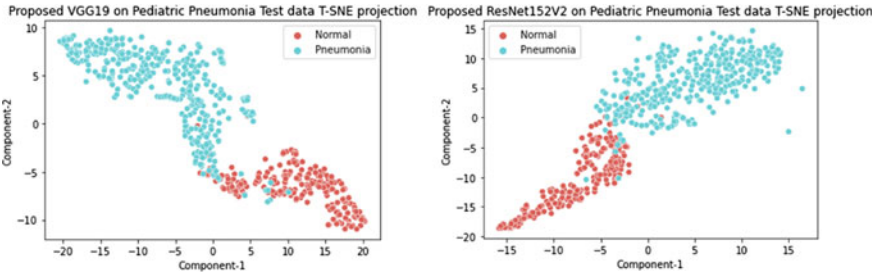


Fig. 8 t-SNE feature representation of the test data extracted from the proposed model

7 Conclusion and Future Work

This work proposes a computer-aided diagnosis tool for pediatric pneumonia diagnosis using easily replicable attention-aware CNN architectures. The individual predictions from each proposed model are classified into normal and pneumonia

Table 4 Performance of other recent works on the Kermany et al. [9] dataset

Author	Accuracy	Precision	Recall	F1-score	AUC
Kermany et al. [9]	92.8	90.1	93.2	–	–
Stephen et al. [12]	93.73	–	–	–	–
Siddiqi et al. [20]	94.39	92.0	99.0	–	–
Rajpurkar et al. [13]	88.78	–	–	–	–
Mittal et al. [21]	96.36	–	–	–	–
Proposed model	**96.79**	**96.48**	**98.46**	**97.46**	**96.24**

using a weighted average ensemble classifier. The extracted features from the penultimate layer of the proposed models are visualized using t-SNE plots. These plots indicated the need for a better model for outliers and overlapping clusters. The proposed pipeline achieves an accuracy of 96.79% and an AUC of 96.24%. With the proposed pipeline performing better than ChexNet [13], pediatric pneumonia diagnosis can be accelerated, thereby reducing the totality of such ailments.

Future works can include expanding the work to cover other pneumonia datasets. Generative adversarial networks can be used for data augmentation to get improvised results. Unsupervised learning frameworks can be utilized to learn from unlabeled datasets. Various other ensembled models can be implemented to get better results.

References

1. Yadav P et al (2021) Lung-GANs: unsupervised representation learning for lung disease classification using chest CT and x-ray images. IEEE Trans Eng Manag
2. Liang G, Zheng L (2020) A transfer learning method with deep residual network for pediatric pneumonia diagnosis. Comput Methods Prog Biomed 187:104964
3. Kör H, Erbay H, Yurttakal AH (2022) Diagnosing and differentiating viral pneumonia and COVID-19 using X-ray images. Multimedia Tools Appl, 1–17
4. Ramírez J et al (2010) Computer aided diagnosis system for the Alzheimer's disease based on partial least squares and random forest SPECT image classification. Neurosci Lett 472(2):99–103
5. Jiang M et al (2022) Image classification of alzheimer's disease based on external-attention mechanism and fully convolutional network. Brain Sci 12(3):319
6. Babu GS et al (2021) Tuberculosis classification using pre-trained deep learning models. Adv Auto, Signal Process, Instrument, Cont, 767–774
7. Karthikeyan S et al (2019) Detection of multi-class retinal diseases using artificial intelligence: an expeditious learning using deep CNN with minimal data. Biomed Pharm J 12(3):1577–1586
8. Vaisali B, Parvathy CR, Vyshnavi AMH, Namboori PKT (2019) Hypoxia diagnosis using deep CNN learning strategy—a theranostic pharmacogenomic approach. Int J Prognostics Heal Manag 10:7
9. Kermany DS et al (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172(5):1122–1131
10. Saraiva AA et al (2019) Models of learning to classify c-ray images for the detection of pneumonia using neural networks. Bioimaging

11. Nafiiyah N, Setyati E (2021) Lung x-ray image enhancement to identify pneumonia with CNN. In: 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT). IEEE
12. Stephen O et al (2019) An efficient deep learning approach to pneumonia classification in healthcare. J Health Eng
13. Rajpurkar P et al (2017) Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225
14. Mahajan S et al (2019) Towards evaluating performance of domain specific transfer learning for pneumonia detection from x-ray images. In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT). IEEE
15. Habib N et al (2020) Ensemble of CheXNet and VGG-19 feature extractor with random forest classifier for pediatric pneumonia detection. SN Computer Sci 1(6):1–9
16. Jiang Z et al (2021) A new oversampling method based on the classification contribution degree. Symmetry 13(2):194
17. Kundu R et al (2021) Pneumonia detection in chest X-ray images using an ensemble of deep learning models. PloS one 16(9):e0256630
18. Woo S et al (2018) Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV)
19. Wang F et al (2017) Residual attention network for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition
20. Siddiqi R (2019) Automated pneumonia diagnosis using a customized sequential convolutional neural network. In: Proceedings of the 2019 3rd international conference on deep learning technologies
21. Mittal A et al (2020) Detecting pneumonia using convolutions and dynamic capsule routing for chest X-ray images. Sensors 20(4):1068

An Integrated Deep Learning Deepfakes Detection Method (IDL-DDM)

Warusia Yassin, Azwan Johan, Zuraida Abal Abas, Mohd Rizuan Baharon, Wan Bejuri, and Anuar Ismail

Abstract Deepfakes have fascinated enormous attention in recent times ascribable to the consequences of threats in video manipulation. Consequently, such manipulation via intelligent algorithm contributes to more crucial circumstances as electronic media integrity become a challenging concern. Furthermore, such unauthentic content is being composed and outstretched across social media platforms as detecting deepfakes videos is becoming harder nowadays. Nevertheless, various detection methods for deepfakes have been schemed, and the accuracy of such detection models still emerges as an open issue, particularly for research communities. We proposed an integrated deep learning deepfakes detection model namely IDL-DDM to overcome ongoing criticism, i.e., difficulties in identifying the fake videos more accurately. The proposed IDL-DDM comprises side-by-side deep learning algorithms such as Multilayer Perceptron and Convolutional Neural Network (CNN). In addition, the Long Short-Term Memory (LSTM) approach is applied consecutively after CNN in order to grant sequential processing of data and overcome learning dependencies. Using this learning algorithm, several facial region characteristics such as eyes, nose, and mouth are extracted and further transformed into numerical form with the intention to identify video frames more precisely. The experiments were performed via different datasets such as the Deepfakes Detection Challenge Dataset (DFDC) and Unseen (YouTube Live) videos which comprise a wealth of original and fake videos. The experimental results represent a higher achievement for the IDL-DDM in contrast to other previous similar works.

Keywords Deep learning · Long short-term memory (LSTM) · Deepfakes · Convolutional neural network (CNN)

W. Yassin (✉) · A. Johan · Z. A. Abas · M. R. Baharon · W. Bejuri
Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
e-mail: s.m.warusia@utem.edu.my

A. Ismail
Ask-Pentest SDN BHD, Kuala Lumpur, Malaysia

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_6

1 Introduction

Deepfakes have lately gained popularity and have been extensively used in high-profile crimes due to their capability of assembling images or videos and the ease of use of their apps for a wide spectrum of persons with varying computing skills. A video that has been manipulated using an algorithm to replace the person in the original video with a different person (particularly a public figure) in a way that makes the video look genuine is an example of a deepfakes, according to [1]. Today, deepfakes technology was brought in harmful nature where it uses in an unethical way on various of channels, particularly social media. The assembled authentic images or videos are further used to spread fake news, financial fraud, malicious impostures, and criminal activities [2]. For instance, Former US President Donald Trump's tweet was faked using deepfakes and the tweets portray Trump mocking Belgium for remaining with the Paris Climate Agreement. If such fake news is not eradicated, it could cause a world war and can impact the whole world [3].

Various researchers have proposed and developed detection methods to combat deepfakes that could cause chaotic events around the globe. Furthermore, most of the researchers employed a single method such as [4–9] and many more to detect the deepfakes features. Such an approach is common and incapable to detect robustness deepfakes, which are created by the highly skilled person using a sophisticated method. Consequently, the trend of deepfakes becomes surprising and escalates within a short period of time frames and keeps worrying the social media user [10].

In this work, we proposed an integrated deep learning deepfakes detection method called IDL-DDM. In contrast to previous solutions, the innovation of the proposed integrated method comprises two parallel approaches; the first layers with MLP and the latter with CNN and LSTM. The first layers used for Facial Landmark Detection consist of temporal analysis stages before the entire features feed into MLP. The second layer is applied for feature extraction using CNN and LSTM after the entire images are converted into numerical form. Such a solution is able to solve challenging tasks such as lower accuracy, overfitting, and sequencing problem.

The remaining section is organized as follows. The related work of existing works is discussed in Sect. 2, and the detail of the proposed method is in Sect. 3. Section 4 describes the experimental result, while Sect. 5 presents conclusion and future work.

2 Related Work

The characteristic of deepfakes has been rising tremendously and directly impacts the detection methods' capability and bring challenges for improvement. There are two types of deepfakes detection that have been broadly discussed by researchers, i.e., shallow classifiers and deep classifiers [2]. Using shallow classifiers, the fake and real images or videos can be differentiated via inconsistency features such as

texture and color around the face, facial region, and many more [3]. Instead, deep classifiers rely on affine face-warping methods such as rotating, scaling, and shearing with minimal resolutions. Furthermore, to defeat the advanced deepfakes method, a number of researchers looking into potential deep learning methods such as CNN models with VGG and ResNet, MLP, and LSTM.

Masood et al. [11] performed deepfakes detection by employing CNN and LSTM to combat highly trained professionals' deepfakes videos. The proposed approach is made up of two primary parts: LSTM for temporal sequence analysis and CNN for frame feature extraction. The detection methods have been tested with a realistic set of sequence samples from renowned films with an emphasis on human activities. This method was able to achieve higher accuracy at 97% using the HOHA dataset.

Besides, Kolagati et al. [12] exercised four different deep learning methods namely LightCNN, ResNet, DenseNet, and SqueezeNet for deepfakes detection. The author's combine such methods with eyebrow recognition as the affected eyebrow regions (abnormality) in synthesized images can be identified easily. In addition, the presence of anomalies identified correctly considered approach using eyebrow matching that relies on visible structural artifacts on color inconsistencies. The detection method has been tested with high-quality realistic images that have been presented in Celeb-DF dataset. The author's proposed method manages to obtain 87.9% as AUC which is difficult to achieve in various previous works.

Furthermore, Xie et al. [13] proposed a modified and lighter version of AlexNet via CNN architectures. The author's used fewer layers than usual for AlexNet in this study, including three convolutional layers, three max-pooling layers, one flattens layer, one dense layer, one activation layer, and an optional dropout layer. The AlexNet was modified to generate filters that could increase size sequentially and ReLU activation function as an advantage for robust feature extraction and pattern recognition. Three different datasets have been used to evaluate the proposed detection methods such as UADFV, FaceForensics++, and Celeb–DF. This method utilized lesser training time and managed to maintain the accuracy rate at 98.73%, 91.32%, and 98.85% for UADFV, FaceForensics++, and Celeb–DF datasets.

Moreover, to further improve the deepfakes detection methods capability, Montserrat et al. [14] have proposed RNNs and CNNS that are able to identify manipulations by extracting visual and temporal characteristics of face video. The proposed method utilized automatic face weighting, whereas the most reliable regions were emphasized as compared to the least reliable regions in determining a video-level prediction. Such a mechanism has eliminated the false positive in which most existing deepfake detection incorrectly reports the background regions of the frames that contain faces. The proposed method has been trained and tested with a DFDC dataset that comprises highly realistic manipulation images. The recorded accuracy is 91.85% and better than others such as Xception, LSTM, and Exception Net5.

Furthermore, Su et al. [15] employed convolutional Long Short-Term Memory (Convo LSTM) and Convolutional Gated Recurrent Unit (Conv GRU) to extract features from video frames. This method overcomes the weaknesses of existing deepfakes detection such as the inability in detecting deepfake in compressed video and varied temporal characteristics. The video feed into the deep learning network as

a 3D vector to maintain inter-frame information and boost detection accuracy. The author's applied four different datasets, namely Celeb -DF(v2), DF-Timit, FF-DF, and DFDC to evaluate the proposed method and achieved 89.3% accuracy of AUC and outperform other detection such as Capsule and Face X-ray.

Moreover, Caldelli et al. [16] employed CNN for deepfake detection via an embedded optical flow approach. This approach is able to represent how the apparent motions of the elements in a scene are affected by the relative motion of the observer (such as the camera) and the environment. The fundamental architecture has two phases in which in the first phase the face is cropped to estimate optical flow fields, while the later phase involves the CNN function to determine the quality of the frame either tampered or original. The proposed approach has the ability to distinguish possible motion dissimilarities in the temporal structure of video sequences and the highest accuracy rate recorded is nearly 97.35% using the FaceForensics++ dataset.

Wodajo et al. [17] proposed a distinct Convolutional Neural Network (CNN) along with Vision Transformer (ViT) to overcome the weaknesses of generalizations. The CNN assists in extracting learnable features, while the ViT uses an attention mechanism to classify the learned features as input. The proposed CViT learns the features of input images and determines whether a specific video is fake or genuine. The method evaluates using Deepfakes Detection Challenge Dataset (DFDC) with 400 unseen videos and obtained 91.5% and 0.91 as an accuracy and AUC rate.

More recently [2], has proposed MLP and CNN to detect deepfakes videos. Various facial attributes such as the shape of eyes, nose, and lips were extracted using landmarks detection before differentiating the genuine and fake videos by MPL. Parallelly, CNN is applied to extract features and train models after the entire images are converted into numerical form. DESSA and DFDC dataset is applied to evaluate the proposed detection method and the result is 84% of accuracy.

The related works study has revealed that current deepfakes detection methods still possess limitations in differentiating the original and genuine images or videos. Consequently, several researchers have highlighted the lower accuracy usually contributed when dealing with low light environments, and some methods do not have dropout layers to overcome overfitting and sequencing issues. Nevertheless, numerous works proposed lately but still require improvement as the detection accuracy is not promising. The previous works' achievements, methods, and applied datasets have been provided in Table 1.

Table 1 Previous work

Author	Method	Dataset	Result
Guera et al. [11]	CNN + LSTM	Celeb-DF V2 HOHA	96.7% accuracy
Nguyen et al. [12]	Eyebrow recognition (Light CNN, ResNet, DenseNet, SqueezeNet	Celeb-DF	0.832 AUC
Xie et al. [13]	Modified AlexNet + CNN architecture	UADFV FaceForensics++ Celeb-DF	98.73% UADFV 91.32% FaceForensics++ 98.85% Celeb-DF
Zhao et al. [18]	Two level features with two stream CNN	FaceForensics++	99.93% accuracy
Montserrat et al. [14]	CNN + RNN with Automation Face Weighting	DFDC Dataset	91.85% accuracy
Wodajo et al. [17]	Convolutional Neural Network (CNN) + Vision Transformer (ViT)	DFDC	91.5% accuracy
Su et al. [15]	Conv LSTM + Conv GRU	Celeb-DF(V2) DF-Timit FF-DF DFDC	0.893 AUC
Caldelli et al. [16]	Optical flow based on CNN	FaceForensics++	97.35% accuracy
Kolagati et al. [2]	CNN + MLP	DFDC dataset (deepfake detection challenge)	0.877 AUC 84% accuracy

3 Proposed Integrated Deep Learning Deepfakes Detection Method

Based on the literature study, after performing the gap analysis, we proposed integrated deep learning methods to overcome existing limitations specifically in deepfakes detection. We considered the Residual CNN network which can optimize high-performance deep neural networks. ResNext 50 has been chosen as it can reduce the numerous obligatory hyperparameters in contrast to ResNet. Using ResNext 50, a similar transformation in ResNet was applied 32 times while cardinality was four times and the entire result was aggregated at the end.

Referring to Fig. 1, the extraction process in the proposed methods is executed by Facial Landmark Detection and Convolutional Neural Network (CNN). However, before this extraction process, the entire data was preprocessed such as the fraction of video frames into images and normalization. Using Facial Landmark Detection, the face region (as illustrated in Fig. 2) of each image comprises 68 facial landmarks

specifically the region from eyes, mouth, nose, and face and located as (x, y) coordinates. The nose features have not been considered as most previous works focus on the eyes, mouth, and the rest face region. Characteristics of deepfakes has been rising tremendously and directly impacts the detection methods' capability and bring challenges for improvement.

Contrastingly, using CNN, the entire extracted images from the videos were transformed into numerical values by resized images at $224 \times 224 \times 3$ (RGB Layer) as illustrated in Fig. 3 for easier processing and consistency. The CNN comprises numerous iterations, including densely linked layer, convolution, Rectified Linear Unit (ReLU), and pooling. A complete connection layer is added once the next layer is flattened, and the appropriate batch normalization and dropout procedures are undertaken further. The intention is to speed up the training process and increase the generalization of the created model.

Next, for image classification, we employed MLP while LSTM for sequence processing. The MLP consists of a neural network layer with the activation function and is divided into two parts of fully connected layers, i.e., the dense layer with ReLU activation and the hidden layer with ReLU activation. ReLU has chosen to avoid vanishing gradient drawbacks and for better computation as it does not activate the entire neuron at the same time. On the other hand, LSTM is applied basically to aid in the sequential processing of data and overcome the learning of long-term dependencies. Upon receiving input from CNN, the frames are then processed sequentially using LSTM, after which the features of the frames are compared over time.

The next stages involve the concatenation layer, in which the output from LSTM and MLP modules will be conjugated. Once, the output is conjugated (combined and average), the softmax function is further applied to determine the probability of the output either being genuine or fake. Figure 4 illustrates the flowchart of the deepfakes detection which has been evaluated using the Deepfakes Detection Challenge (DFDC) dataset using 318 videos as 199 are fake and the remaining 119 are genuine and unseen videos (YouTube Live) with 33 fake and 33 genuine.

4 Experiments and Results

The proposed solution has been evaluated using several data such as Deepfakes Detection Challenge (DFDC) [20] and videos from YouTube Live (unseen) [21]. Table 2 illustrates the distribution of videos used in the experiment, where the total video evaluated is 318.

The performance evaluation metrics applied to the proposed detection methods are illustrated in Table 3. This metric is commonly followed by the research community. True positive specifies the total correctly fake video frames classified as fake, while true negative is the correctly genuine video frames classified as genuine. Contrastingly, false positive specifies as total incorrectly genuine video frames classified as fake, while false negative specifies as total incorrectly fake video frames classified as genuine.

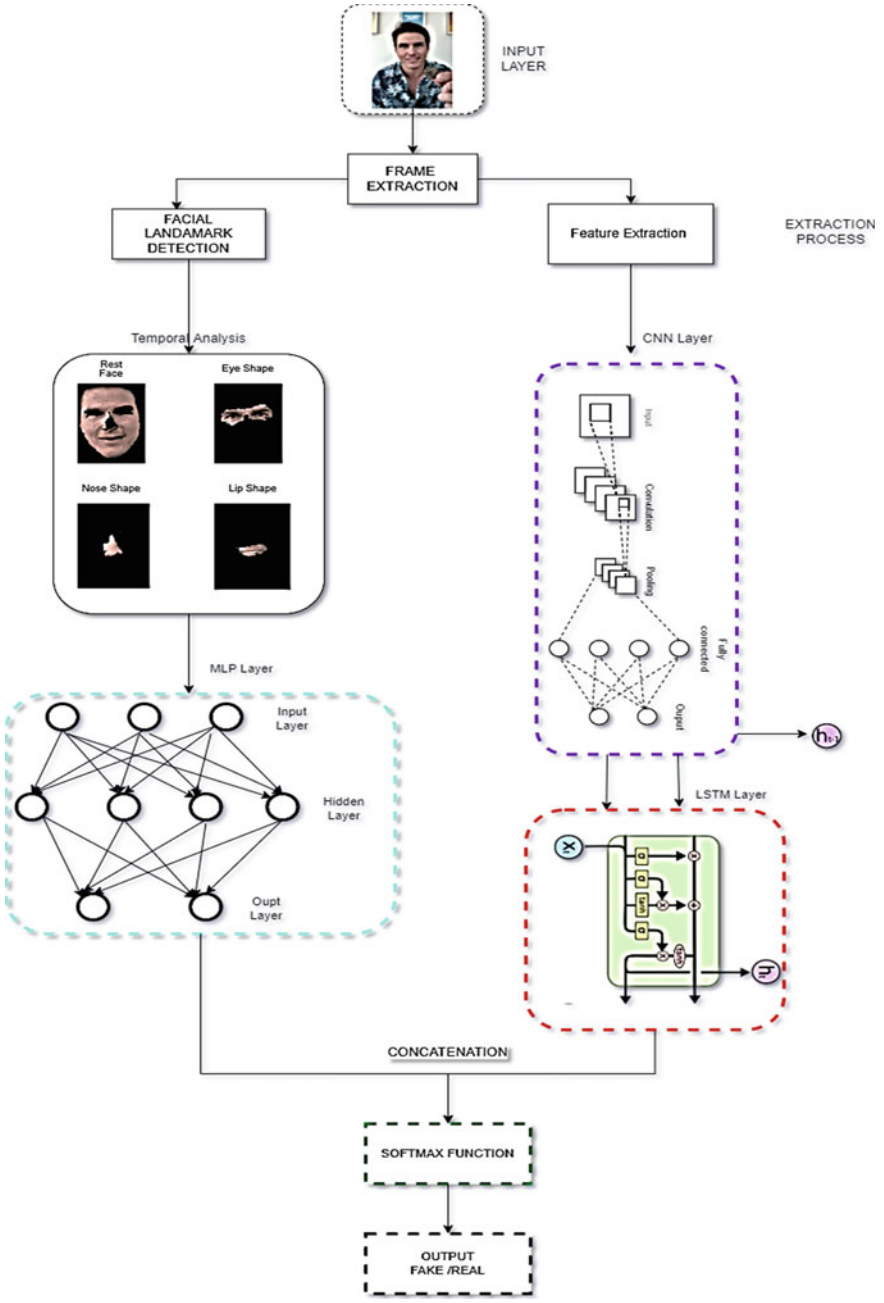


Fig. 1 Proposed deep learning mode

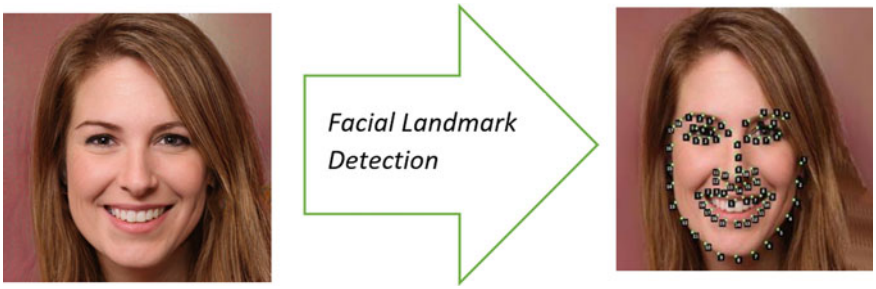


Fig. 2 Facial landmark detection

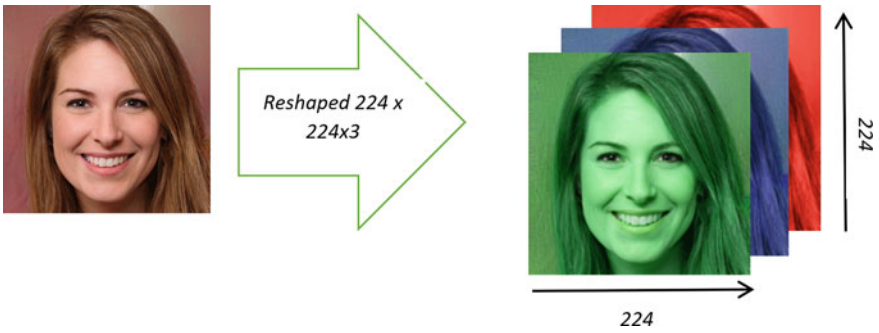


Fig. 3 Image to numerical form extraction

Performance evaluation of the proposed detection methods against each dataset is illustrated in Table 4. Based on this table, evaluation using DFDC the dataset, the IDL-DDM has achieved 98% of accuracy with 0.94 as AUC. Furthermore, using unseen data from YouTube Live has obtained 95% of accuracy and 0.85 AUC.

A performance comparison of the proposed detection methods called IDL-DDM against the DFDC dataset and existing available methods is illustrated in Table 5. Referring to this table, the proposed IDL-DDM has outperformed the other existing works with higher accuracy and AUC. Hence, the proposed solution has achieved the aim to increase accuracy by correctly identifying fake and genuine videos.

5 Conclusion and Future Work

In this work, we proposed integrated deepfakes detection methods using MLP, CNN, and LSTM. The aim is to extract the significant facial characteristics such as eyes, nose, and face region and perform classification to identify genuine and fake video frames. Compared to existing works, the proposed detection methods consist of LSTM to aid in the sequential processing of data and overcome the learning of

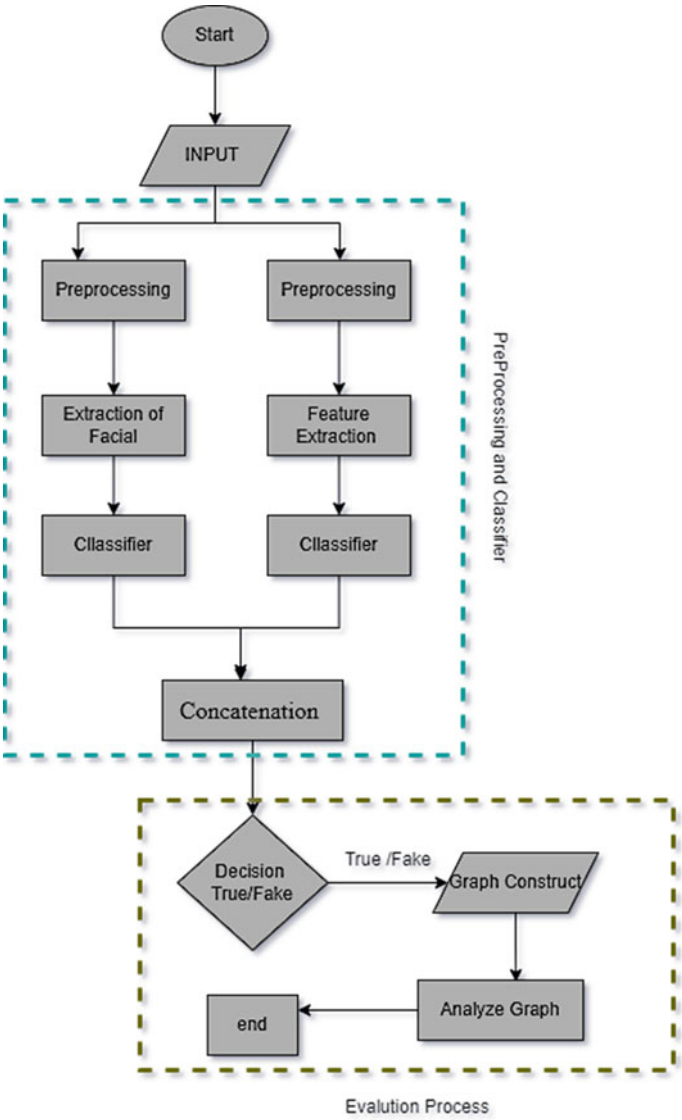


Fig. 4 Proposed detection flow

Table 2 Dataset distribution

Dataset	Real	Fake	Total
DFDC	119	199	318
Unseen (YouTube)	21	45	66

Table 3 Dataset distribution

Metrics	Formula
Accuracy	(True Positive + True Negative)/(True Positive + True Negative + False Positive + False Negative)

Table 4 Performance evaluation IDL-DDM

Dataset	Accuracy (%)	AUC
DFDC	98	0.94
Unseen (YouTube)	95	0.85

Table 5 Performance comparison (Accuracy) using DFDC dataset

Authors	Methods	Accuracy (%)	AUC
Montserrat et al. [14]	CNN + RNN	91.85	–
Su et al. [15]	LSTM + GRU	–	0.893
Wodajo et al. [17]	CNN + ViT	91.5	–
Kolagati et al. [2]	CNN + MLP	84	0.877
Proposed IDL-DDM	MLP + CNN + LSTM	98	0.94

long-term dependencies. The experiments via DFDC and unseen dataset against the proposed detection showed an increasing accuracy rate compared to existing works. Extraction of more characteristics that exist in the human face such as eye blinking can be considered for future work. Moreover, Yassin et al. [19] have highlighted the application of deep learning for motion recognition in real-time that can be further studied.

Acknowledgements This publication is funded by UTeM and ASK-PENTEST SDN BHD under the industrial grant INDUSTRI/APSB/FTMK/2021/I00063.

References

1. Masood M, Nawaz M, Malik KM, Javed A, Irtaza A, Malik H (2022) Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. Appl Intell. https://doi.org/10.1007/s10489-022-03766-z
2. Kolagati S, Priyadharshini T, Rajam VAM (2022) Exposing deepfakes using a deep multilayer perceptron—convolutional neural network model. Inter J Info Manage Data Insights 2(1). https://doi.org/10.1016/j.jjimei.2021.100054
3. Zhang T (2022) Deepfake generation and detection, a survey. Multimed Tools Appl 81(5):6259–6276. https://doi.org/10.1007/s11042-021-11733-y
4. Johnson D, Gwyn T, Qingge L, Roy K (2022) Deepfake detection using CNN trained on eye region. In: Advances and trends in artificial intelligence. Theory and practices in artificial intelligence, pp 443–451

5. Jolly V, Telrandhe M, Kasat A, Shitole A, Gawande K (2022) CNN based deep learning model for deepfake detection. In: 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), pp 1–5. https://doi.org/10.1109/ASIANCON55314.2022.9908862
6. Mustafa A et al (2022) A comparative analysis for extracting facial features to detect deepfake videos by various machine learning methods. SSRN Electron J. https://doi.org/10.2139/ssrn.4202285
7. Raza A, Munir K, Almutairi M (2022) A novel deep learning approach for deepfake image detection. Appl Sci 12(19). https://doi.org/10.3390/app12199820
8. Rani R, Kumar T, Sah MP (2022) A review on deepfake media detection. Comm Intell Syst, 343–356
9. Suratkar S, Kazi F (2022) Deep fake video detection using transfer learning approach. Arab J Sci Eng. https://doi.org/10.1007/s13369-022-07321-3
10. Vamsi VVVNS et al (2022) Deepfake detection in digital media forensics. Global Trans Proceed 3(1):74–79. https://doi.org/10.1016/j.gltp.2022.04.017
11. Guera D, Delp EJ (2018) Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Nov, pp 1–6. https://doi.org/10.1109/AVSS.2018.8639163
12. Nguyen HM, Derakhshani R (2020) Eyebrow recognition for identifying deepfake videos. In: International Conference of the Biometrics Special Interest Group (BIOSIG) 2020:1–5
13. Xie D, Chatterjee P, Liu Z, Roy K, Kossi E (2020) DeepFake detection on publicly available datasets using modified AlexNet. In: 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Dec, pp 1866–1871. https://doi.org/10.1109/SSCI47803.2020.9308428
14. Montserrat DM et al (2020) Deepfakes detection with automatic face weighting. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun, pp 2851–2859. https://doi.org/10.1109/CVPRW50498.2020.00342
15. Su Y, Xia H, Liang Q, Nie W (2021) Exposing DeepFake videos using attention based convolutional LSTM network. Neural Process Lett 53(6):4159–4175. https://doi.org/10.1007/s11063-021-10588-6
16. Caldelli R, Galteri L, Amerini I, del Bimbo A (2021) Optical Flow based CNN for detection of unlearnt deepfake manipulations. Pattern Recognit Lett 146:31–37. https://doi.org/10.1016/j.patrec.2021.03.005
17. Wodajo D, Atnafu S (2021) Deepfake video detection using convolutional vision transformer
18. Zhao Z, Wang P, Lu W (2020) Detecting deepfake video by learning two-level features with two-stream convolutional neural network. In: ACM International Conference Proceeding Series, Apr, pp 291–297. https://doi.org/10.1145/3404555.3404564
19. Yassin WM, Abdollah MF, Muslim Z, Ahmad R, Ismail A (2021) An emotion and gender detection using hybridized convolutional 2D and batch norm residual network learning. In: 2021 The 9th International Conference on Information Technology: IoT and Smart City, Dec, pp 79–84. https://doi.org/10.1145/3512576.3512590
20. Deepfake Detection Challenge. https://www.kaggle.com/c/deepfake-detection-challenge. Accessed 15 July 2022
21. YouTube Videos. https://www.youtube.com. Accessed 15 July 2022

Melanoma Detection Using Convolutional Neural Networks

Venkata Sai Geethika Avanigadda, Ravi Kishan Surapaneni, and Devika Moturi

Abstract The prevalence of skin cancer is a huge social issue. Melanoma is one type of skin cancer which is known as malignant melanoma. It is the most dangerous skin cancer which is spreading more vastly. Melanoma makes up the majority of skin cancer deaths roughly 75% of them. Detecting melanoma cancer as early as possible and receiving therapy with little surgery are the best ways to beat it. This model quickly categorizes melanoma disease by utilizing efficient higher resolution convolutional neural networks. By using the efficient MobileNetV2 architecture model, the automated melanoma detection model can be developed to identify the skin lesion images. The MobileNetV2 architecture is incredibly lightweight and can be utilized to extract more functionality. The HAM10000 dataset has been used for the evaluation. It uses the global average pooling layer which is connected with the fully connected layers. The proposed system can be used to detect whether the disease is melanoma or not. The model has an accuracy rate of 85%.

Keywords Skin cancer · Melanoma · Convolutional neural network (CNN) · MobileNetV2 · Web application · Model accuracy

1 Introduction

Melanoma is the most dangerous skin cancer causing more mortal in society which will be caused by the rapid growth of melanocytes in our body. Melanocytes will produce a greater number of cells that consist of melanin. This melanin will be used

V. S. G. Avanigadda (✉) · R. K. Surapaneni · D. Moturi
Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India
e-mail: geethikaavanigadda123@gmail.com

R. K. Surapaneni
e-mail: suraki@vrsiddhartha.ac.in

D. Moturi
e-mail: devikamoturi@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_7

for turning the eyes, hair, and skin into black color. Sometimes these cells which produce melanin will be turned into malignant melanoma. Malignant melanoma is dangerous when compared to benign malignant. To reduce malignant melanoma, we have to take prior treatment. Malignant melanoma should be identified, and the treatment should be taken as early as possible. Otherwise, it will spread to the other organs. The chance of recovery will get improved by identifying the problem as fast as possible. It is very difficult to differentiate between malignant melanoma and benign melanoma. So, the appropriate convolutional neural network should be used to differentiate between them. The detection of malignant melanoma can be made easier with the help of deeper, wider, and higher resolution MobileNetV2 architecture which resides on convolutional neural networks. It contains two fully connected pooling layers. MobileNetV2 architecture is lightweight architecture, so it is compatible with any device. An image-based artificial intelligence has been used for classifying dermoscopic images to differentiate between benign melanoma and malignant melanoma. But scientists concluded that deep learning is the best technology when compared to artificial intelligence. The proposed system uses the MobileNetV2 architecture for detecting the disease. By the addition of inverted modules with the linear bottleneck residuals modules, the MobileNetV2 architecture has been proposed. MobileNetV2 is inherited from the MobileNetV1 architecture. The entire architecture of the model is based upon the depth-wise separable convolution. The 2D architecture which was used in the model will collect the multiple input images and goes through several channels to produce one output channel. The depth-wise convolution input channel which was given will be divided into several different layers of channels. Based upon the input channel that we have given, it will convolve each and every input channel with the corresponding filtered output channel. The output channels have been filtered and generated, and then, those output channels are stacked back. Inseparable convolution, the outputs are stacked back and then filtered known as the 1×1 convolution technique also known as the point-wise convolution. For combining the stacked output channels into one output channel, point-wise convolution is used. It is indicated that it is more efficient and structured because it will reduce the number of parameters. The regular convolution procedure and the depth-wise separable convolution both yield the same results. By differentiating between the depth-wise and point-wise convolution, it has been concluded that the MobileNetV1 architecture has 28 convolution channel layers that produce the output in the size of $7 \times 7 \times 1280$ pixels size. But the input images were taken by both models; they are same with the size of $224 \times 224 \times 3$ pixels. In order to make the dataset images of this size, they are resized and then detection will be done. The main advantage CNN has over its forerunners is that it can identify crucial components without human interaction, making it the most well-liked. In the previous studies, they used machine learning and computer vision for detecting the disease which includes human intervention. For the improvement of the result, we used the MobileNetV2 model. Currently, in the field of diagnosis of skin cancer, the CNN model gives more accurate results. The model discussed above gives the major advantage for detecting the disease more accurately when compared to other techniques.

1.1 Existing Models & Limitations

Computer vision is the technique that allows computers to think like a human and be trained to distinguish objects based on images and audio files. In the center of non-invasive diagnostics, computer vision is used. Computer vision can be used to differentiate between benign melanoma and malignant melanoma. The methodologies and techniques which are used in computer vision and the convolutional neural networks will differ from each other. For the accurate classification of melanoma lesions, deep learning should be used. With the help of more training data, the classification is improved and it gives better accurate results.

Drawbacks:

- With the help of two parameters manual feature determination and image segmentation, there will be more possibility of getting errors, but deep learning does not rely on these two parameters. So, the possibility of getting stacked errors will also be very less.

1.2 Contributions of Proposed System

- To analyze the existing problems that are in detecting melanoma.
- To build a CNN-based model to classify melanoma skin cancer.
- To develop a Web application that allows the user to upload the input image for detecting whether the disease is melanoma or not.

2 Related Work

Pereira et al. [1] proposed a melanoma detection model based on multiple instances of learning using 3D features. The model was created based on several new color features of dermoscopic images. Multiple instance learning (MIL) and DL has been used. The functionality of the two learnings differs from each other, MIL is used to perform 3D feature extraction on the dermoscopic images, and DL is used for the resorting classification of RGB data. Additionally, the combination of 2D and 3D characteristics should be used to identify more characteristics of the image.

Yao et al. [2] proposed a deep learning model for skin categorization on unbalanced dataset. In this methodology, the classification of skin lesions can be done based upon the innovative single model approach strategy. The problem of overfitting can be resolved by using the dropout and drop block. To handle the uneven sample sized dataset, cumulative learning strategy will be used. This learning strategies will be used only for the small and imbalanced dataset. This will not be suitable for the large and balanced datasets.

Nguyen et al. [3] proposed a sequential dermoscopic pictures for early melanoma diagnosis. Based upon the time points, the classification and detection of the disease will be performed. The lesion images will be given different time points for each and every individual images based upon the coordinates. Then, by identifying the pixel level differences between various images, the lesion can be extracted. They developed two-stream network model which is known as spatial–temporal features to extract the features. When the shapes of the images are of varying sizes, this model does not produce accurate results.

Thurnhofer et al. [4] proposed a skin classification methodology by using the convolutional networks and regularly spaced shifting. In this methodology, multiple spaced shifted versions will be used for testing the input skin lesion image. Each of the image which was shifted will be allocated to one of the network ensembles. After shifting, this will be given to the Google Net for identifying the various transformations which are present in the input image. The processing time that is required to process the network is very high.

Rastghalam et al. [5] proposed a skin melanoma detection method employing HMM-based asymmetric analysis and expectation maximization on microscopic pictures. The suggested system initially extracts the input image's texture and statistical histogram information. The assessment of the tumor can be categorized based on the asymmetric analysis and the derived features that have been extracted. The texture feature extraction can be done by using the LDP algorithm. Based upon the technique fusion-base HMM classifier the feature can be easily extracted and final results can be concluded. The drawback in it is it requires the highest efficient software for taking high resolution images as input.

Zhang et al. [6] proposed a melanoma screening with novel to identify the short-term lesion using a Siamese neural network. To identify the similarity between the two or more dermoscopic images the change detection is used in the proposed system. In order to avoid the image distortion and alignment problem the spatial transformer on the dermoscopic images, the ResNet model has been introduced. When the dermoscopic images contain more amount of noise or when they are integrated with the bubbles or hair, it is difficult to detect the dermoscopic image.

Wang et al. [7] proposed skin disease image recognition by using the deep learning: A Review. The proposed system makes 45 efforts for identifying the skin lesion images. Image recognition, deep learning framework, and evaluation indicators are various types of data augmentation techniques. This model suggests that deep learning is better than the computer aided technologies. The drawback of this approach is less utilization of deep learning frameworks.

Albahli et al. [8] proposed a melanoma detection model by using the segmentation with active contour and YOLOv4-DarkNet. In this methodology, the various dermoscopic images, artifacts such gel bubbles, hair, and other clinical signs will be eliminated. By using the various morphological operations, the image regions can be sharpened. By using the YOLOv4, we will get the less recall score and there will be chances to get more localization error.

3 Proposed Model

Designing and creating the methodology will help the person to identify whether the disease is melanoma or not. MobileNetV2 architecture is the main model in the methodology. In our suggested system diagram, the main modules are preprocessing data, dataset partitioning, training and testing of the data in the dataset, implementation of MobileNetV2 architecture model, and classification for the given input image. The data is preprocessed by using the resampling technique at the preprocessing stage. Based on the number of images that have been taken, the dataset will be differentiated and prepared. Then, later training and testing of the images will be performed. In our project, we considered 80% for training and 20% for testing. After performing the training and testing, it will be given to the MobileNetV2 architecture model for classification. The classification results the output whether it is melanoma or not.

A. **Dataset Preparation**

The dataset used in this methodology is the HAM1000 dataset [10]. We collected various dermoscopic images from different populations from different modalities. The dataset also consists of various types of skin lesion images for classification. The attributes that contained in the dataset are lesion id, image id, dx, dx type, age, sex, and localization. Majorly, the lesion id, image id, age, and localization are majorly considered. Data preparation also involves the data cleaning technique that helps to clean or remove unwanted or duplicate data. Based on the data preparation, the final model analysis will be performed (Fig. 1).

B. **Preprocessing**

The dataset entirely consists of various types of skewed images. The dataset is inherently imbalanced. Preprocessing is a crucial phase in the deep learning model that determines our project's input and output. Preprocessing converts the raw data into the format needed by the user. In this case, to match the given raw size input into the image of input layer, preprocessing is performed. We can also preprocess the data to improve the features of the data and to improve the performance of the model.

C. **MobileNetV2**

MobileNetV2 architecture is a convolutional neural network model. It consists of 53 layers to perform the classification for the given input image. The performance of the algorithm gets improved because of its lightweight architecture. MobileNetV2 has been developed by taking the reference from the MobileNetV1 for improving accuracy. The 2D convolution architecture filters the given inputs into just one output channel. For producing the output, it will be transferred into different channels. The stacked output channels are filtered by using only 1×1 convolution, which is known as point-wise convolution. By differentiating between the depth-wise and point-wise convolution, MobileNetV1 has 28 which is different from the MobileNetV2 architecture. The image size that has to be given for the MobileNetV1 and MobileNetV2

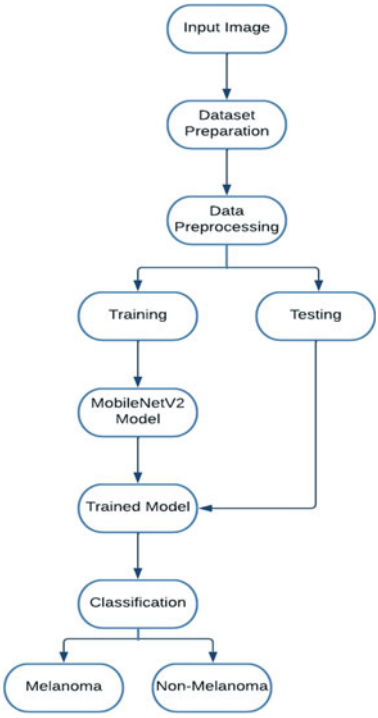


Fig. 1 Methodology of the proposed model

is 224 × 224 × 3 pixels. MobileNetV2 architecture is used for both the classification and for detection of the object (Fig. 2).

D. **Classification**

The classification is the major module for detecting the disease. In this system, along with the MobileNetV2 architecture, we added a convolution layer with activation type ReLU, max-pooling layer, and also set dropout to 0.4. The dataset contains 10,015 images belonging to 7 classes and is partitioned into 6898 images for training 1703 images for testing and 1414 images for validation. The images are preprocessed by setting the rescaling factor to 1. /255, rotation range to 10, and height shift range to 0.02. After the model is trained with the training images, the input image is classified to which type of skin disease. It results the output whether the given input image is melanoma or non-melanoma. By taking the global pooling results from the MobileNetV2 architecture model, the classification is performed.

E. **Flow of Model Execution**

Steps:

Step 1: Collect the HAM10000 dataset and perform the data preparation.

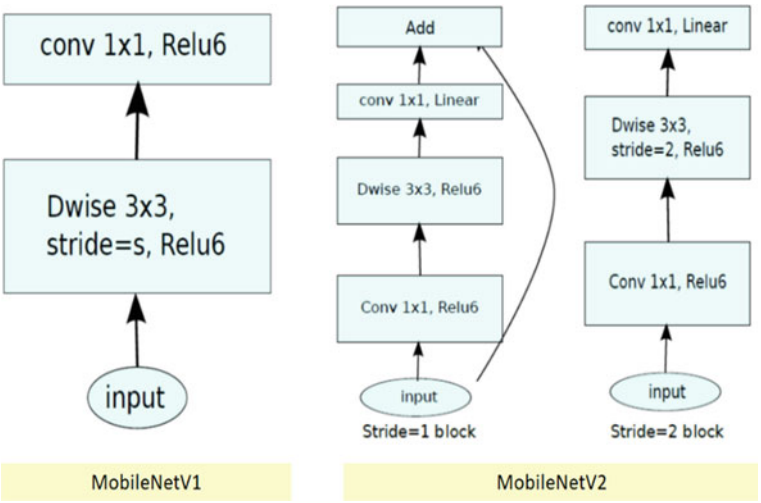


Fig. 2 Enhancement of MobileNetV2 from MobileNetV1 [9]

Step 2: After the preparation of the data, preprocessing will take place. In this stage, the dataset becomes balanced by converting the raw data into the required pixel image format.

Step 3: Dataset partitioning will take place for classifying the different diseases that are present in the dataset.

Step 4: Fixing or setting the pixel level values for the input image. Performing the re-sizing for the input images when their pixels are different.

Step 5: Putting the model through testing and training. 20% for testing, 80% for training.

Step 6: Implementation of MobileNetV2 architecture.

Step 7: Classification will be performed from the global pooling layers. It results the output whether it is melanoma or non-melanoma.

4 Results and Analysis

A. Model Accuracy

The statistic used to assess which model is most effective in finding connections and patterns between variables in a dataset is model accuracy, which is based on the input, or training, data. Model accuracy and model loss are the two parameters that are to be considered to improve the efficiency or accuracy of the model (Fig. 3).

train: The term Train represents the training of the model.

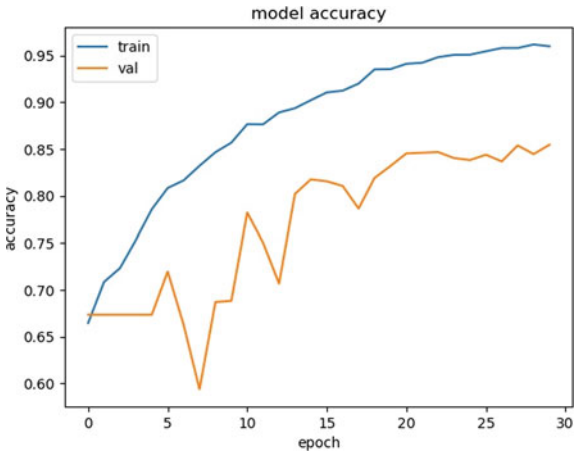


Fig. 3 Model accuracy plotting

val: The term Val represents the validation for the model.

As the n number of epochs is increased, the accuracy of the model is gradually increased and we achieved a val_accuracy of almost 85%. After each training phase, the val_accuracy variable shows how accurately a randomly divided validation set is predicted. It is also observed that the val_accuracy does not decrease and remains constant at about 85%.

B. **Model Loss**

Model loss is the discrepancy between the outcome generated by the machine learning model and the intended outcome (Fig. 4).

C. **Training Loss and Accuracy**

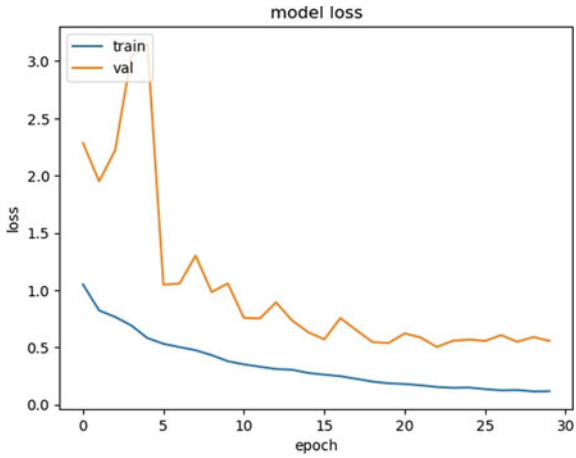


Fig. 4 Model loss plotting

Fig. 5 Training loss and accuracy

Training loss is used to check how a deep learning model fits the training data. Training accuracy is the accuracy that is measured when the images used for both training and testing are same (Fig. 5).

train_loss: To determine whether a model matches the training set of data. The training loss on roughly 6898 training images reaches zero after 30 epochs.

val_loss: Validation loss is a metric used to gauge how well a model is working. After 30 epochs, the validation loss is approximately 0.6.

train_acc: Accuracy in training means that the same images are utilized for training and assessment. After 30 epochs and 6898 training photographs, the training accuracy is approximately 0.95.

val_accuracy: Accuracy of validation denotes the ability to recognize independent images that were not utilized in training. After 30 epochs, the validation accuracy is approximately 0.85.

D. **Web Application Outputs**

Initially, the user will be prompted to upload the skin lesion image. Then, after the uploading image, the backend classification and object detection for the input image will be performed. The classification results will be displayed (Figs. 6 and 7).

5 Conclusion and Future Work

In this work, a model is developed that will detect whether the input image is melanoma or non-melanoma. We used the MobileNetV2 architecture model for classifying and detecting disease. The accuracy for the current model is 85%. Our future work is to improve the algorithm accuracy and to develop the main Web application that will identify the percentage of melanoma that has affected the person. So that

Fig. 6 Output image displaying the melanoma disease

Fig. 7 Output image displaying the non-melanoma disease

the doctors can identify the type of disease and percentage of disease affected in very less time with lesser false results.

References

1. Pereira PM, Thomaz LA, Tavora LM, Assuncao PA, Fonseca-Pinto R, Paiva RP, Faria SM (2022) Multiple instance learning using 3D features for melanoma detection. IEEE Access 10:76296–76309
2. Yao P, Shen S, Xu M, Liu P, Zhang F, Xing J, Xu RX (2021) Single model deep learning on imbalanced small datasets for skin lesion classification. IEEE Trans Med Imaging 41(5):1242–1254

3. Yu Z, Nguyen J, Nguyen TD, Kelly J, Mclean C, Bonnington P, Zhang L, Mar V, Ge Z (2021) Early melanoma diagnosis with sequential dermoscopic images. IEEE Trans Med Imag 41(3):633-646
4. Thurnhofer-Hemsi K, López-Rubio E, Domínguez E, Elizondo DA (2021) Skin lesion classification by ensembles of deep convolutional networks and regularly spaced shifting. IEEE Access 9:112193–112205
5. Rastghalam R, Danyali H, Helfroush MS, Celebi ME, Mokhtari M (2021) Skin melanoma detection in microscopic images using HMM-based asymmetric analysis and expectation maximization. IEEE J Biomed Health Inform 25(9):3486–3497
6. Zhang B, Wang Z, Gao J, Rutjes C, Nufer K, Tao D, Feng DD, Menzies SW (2020) Short-term lesion change detection for melanoma screening with novel siamese neural network. IEEE Trans Med Imag 40(3):840–851
7. Li LF, Wang X, Hu WJ, Xiong NN, Du YX, Li BS (2020) Deep learning in skin disease image recognition: a review. Ieee Access 8:208264–208280
8. Albahli S, Nida N, Irtaza A, Yousaf MH, Mahmood MT (2020) Melanoma lesion detection and segmentation using YOLOv4-DarkNet and active contour. IEEE Access 8:198403–198414
9. Indraswari R, Rokhana R, Herulambang W (2022) Melanoma image classification based on MobileNetV2 network. Procedia Comput Sci 197:198–207
10. https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000

Reinforcement Learning Based Spectrum Sensing and Resource Allocation in WSN-IoT Smart Applications

J. V. N. Raghava Deepthi, Ajoy Kumar Khan, and Tapodhir Acharjee

Abstract There is a tremendous demand for resources due to the different emerging IoT applications that must be fulfilled. The large-scale resource-constrained IoT ecosystem needs a novel and robust technique to manage resources. An optimal resource provisioning in IoT ecosystem has to handle the mapping of request-resource and is hard to achieve as the IoT requests and resources are dynamic and heterogeneous. Therefore, artificial intelligence (AI) has a vital role in the IoT network. As changes are dynamic, global optimization cannot be achieved due to classifiers of the wireless signals and high interference. In this research, spectrum selection is integrated with spectrum access by the use of reinforcement learning with stochastic-based reward measures for efficient resource allocation in WSN-IoT applications. Here, state–action–reward–state–action approach with the Gittins index named SARSA-GI is involved. The role of the state–action–reward–state–action model is developed with an energy-efficient approach for optimizing the channel. Next, the Gittins index is designed to reduce the delay and enhance the accuracy of spectrum access. The simulation results are compared with two state-of-the-art methods which achieve 91.4% of reliability, 88.4% of transmission probability, 93.4% of throughput, 66.8% of collision performance, and 57.2% of signal-to-interference noise ratio.

Keywords Resource allocation · Spectrum sensing · Reinforcement learning · Index model · Internet of Things (IoT)

J. V. N. R. Deepthi (✉) · T. Acharjee
Department of Computer Science and Engineering, Assam University, Silchar, Assam 788011, India
e-mail: deepthijonnalagadda1@gmail.com

A. K. Khan
Department of Computer Engineering, Mizoram University, Aizawl, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_8

1 Introduction

Recently, numerous Internet of Things (IoT) are associated with radio frequency (RF) platform with faster rates than before. An investigation made in [1] revealed that about 50 billion IoT connections are available. In spite of the growing need for wireless IoT applications, the bandwidth of RF spectrum is a significant challenge for IoT devices due to the enormous volume of data they generate. Mostly, these devices are operated on medical, scientific, and industrial, RF bands [2]. Thus, sharing of channels causes serious collision and interference. To overcome these issues, and spectrum scarcity in the development of IoT applications, an efficient spectrum sensing and sharing approach is required. Sensing is the key role for secondary users (SUs) to share the spectrum, since it identifies the prime users (PUs) unused or vacant channel(s), also known as licensed users (LUs). Transmission performance when using a shared network of IoT devices requires careful planning of spectrum sensing to ensure the security of base-level IoT devices. Energy detection (ED) [3, 4], matched-filter detection [5, 6], and cyclic stationary-feature detection [7] are only a few of the spectrum sensing methods that have been studied and implemented in wireless devices as a result of the development of CR technology [8, 9]. This demands for better sensing quality which is possible by increasing the samples of signals received or the strengthening of the signal-to-noise ratio (SNR) received while sharing IoT devices or at secondary units. In contrary to the conventional WSNs, cognitive radio-enabled sensor networks (CRSNs) function with licensed bands [10], spectrum sensing is done periodically, and vacant channels are identified. To accomplish this, cognitive radio-enabled sensor devices (CRSNs) aim should be to sense the licensed channels often to determine the idle and active status of signal in PUs with severely limited interference to PUs. In contrast to CR networks [11], however, CRSNs adopt some of the fundamental constraints of traditional WSNs, such as a shorter network lifetime due to energy constraints. Moreover, SS is the key while designing CRSN with the consideration of energy consumption. When numerous SUs take part in spectrum sensing, more energy is consumed providing less lifetime [12]. Thus, boosting nodes' participation in SS while guaranteeing the accuracy of spectrum sensing [13] is intended to maximize the network's energy efficiency. Additionally, it is seen that CSS not only handles multipath fading and shadow effects, but also improves SS precision. The idea behind the introduction of CSS is to involve several SUs and then integrate their outcome at a fusion center (FC) [14]. Two available CSS strategies are every node take part in CSS, and other is that few nodes perform spectrum sensing. The second option suits when both of their performance is similar. The growing number of wireless nodes, both licensed and unlicensed, is what prompted the research presented in this study. Conventional rigid spectrum allocation policies possibly provide no solution for spectrum scarcity. As a result, decision Support systems (DSS) are implemented to promote dynamic spectrum access and supply adaptable spectrum management for these licensed and unlicensed IoT users. Still, the three paramount considerations to be kept in mind while designing an IoT-based wireless networks are power, cost, and hardware complexity. Since the cost of acquiring

a spectrum resource from the license-based wireless communication system is out of reach for most IoT users, this presents a significant barrier to their widespread adoption. The DSS model suggested in this research has wide applicability across a variety of Internet of Things (IoT) contexts, including "smart" settings like farms, hospitals, cities, power grids, and transportation networks.

The rest of the paper is organized as follows: Sect. 2 reviews a library of existing approaches pertinent to this study. In Sect. 3, we present the proposed method and a brief theoretical overview. In Sect. 4, we discuss the work's potential based on the results of the simulations. In the final section, we make some conclusions and discuss some potential directions for further research.

2 Related Works

Adjustments including traffic prioritization, clear channel evaluation, and the development of a new DSS that does not account for the effects of IEEE 802.15.4 CSMA/CA parameters are some of the many ways researchers have looked to boost spectrum sensing's efficacy in WSN-IoT. In [15], Adaptive channel Access model for real-time operations (ACAMRO) was constructed to make the best use of channels by modifying channel conditions. Throughput was increased compared to the standard IEEE 802.15.4 configuration. The implementation's low memory requirements (less than 20 kB) were offset by the benefit of a very short sequence. In [16], a stochastic model (SM) was developed for unslotted operations of IEEE 802.15.4 networks using low-power, low-data-rate wireless communications. Fading of signal is a common and significant aspect of mobile communications. Based on the fading level, signal-to-noise ratio (SNR) obtains values that change dynamically. IEEE 802.15.4 uses the maximum data rate of successful transmissions between a pair of communication nodes in a mobile channel to estimate the size of the frame. In [17], a general resource allocation model was introduced in which channel allocation and power of SU were optimized when the location of SU were known. In [18], evolutionary algorithms were used to dynamically control intrusion in a cognitive radio network based on WSN-IoT in order to optimize the network's performance by adjusting the level of interference between the primary user (PU) and the secondary user (SU). In [19], reinforcement learning (RL) strategy was suggested which had the ability to determine the closest route by utilizing opportunistic routing. In WSN, cooperative learning permits the sharing information among neighbor nodes such that the learning process is accelerated and thus obtains longer network lifetime. A multi-tasking learning (MTL) was suggested in [20] among the relational network (RN). Its general prediction method predicted the views of TV drama. In [21], congestion of backhaul network traffic was reduced by predicting the requests made by the user with the help of file popularity and its patterns. However, this method did not address problems associated with the placement of the secondary base station, which had a direct impact on interference between the primary and secondary systems and on the allocation of

common channel resources. Moreover, another drawback is dynamicity, and ambiguity in modern operating platform; for instance, IoT ecosystem are not properly addressed. This study presents a new method for dealing with the aforementioned issues.

3 System Model

To model the entire scenario of resource management, an effective service arrival process is required and the desired scheduling policy has to be formed for efficient resource mapping based on the requests made. To collect information about the radio environment and store the gathered status/parameter, spectrum is shared with the help of several agents deployed in each node. The allocation of spectrum is performed with a novel method, and reinforcement learning supports the procedure incorporated in agents to properly decide based on the allocated resources and spectrum management. This proposed model comprises of several components at various stages, which includes service request interface, a repository with available resources, IoT resource pool, a decision support system SARSA-GI and optimized and prioritization unit. Figure 1 depicts the framework of the proposed model.

3.1 Service Request Interface

One server with multiple resources is synthesized from a pool of resources that includes a wide variety of types of servers. It is possible for a service request to stake a claim on any number of the pool's available resources. A queue model is useful for representing these requests in an IoT ecosystem. A user's request for a network

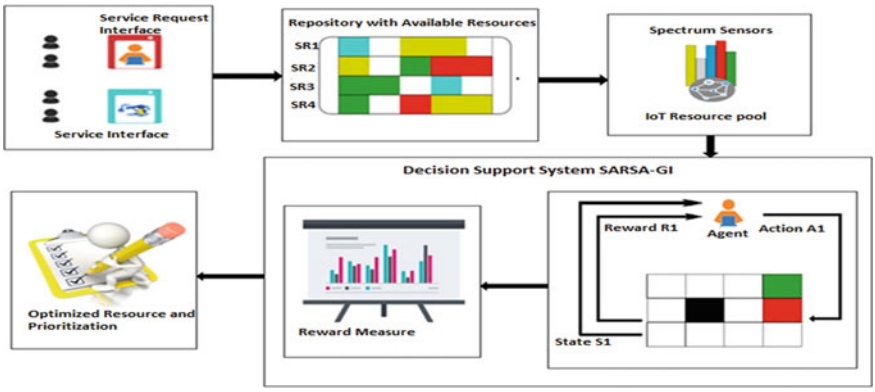


Fig. 1 Overall architecture of proposed model

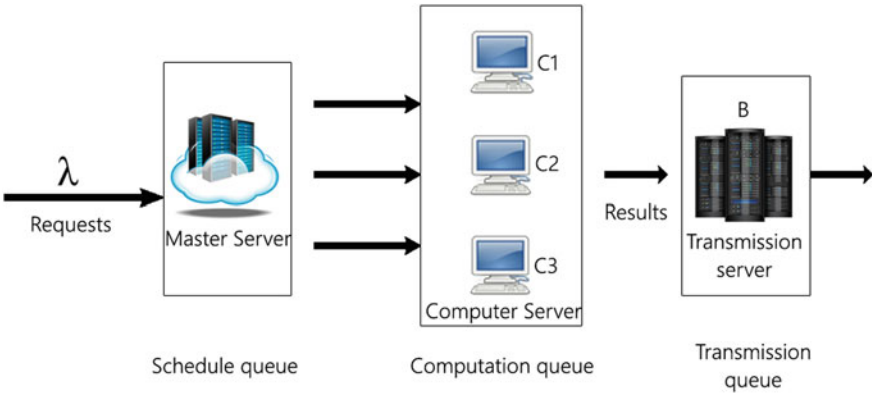


Fig. 2 User request queue model

service is mapped out in a queuing model to determine how scarcely those resources are distributed. The service requested is accomplished which equals the completion of the resource in queue model as shown in Fig. 2.

Notations used are λ, $s(t)$, $c(t)$, $\beta(t)$ which denotes requests, master server, computing server, and transmission server, respectively.

For maintaining the stability of the queue, $\beta(t)$ has to be less than $< s(t)$. Schedule queue response time for the request is given as $T(\text{sing}) = \frac{1/s(t)}{1-\beta(t)/s(t)}$. As several computational queues follow the similar service procedure denoted by p_i which is the request probability given to ith computational queue. Hence, $\beta(t) = p_i{}^* \beta(t)$ and $\sum_{i=1}^{n} p_i = 1$. At the computational queue, the service time is based on the assigned computational resource $c(i)$ t and the size of the task F.

According to the results of the preceding analysis, in the case of a single service, the equilibrium response time equals the sum of the times at which each of the three steps in the process takes place and is given by

$$\begin{aligned} T_{\text{tot}}(t) &= T_{\text{sch}}(t) + T_{\text{comp}}(t) + T_{\text{tran}}(t) \\ &= \frac{1/s(t)}{1-\beta(t)/s(t)} + \sum_{i=1}^{n} \frac{p(i)F/c(t)}{1-p(i)\beta(t)/c(t)} + \frac{d/b(t)}{1-\beta(t).d/b(t)} \end{aligned} \tag{1}$$

The response time minimization is computed as,

$$\begin{aligned} \text{Minimize } & (s(t), c(t), \ldots, cn(t), b(t) \\ & \beta(t) < s(t) \\ & P_i(t)F < c, \quad \in i = 1, 2, 3 \ldots, n \\ & \beta(d) < b(t) \\ & z = [\alpha s(t) + \beta \text{B} b(t)]\, t \ \leq c_{\max} \end{aligned}$$

The first three constraints represent the minimum requirements for resource in every queue, and hence z is the budget constraint. When more budget is assigned on computational resource, it will be less for bandwidth and schedule and hence leads to unbalanced allocation of resources.

3.2 Repository with Available Resource

The information gathered by a swarm agent from many channels or nodes is stored here, along with descriptive details about the sensed channel, to aid in subsequent decision-making. Assume that the IoT resource pool {RP} consists of n dissimilar elements, or types, as $\{RP_1, RP_2, RP_3, \ldots, RP_n\} \in$ RP. Every request has its own unique resource requirements as $SR(i)) = rp(i_1), rp(i_2), rp(i_3), \ldots, rp(ij)$ for$(j < n)$. The mapping of resource allocation is given by $\{RS\} \rightarrow \{rs(i)\}$. The aim here is to increase QoS(max(QoS(SRi))) and reduce energy consumption(min(EC(SRi))) of every request. Hence, cumulative maximized QoS and minimized energy consumption is achieved.

$$\text{Resource } (Sr_i) = A(SR) * (SR) \tag{2}$$

$$(SR_i) = \max\left(\sum_i qos(SR_i) + \sum_i 1/EC(SR_i)\right) \tag{3}$$

where $A()$ is an allocation function for SR_i according to the specified policy (SR(i)), and resource (SR_i) is the resulting mapping of resources for request i.

3.3 State–Action–Reward–State–Action–Gittins Index (SARSA-GI) Method

The agents behavior is mapped from state space S to action space A according to the policy, which is written as π: $st \in S \rightarrow$ at $\in A$. Time for action A includes the power level and the spectrum sub-band. Moreover, action at A relates to the power level selection and spectrum for the links of IoT communication. To maximize expected future discounted rewards over the long term, an optimal policy is obtained through reinforcement learning techniques Gt. The value of the state for a given state and action (st, at), Q(st, at), of π is stated as Gt when an action is taken at $\in A$ and thereafter at π. For a given state, this state value helps in measuring the quality of action. By simply estimating an action when given the state values Q(st, at), a better policy can be obtained.

$$A(t) = \arg\max Q(\text{st}, a) \tag{4}$$

The optimum policy with Q-values $Q*$ is determined with no knowledge of system dynamics which is based on the update equation given as,

$$Q_{(\text{new})}[\text{st,at}] = Q_{\text{old}}(\text{st,at}) + \alpha[rt + 1 + \beta_{\max} Q_{\text{old}}(\text{st,at}) - Q_{\text{old}}(\text{st,at}) \tag{5}$$

$$\text{Loss}\ (t) = \sum_{(\text{St,at}) \in D}^{n} (Y - q(\text{st,at}) \tag{6}$$

$$\text{Loss}\ (t) = rt + \max\ Q_{\text{old}}(\text{st,a,t}) \tag{7}$$

The reward function is described as,

$$R(t) = \alpha c \sum_{m=0}^{n} cm + \alpha d + \sum_{k=i,j}^{n} c(k, j)(To - Ui) \tag{8}$$

By using the Gittin's index, the reward function can be formulated as,

$$V(i) = \frac{\sum_{t=0}^{T-1} \mu^t R[z(t)]z(n) = i}{[\sum_{t=0}^{T-1} \mu^t]z(n) = i} \tag{9}$$

where $z(n)$ is a stochastic process, $R[z(t)]$ is the reward associated to the discrete state i, μ is the probability that the stochastic process does not terminate.

$$X(c) = \sum_{k=x(n)}^{n} xP\left\{X = \frac{x}{c}\right\} \tag{10}$$

If the probability of survival $\beta(i)$ is depends on the state I, then the maximum discounted total reward per chance of termination is given as,

$$\Omega\,(I) = \frac{R(i)}{Q(i)} \tag{11}$$

where

$$R(i) = \sum_{t=0}^{T-1} R[z(t)] \tag{12}$$

$$Q(i) = 1 - \prod_{t=0}^{T-1} \gamma[z(t)] \tag{13}$$

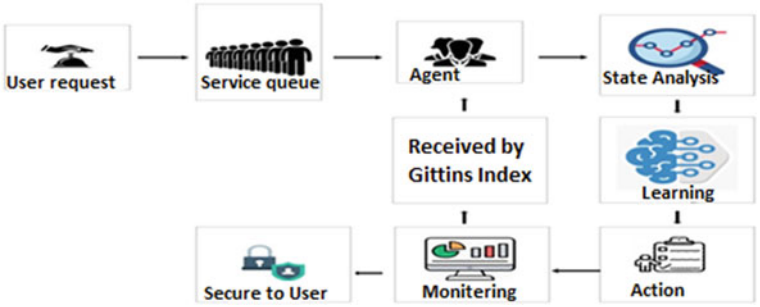


Fig. 3 Architecture of proposed SARSA-GI method

At some point in the future, at time t, the agent performs actions such as deciding which messages and sub-channels will be broadcast. Each message's action space has a dimension equal to NRB + 1. When an agent's action starts at the first NRB, the message is quickly broadcast in the appropriate sub-channel. Otherwise, the message will not be sent at that time (Fig. 3).

3.4 Optimized Resource and Prioritization

The repository defined by an array of structures is denoted by the abbreviation RP. Each child script keeps track of information about its users in a separate data structure called Rs = {0, 1, 2, …, m}, where m is the number of parameters. Data having high data rate for greatly demanded by users which means that lesser number of users are not allocated with channels of higher data rates. Unlike video or images, which require a higher data rate channel, textual data can be transmitted over a lower-rate channel. When higher capacity channel is allocated for text data, resources allocation strategy is poor as resources are wasted. This method always helps in assigning low-data-rate channel for users where less quality service is required. The channel estimation is misunderstood because the basic channel sensing algorithm is used to verify channel allocation. Data channels are allocated appropriately and on-time through the use of a priority-based reserved allocation algorithm and reinforcement learning.

Algorithm SARSA-GI

Input-Available resource in repository
Output- Optimized resources
Start
Perform the Mapping process (S,A)
$K = \pi : st \in S \rightarrow at \in A$
Perform the Reward calibration (Gt)
$Gt = Q(st, at)$
If $Gt < Q$
then
perform action process A(t)
$A(t) = \arg\max Q(st,a)$
Find the optimal old and new Q value
Qnew, Qold

End if
Compute the loss function for optimal value
$Loss(t) > x(t)$
$x(t) = f(t) + f'(t) + R(t)$
Include the gittin's index values
$V(i) \leftarrow z(k)$
$Z(k) > X(c)$
Terminate the process

4 Performance Analysis

Comparison of proposed state–action–reward–state–action with Gittin's index (SARSA-GI) method is done with the existing methods such as adaptive channel access model for real-time operations (ACAMRO) [15] and stochastic model (SM) [16] using a wide range of parametric measures like reliability, throughput, collision performance, SINR, and transmission probability.

- **Reliability**

The network's reliability is measured by how likely it is that a given source node will be disconnected from the rest of the network, given a link failure probability of *p*.

Table 1 presents the comparison of existing ACAMRO and SM with proposed SARSA-GI method for reliability.

Figure 4 plots the reliability of the existing ACAMRO and SM with proposed SARSA-GI. The number of iterations is shown on the X-axis, while the percentages of reliability are shown on the Y-axis. When compared to the existing method, the proposed method outperforms ACAMRO by 14.6% and SM by 3%.

Table 1 Analysis of reliability

Number of iterations	ACAMRO	SM	SARSA-GI
100	70	81	86
200	74	85	90
300	78	90	92
400	80	92	94
500	82	94	95

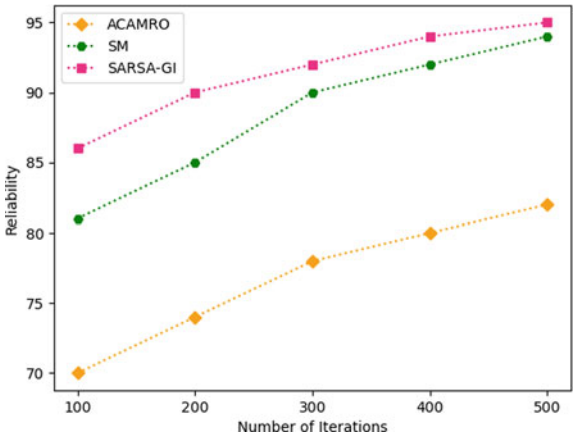


Fig. 4 Comparison of reliability of ACAMRO and SM with SARSA-GI

- **Transmission probability**

This is the total time slots that are busy occurring with probability P-busy and time clots that are idle occurring with probability *P*.

Table 2 presents the comparison of existing ACAMRO and SM with proposed SARSA-GI method for transmission probability.

Figure 5 plots the transmission probability of the existing ACAMRO and SM with proposed SARSA-GI. Transmission probabilities are plotted as a percentage against the number of iterations on the *Y*-axis. By contrast, the proposed method outperforms ACAMRO by 11.8% and SM by 1.2%, while the existing method only manages 76.6% and 87.2%.

Table 2 Analysis of payload transmission probability

Number of iterations	ACAMRO	SM	SARSA-GI
100	70	82	85
200	74	85	85
300	77	88	89
400	80	90	91
500	82	91	92

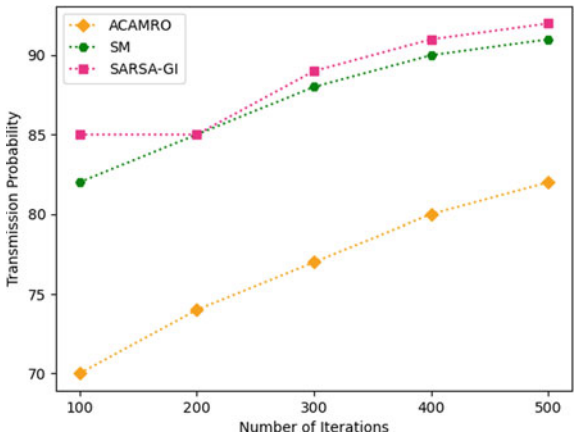


Fig. 5 Comparison of transmission probability of ACAMRO and SM with SARSA-GI

- **Throughput**

Rate of success is defined as the percentage of a platoon that reaches its objective within a given time limit. To determine the productivity of a single platoon's members, we will use the symbol T_i to denote the amount of time elapsed. Let us call the final determination of channel allocation x_{ij}, with $x_{ij} = 1$. Then, the maximum output can be represented as

$$x_{ij} = \max \sum_{i=1}^{m} T_i \tag{14}$$

Table 3 presents the comparison of existing ACAMRO and SM with proposed SARSA-GI method for throughput.

Figure 6 compares the throughput of the existing ACAMRO and SM with proposed SARSA-GI. Throughput values, expressed as a percentage, are plotted against the number of cycles (represented by the *X*-axis) that they were calculated. The existing method has a 78.2% and 83.2% success rate, while the proposed method has a 15.2% higher success rate than ACAMRO and a 10.2% higher success rate than SM.

Table 3 Analysis of throughput

Number of iterations	ACAMRO	SM	SARSA-GI
100	71	77	87
200	74	80	92
300	80	82	94
400	82	87	96
500	84	90	98

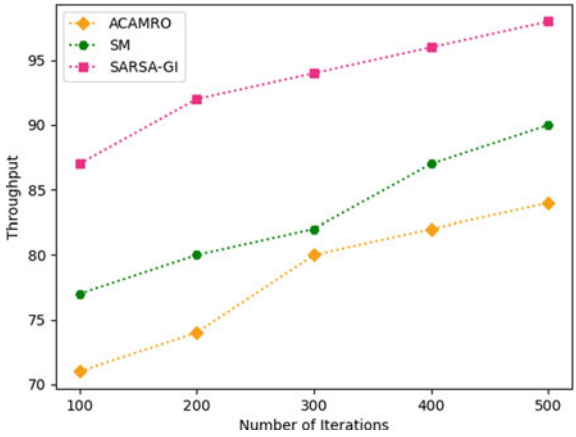


Fig. 6 Comparison of throughput of ACAMRO and SM with SARSA-GI

- **Collision performance**

When a platoon member has a chance of communicating with someone else in the same platoon, with a probability of p_s, the probability of communication with someone in a different platoon is 1, and the collision probability is p_o. Finally, we can express P_c, the probability of collision, as

$$P_c = \alpha p_s + (1 - \alpha) p_o \tag{15}$$

Table 4 presents the comparison of existing ACAMRO and SM with proposed SARSA-GI method for collision performance.

- **SINR**

An individual platoon member's signal-to-interference-and-noise ratio (SINR) is determined by subtracting the SINR at the transmitter vehicle, measured at its current power level, from the SINR at the receiver vehicle, measured at its target power level, as given by

$$\text{SINR} = \frac{P_{ij}}{P_{\text{noise}} + \sum P_{kj}} \tag{16}$$

Table 4 Analysis of payload collision performance

Number of iterations	ACAMRO	SM	SARSA-GI
100	75	70	60
200	76	74	64
300	78	76	68
400	80	79	70
500	82	80	72

Table 5 Analysis of SINR

Number of iterations	ACAMRO	SM	SARSA-GI
100	65	60	50
200	66	64	54
300	68	66	58
400	70	69	60
500	72	70	64

where P_{noise} is the signal strength of the Gaussian noise, P_{ij} is the received signal strength due to interference, and P_{kj} is the received signal strength due to fading.

Table 5 presents the comparison of existing ACAMRO and SM with proposed SARSA-GI method for SINR (Fig. 8).

Figure 7 plots the collision performance of existing ACAMRO and SM with proposed SARSA-GI. The *X*-axis represents the total number of cycles, while the *Y*-axis displays the percentage collision rate. Existing methods get 68.2% and 65.8%, respectively, while the proposed method gets 11% better results than ACAMRO and 8.6% better results than SM.

Table 6 provides an overview comparison of existing ACAMRO and SM with proposed SARSA-GI method.

Figure 9 displays a comparison of the parameter values. Parameters used in the analysis (*X*-axis) and numerical results (*Y*-axis) are presented as percentages.

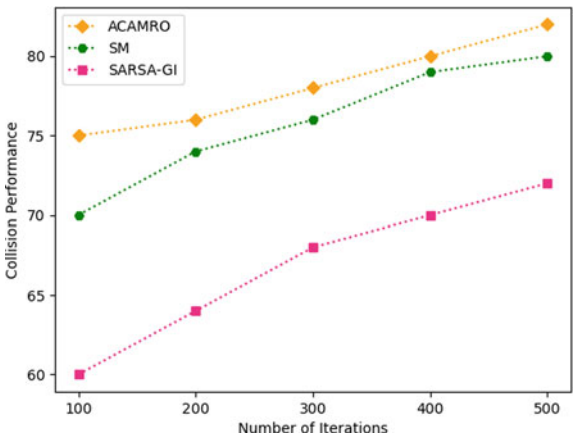


Fig. 7 Comparison of collision performance of ACAMRO and SM with SARSA-GI

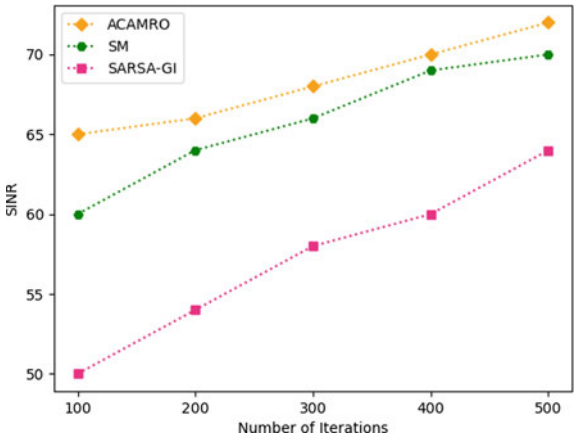


Fig.8 Comparison of signal-to-interference noise ratio (SINR) of ACAMRO and SM with SARSA-GI

Table 6 Overall comparative analysis

Parameters	ACAMRO (%) [15]	SM (%) [16]	SARSA-GI (%)[proposed]
Reliability	76.8	88.4	91.4
Transmission probability	76.6	87.2	88.4
Throughput	78.2	83.2	93.4
Collision performance	78.2	75.8	66.8
SINR	68.2	65.8	57.2

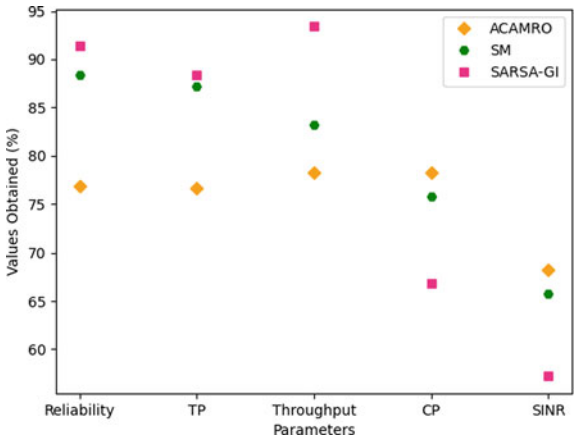


Fig. 9 Overall analysis between ACAMRO and SM with SARSA-GI

5 Conclusion

By the use of allocation policies, IoT resources are managed based on the requests made by the user. A distributed algorithm is introduced in this paper which obtains the optimal location and channel for managing resource allocation and intrusions in IoT sensor networks. This proposed model enhances the secondary system's capabilities while maintaining the primary system's efficiency. The platform's potential for different resource allocation strategies was investigated using Gittins' index and reinforcement learning, and the best approach for doing so was chosen. Based on the results of the simulations, it is clear that the proposed algorithm outperforms ACAMRO and SM significantly in terms of reliability (91.4%), transmission probability (88.4%), throughput (93.4%), collision performance (66.8%), and signal-to-interference noise ratio (57.2%). The future work is to examine the effects of PHY and MAC layers with challenging channel conditions and different collision features by applying reinforcement learning.

References

1. Diro A, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for internet of things. J Future Gener Comput Syst 82:1–11
2. Diro A, Reda H, Chilamkurti N (2018) Differential flow space allocation scheme in SDN based fog computing for IoT applications. J Ambient Intell Human Comput 22:1–11
3. Hossain E, Niyato D, Han Z (2009) Dynamic spectrum access and management in cognitive radio networks. Cambridge University Press, Cambridge
4. Wang N, Gao Y, Zhang X (2013) Adaptive spectrum sensing algorithm under different primary user utilizations. IEEE Commun Lett 17(9):1838–1841
5. Tandra R, Sahai A (2005) Fundamental limits on detection in low SNR under noise uncertainty. In: International conference on wireless networks, communications and mobile computing, pp 464–469
6. Rauniyar A, Shin SY (2015) Multiple antenna-aided cascaded energy and matched filter detector for cognitive radio networks. Int J Distrib Sens Netw 11(9):1–9
7. Khambekar N, Dong L, Chaudhary V (2007) Utilizing OFDM guard interval for spectrum sensing. In: Proceedings of the IEEE wireless communication and networking conference, pp 38–42
8. Zeng Y, Liang YC, Hoang AT, Zhang R (2010) A review on spectrum sensing techniques for cognitive radio: challenges and solutions. EURASIP J Adv Signal Process, pp 1–15
9. Liang YC, Chen K, Li GY, Mahonen P (2011) Cognitive radio networking and communications: an overview. IEEE Trans Veh Technol 60(7):3386–3407
10. Cheng H, Su Z, Xiong N, Xiao Y (2016) Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model. Inf Sci 329:461–477
11. Larsson E, Skoglund M (2008) Cognitive radio in a frequency-planned environment: some basic limits. IEEE Trans Wireless Commun 7(12):4800–4806
12. Axell E, Leus G, Larsson EG, Poor HV (2012) Spectrum sensing for cognitive radio: state-of-the-art and recent advances, IEEE Signal Process Mag 29(3):101–116
13. Noh G, Wang H, Jo J, Kim BH, Hong D (2011) Reporting order control for fast primary detection in cooperative spectrum sensing. IEEE Trans Veh Technol 60(8):4058–4063
14. Vien QT, Stewart BG, Tianfield H, Nguyen HX (2013) Efficient cooperative spectrum sensing for three-hop cognitive wireless relay networks. IET Commun 7(2):119–127

15. Gezer A , Okdem S (2020) Improving IEEE 802.15. 4 channel access performance for IoT and WSN devices. Comput Electr Eng
16. Okdem S (2017) A real-time noise resilient data link layer mechanism for unslotted IEEE 802.15. 4 networks. Int J Commun Syst
17. Xue T, Dong X, Shi Y (2017) Resource-allocation strategy for multiuser cognitive radio systems: location-aware spectrum access. IEEE Trans Veh Technol 66(1):884–889
18. Khan H, Yoo SJ (2015) Dynamic interference control in OFDM-based cognitive radio network using genetic algorithm. Taylor & Francis
19. Bhorkar AA, Naghshvar M, Javidi T, Rao BD (2012) Adaptive opportunistic routing for wireless ad hoc networks. IEEE/ACM Trans Netw
20. Trzcinski T, Rokita P (2017) Predicting popularity of online videos using support vector regression. IEEE Trans Multimedia 19(11):2561–2570
21. Tang F, Fadlullah ZM, Mao B, Kato N (2018) An intelligent traffic load prediction based adaptive channel assignment algorithm in SDN-IoT: a deep learning approach, IEEE Internet Things J

Planning and Construction of a Quantum Model of the SHA-1 Algorithm Using IBM's Qiskit

Sandip Kanoo, Sumit Biswas, and Prodipto Das

Abstract The hash produced by a cryptographic hash function is inherently impossible to reverse-engineer or duplicate. You can protect yourself from hacking attempts more effectively with this. In a quantum environment, attacks on information systems would look significantly different, calling for a complete overhaul of current cryptographic practices. For the purpose of constructing a quantum-based Secure Hash Algorithm (SHA-1) from the ground up, a complete guide is provided to the design of quantum addition along with the associated circuits of XOR, AND, NOT, OR, MOD, CARRY, and SUM. To boost the efficiency of the standard SHA-1 algorithm, the q-SHA-1 is created in a Qiskit-based quantum environment. Experimental results from a wide range of conditions will be compared to evaluate the relative efficacy of the classical and quantum SHA-1 algorithms. However, it has been discovered that the execution time is highly dependent on the particular CPU type that is being used. A more powerful processor ought to cut down on the amount of time needed for the overall execution.

Keywords SHA · Hash · Data integrity · Massage authentication · Digital certificate · Quantum computing

1 Introduction

Hash functions include cryptographic hash functions. It accepts any size input string block and outputs a fixed-size bit of string. Different cryptographic hash values prevent any unintentional or deliberate alteration to the material. The hash value is frequently referred to as the message digest, and the data that need to be encoded as the message. The SHA algorithm is used for data integrity, message authentication,

S. Kanoo · S. Biswas (✉) · P. Das
Department of Computer Science, Assam University, Silchar, India
e-mail: biswassumit812@gmail.com

P. Das
e-mail: prodipto.das@aus.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_9

and digital certificates. SHA, a U.S. Federal Information Processing Standard (FIPS) developed by N.I.S.T. and used with digital signature applications, is a fingerprint that identifies the data [1]. A valid file needs to be downloaded from peer-to-peer (P2P) servers and networks in order to accommodate the growing number of people using the Internet. Since there are many files with the same name in different locations, it can be challenging to locate the original, and hence, message digests are crucial in these kinds of downloads. These kinds of files could be attached with data authentication codes that demonstrate that the source is authentic; otherwise, a warning that the origin of this is not valid is displayed.

Hash function applications:

- Message authentication: used to determine whether a message has been altered.
- Digital signatures: using a private key to encrypt the digest.
- Password storage: Hackers cannot obtain a password from storage after comparing the password's digest to that in the storage.
- Key generation: To avoid brute-force assaults, the key can be produced using the passphrase's digest and made computationally expensive
- Pseudo random number generation: Creating pseudorandom numbers by repeatedly hashing a seed value.
- Intrusion detection and virus detection: maintain and verify the system's file hashes.

The cryptographic hash function known as SHA-1, which stands for Secure Hash Algorithm 1, produces a hash value that is 160 bits in length. The phrase "message digests" means this hash value. Typically, this message digest yields a 40-digit hexadecimal number as the final output. This is a Federal Information Processing Standard in the United States, developed by the NSA. As of 2005, SHA-1 has been considered insecure. By 2017, major browsers from industry giants including Microsoft, Google, Apple, and Mozilla no longer supported SHA-1 SSL certificates.

This work is organized such that the reader can gain the necessary context and intuition to grasp the core ideas. To begin, a conceptual introduction to classical and quantum cryptography is provided. The procedures and tests that were run are then outlined. Details of our experiments outcomes are provided after their methodologies are described. A descriptive analysis of the outcomes follows an examination of the experiments. Last but not least, the authors offer their final thoughts.

2 Related Works

Since 2005, the Secure Hash Algorithm 1 (SHA-1) has been discredited as an effective security measure against well-resourced adversaries, and as of 2010, various groups have advocated for its replacement [2].

In 2011, the National Institute of Standards and Technology (NIST) of the United States Government formally deprecated the usage of SHA-1, and in 2013, it barred

its use for the creation of digital signatures. As of the year 2020, it will be possible to launch chosen prefix attacks against SHA-1 [3].

It is strongly suggested that SHA-1 be removed from goods quickly afterward feasible and that SHA-2 or SHA-3 be used in its place. It is imperative that SHA-1 be replaced everywhere it is utilized for the creation of digital signatures [4].

In 2017, the acceptance of SHA-1 SSL certificates was discontinued by all of the main online browser vendors. In February of 2017, CWI Amsterdam and Google revealed that they have successfully conducted a collision attack on SHA-1 by uploading two PDF files that are very different from one another but produce the same SHA-1 hash [5].

The use of SHA-1 for HMAC is still safe [6]. Support for SHA-1 code signing will be removed from Windows Update on August 7, 2020, according to Microsoft [7].

A great number of works have been documented in the most recent advancements in post-quantum cryptography (PQC). A variety of research projects, such as PQCrypto, SAFEcrypto, CryptoMathCREST, and PROMETHEUS, in addition to various standardization initiatives, have been addressing the topic of post-quantum cryptography, which is currently a popular research topic. These initiatives have been laid out at a variety of different levels. Above all else, the National Institute of Standards and Technology (NIST) of the United States Government is working on developing post-quantum public-key cryptosystems. To date, there have been two phases of this project, and it is anticipated that the first standard drafts will be delivered between the years 2022 and 2024 [8].

Despite the prevalence of quantum computers, traditional cryptographic algorithms such as codes, hashes, lattices, and multivariate techniques remain secure; there were challenges in cryptography in the 1990s decade brought about by the invention of the Shor and Grover algorithm. These challenges allowed popular algorithms such as RSA (1978), Diffie–Hellman (2002), and Elliptical curve (1985) to be cracked [9].

A network platform that is immune to quantum attacks is essential at this point in time. In this vein of thought, other than the work being undertaken by NIST, the Internet Engineering Task Force (IETF) has published a Request for Comment (RFC) that can give correction for quantum inertia to the Internet Key Exchange that is extensively used (IKE). In a similar vein, both the International Organization for Standardization (ISO) and the United States Federal Information Processing Standards (FIPS) have developed programs that check to see if cryptographic modules are used in a network in a way that is accurate and trustworthy. The International Organization for Standardization is participating in the Horizon 2020 project known as Post-Quantum Cryptography for Long-Term Security (PQCRYPTO). The Federal Information Processing Standards Organization (FIPS) has published a draft roadmap for post-quantum hardware/software module assessments [10].

PQC algorithms are currently the primary area of concentration for research groups working in the field of quantum cryptography. The post-quantum algorithms use super singular isomorphy, ring learning with errors, and coding as its quantum-resistant cryptographic primitives [11].

3 Cryptography

Cryptography is the study of techniques for making private communications between two parties, where either may be completely confident that the other will not be able to intercept or change the message in transit. Some of the most fundamental tools in contemporary cryptography are those that ensure the integrity, authenticity, non-repudiation, and privacy of data. Online commerce, ATMs, and computer passwords are just a few of the many practical uses for both new and old cryptographic methods. Using a variety of machine learning methods, cryptography encrypts otherwise intelligible material. Sender and recipient both have access to the same set of decryption keys [12]. Encryption, decryption, key generation, authentication, digital signatures, and other cryptographic procedures can all be described by cryptographic algorithms, which consist of a set of rules, actions, or procedures or a number of mathematical equations that have been established. There are two different kinds of cryptographic algorithms, i.e., classical cryptographic approaches and quantum cryptographic approaches.

3.1 Classical Cryptography

Mathematics is the foundation of classical cryptography, which exploits the computational difficulties of factoring huge numbers. The mathematical problem used in classical cryptography is very difficult, and this is what gives it its security (for instance, the factorization of large numbers).

The goal of classical cryptography is to allow for the transmission of plaintext data through insecure channels by encoding it into a different format called cipher text. To regulate the encipherment process, a string of data called a "key" is utilized. In this method, trust in the key's possession by the sole authorized recipient is assumed. Classical cryptography is of two types:

- Symmetrical Cryptography: Because the same key is used for both encryption and decryption, it is the simplest form of cryptography (Fig. 1).

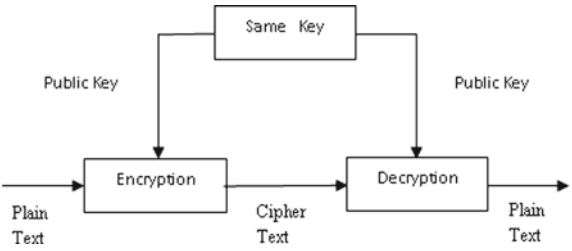


Fig. 1 Symmetric key cryptography encryption and decryption procedure

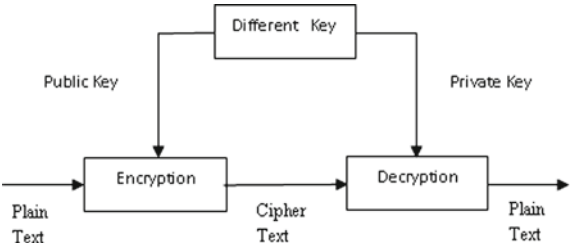


Fig. 2 Asymmetric key cryptography encryption and decryption procedure

- Asymmetrical Cryptography: It is a somewhat intricate cryptographic approach, with separate encryption and decoding keys. Private keys are never exchanged between parties in asymmetric cryptography. In its place, they rely solely on public keys. This makes it more secure and private for users to have private conversations (Fig. 2).

3.2 Quantum Cryptography

Secure data transfer between sender and recipient is made possible by quantum cryptography, which is founded on quantum physics events. It is safe to say that quantum cryptography will have a profound impact on the future of network security [13].

The field of quantum cryptography represents the cutting edge of encryption. It is based on Heisenberg's uncertainty principle and the idea behind polarization of photons, two tenets of quantum mechanics [13].

According to Heisenberg's uncertainty principle, it is possible to prevent yourself from knowing two linked physical qualities by measuring just one of them. To be more specific, all subsequent measurements are affected by the initial decision of which direction to measure in. When light passes through a filter that is aligned vertically, some of it is absorbed, and the remaining light is polarized along the vertical axis. In the next step, a filter with an angle of tilt q absorbs some of the polarized light and lets the rest pass through, depolarizing the light. Polarization of light can be expressed in terms of a basis, which is a pair of orthogonal polarization states like horizontal and vertical [14].

4 Quantum Computing

Quite simply, a quantum computer is any machine with the processing power to perform quantum calculation. A qubit, also known as a quantum bit, is the equivalent of a classical bit in the context of quantum physics. Qubits are used to store the information that is processed by quantum computing. The states of the qubits are

represented as the binary digits $|0\rangle$ and $|1\rangle$, respectively. It is possible for the qubits to hold the values 0 and 1 at the same time. The following characteristics of quantum states are utilized in the construction of quantum computers:

- Superposition:—Quantum systems are capable of coexisting in two distinct states at the same time.
- Entanglement:—It is a phenomenon of quantum mechanics that allows one to characterize the state of entangled particles by making reference to the other particles.
- Interference:—Controlling the likelihood of qubits transitioning into a given measurement condition is the core concept behind quantum computing.

Quantum computing uses superposition, interference, and entanglement to perform computations. Quantum computers conduct these calculations (Table 1).

The value of two bits can be stored in a single qubit. The values of up to four bits can be held by just two qubits. This transforms n qubits into 2^n bits (Table 2).

Table 1 Comparison between bits and qubits

Bits	Qubits
Bits are either 0 or 1	Qubits can be 0, 1, or superposition of 0 and 1
Bits are initialized as 0 or 1	Qubits are initialized to 0
Representation of bits is either 0 or 1	Qubits are represented by vector of two dimensions
In classical, bit state is deterministic	In quantum, qubit state is probabilistic

Table 2 Comparison of size between bits and qubits

Number of qubits	Number of bits	Memory size
1	2	2 bits
2	4	4 bits
3	8	1 bytes
4	16	2 bytes
5	32	4 bytes
6	64	8 bytes
7	128	16 bytes
8	256	32 bytes
9	512	64 bytes
10	1024	128 bytes
11	2048	256 bytes
12	4096	512 bytes
13	8192	1 kb

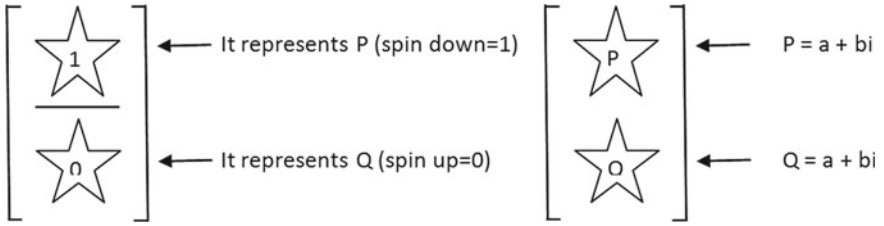


Fig. 3 Representation of qubits

4.1 Qubit Representation

The qubit states can be represented in vector form. These vector forms are:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = (\text{read as Ket 1}) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle (\text{read as Ket 0})$$

$|1>$ is also called spin up or qubit up state, and it is equivalent to classical bit 1.

$|0>$ is also called spin down or qubit down state, and it is equivalent to classical bit 0 (Fig. 3).

If we take the square magnitude of P and Q and sum up them, it will be always 1, i.e., $|P|^2 + |Q|^2 = 1$.

5 Implementation of Quantum SHA-1

For the sake of dealing with quantum computers on the level of individual circuits, pulses, and algorithms, this article makes use of an open-source software development kit called Qiskit. It's designed to help developers create, edit, and test quantum programs that can be executed on IBM's Quantum Experience prototype devices or locally on computer simulators. These programs can also be performed on IBM's Quantum Experience. In order to implement the quantum SHA-1 algorithm on our local computer, we make use of the IBM QASM simulator. For scientific computing, Anaconda is used, and Jupyter is used for interfacing with Qiskit.

The input taken for this algorithm consists of different combinations of text, numbers, and special characters of different lengths. The goal is to examine the difference in output time caused by variations in input length. The same input is executed multiple times to visualize the difference in execution time.

The algorithm used for implementing quantum SHA-1 using QISKIT is given below:

Step 1: Take the input string and convert into its corresponding equivalent binary code.
Step 2: **Add Padding**:

$\text{ap} = \text{dig} + '1'$
$while(len(ap)\%512 != 448:$
$ap += '0'$

Step 3: **Append length**:
Count the total length of the message and convert it into binary equivalent.

$ap_{length} = msg_{length} mod 2^{64}$
$tot_{length} = ap + ap_{length}$

Step 4: **Divide the input into 512 bit blocks**:

$n \rightarrow 512$
$Chunks = [tot_{length}[i: i + n] for\ i\ in\ the\ range(0, len(tot_{length}), n]$

Step 5: **Initialize 5 chaining variables and 4 additive constants**:
Five chaining variables

$cv0 = '01100111010001010010001100000001'$
$cv1 = '11101111110011011010101110001001'$
$cv2 = '10011000101110101101110011111110'$
$cv3 = '00010000001100100101010001110110'$
$cv4 = '11000011110100101110000111110000'$
$f1 = cv0$
$f2 = cv1$
$f3 = cv2$
$f4 = cv3$
$f5 = cv4$

Four constants K, one for each round

$ck0 = '01011010100000100111100110011001'[0 \leq t \leq 19]$
$ck1 = '01101110110110011110101110100001'[20 \leq t \leq 39]$
$ck2 = '10001111000110111011110011011100'[40 \leq t \leq 59]$
$ck3 = '11001010011000101100000111010110'[60 \leq t \leq 79]$

Step 6: **Process the blocks**:

$for\ l\ in\ range(0, len(Chunks))$:
$n \rightarrow 32$

$ch \rightarrow Chunks[l]$
$chu \rightarrow [ch[i: i+n] for\ i\ in\ range(o, len(ch), n)]$
$for\ t\ in\ range(16,80)$: //prepare the message schedule
$M_{[t]} = ROTL^{1}(M_{[t-3]}\ XOR\ M_{[t-8]}\ XOR\ M_{[t-14]}\ XOR\ M_{[t-16]})$
$for\ t\ in\ range(0,80)$:
If (t<=19):
F = (B AND C) OR (NOT B AND D)
If (t>19 and t<=39):
F = B XOR C XOR D
If (t>39 and t<=59):
F = (B AND C) OR (B AND D) OR (C AND D)
If (t>59 and t<=79):
F = B XOR C XOR D
First addition:
Add1 = (F + cv4) mod 2^{32}
Shft5 = $ROTL^{5}$ (cv0)
Second addition:
$Add2 = (Add1 + Shift5) mod 2^{32}$
$cv1 \rightarrow cv0$
Third addition:
$Add3 = (Add2 + M_{[t]}) mod 2^{32}$
Fourth addition:
$Add4 = (Add3 + K_{[t]}) mod 2^{32}$
$cv0 \rightarrow Add4$
$Shift30 \rightarrow ROTL^{30}(cv1)$
$cv4 \rightarrow cv3$
$cv3 \rightarrow cv2$
$cv2 \rightarrow Shift30$
$cv0 \rightarrow (cv0 + f1) mod 2^{32}$
$cv1 \rightarrow (cv2 + f2) mod 2^{32}$
$cv2 \rightarrow (cv3 + f3) mod 2^{32}$
$cv3 \rightarrow (cv4 + f4) mod 2^{32}$
$cv4 \rightarrow (cv5 + f5) mod 2^{32}$
$f1 \rightarrow cv0$
$f2 \rightarrow cv1$
$f3 \rightarrow cv2$
$f4 \rightarrow cv3$
$f5 \rightarrow cv4$

The SHA-1 algorithm that was discussed earlier is identical to the one that was just described. The only thing that is different is that rather of executing a straight forward classical addition, XOR, AND, NOT, OR, and MOD, quantum addition is employed along with the corresponding operations of XOR, AND, NOT, OR, and MOD. The process of doing quantum addition makes use of the idea of a "Full Adder," in which two circuits were created, one for the purpose of computing CARRY and the other for the purpose of computing SUM.

The QASM simulator that is included in QiskitAer is used in order to simulate the quantum circuit. It is necessary to execute the circuit a great number of times,

Fig. 4 SUM Circuit

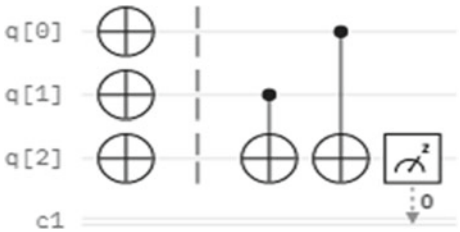


Fig. 5 CARRY circuit

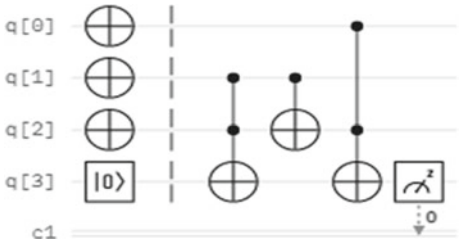
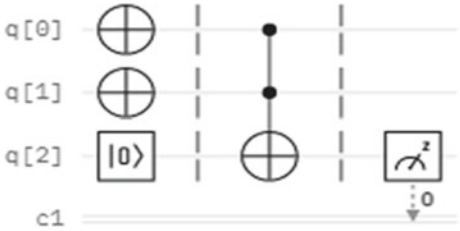


Fig. 6 AND circuit



so that a compilation of statistics regarding the distribution of the bit strings can be made. Through the use of the shots keyword, the execute function allows the user to choose the number of iterations that will be performed on the circuit. A 1024 different pictures were taken for this piece of work.

Quantum addition is the function that can be carried out by the SUM and CARRY circuits, as demonstrated in Figs. 4 and 5. In a similar fashion, the images in Figs. 6 and 7 depict the AND and OR circuits that have been created to carry out the AND and OR operations, and the images in Figs. 8 and 9 depict the NOT and XOR circuits that have been constructed to carry out the particular operations.

6 Experimental Results and Analysis

The execution time of the classic SHA-1 method is significantly faster than the execution time of the Qiskit-implemented algorithm, which is denoted by the notation q-SHA-1 in Table 3. It is probable that the execution of quantum circuits in a classical

Fig. 7 OR circuit

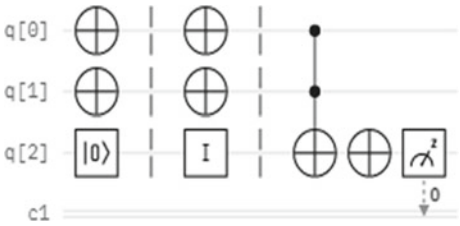


Fig. 8 NOT circuit

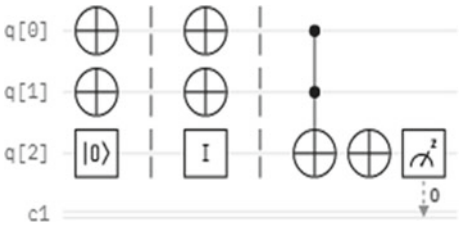
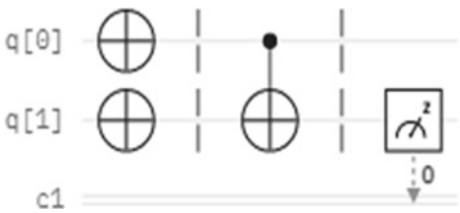


Fig. 9 XOR circuit



computer is one of the reasons why quantum computers have a slower execution time. This hypothesis is based on the observation that classical computers are able to simulate quantum phenomena. There is also the potential that quantum computers address the problems that they encounter bit-by-bit, a process that requires a large amount of time and is another factor that contributes to the slower execution time of quantum computers. The possibility of operating an encryption on a quantum computer rather than a conventional computer, on the other hand, opens the door to future study into the improvement and invention of encryption algorithms that are both superior and more robust.

In Table 4 which can be found below, the execution time of a q-SHA-1 was measured on two separate processors. According to the findings, it appears that the duration of the execution is directly proportional to the type of processor that was employed. This means that a powerful processor should make the execution time shorter. Therefore, if the CPU is more powerful, the traditional computer may not be able to compete with it in terms of performance.

The execution time of the program implemented using Qiskit is slower when compared to that of the classical SHA-1 execution. This could be because all the operations in quantum computing are done bit-by-bit, which consumes a large amount of time, and we have executed the quantum circuits on a classical computer. There need

Table 3 Comparison of SHA-1 and quantum SHA-1 algorithms' performance

Sl. No.	Input	Execution time of quantum SHA-1 in second			Execution time of classical SHA-1 (in second)	Output hash
		1st Run	2nd Run	3rd Run		
1	112345	35.88	34.61	35.23	0.0029981	46A5FE86E6BC827B5C873E3E6B66589E0F471832
2	acjawdjhb	36.06	36.38	36.48	0.0039973	0B6119628C8F48A76D89B9486A0D990930D5BE0B
3	ABDCjbawjh boiuw123	34.86	34.25	34.14	0.0039803	568E03CF4BFD8A4FA8D97BCCE03E4B10CC9AD083
4	TFVGAWDbndawbn!@@#@#BHbhha	36.84	34.78	34.94	0.0040080	2FAA2AD80763B5B6AA717FB5AFAE9B91F96EE2D1
5	!@@#AQHFBAAFBHbfchavhafbhabv !@#!hybahklbf()YHWFVGAUFV !@#JNBAIKFBHJFNJAjfbahfvbahgjvb@#	68.05	67.32	67.20	0.0059804	04B8DD04BE273DF3159F3024F1272840219F919D

Table 4 Comparing the execution time of Quantum SHA-1 in two different processors

Sl. No.	Message to be hashed	Execution time in seconds of two different processors	
		Intel®Core™ i3-5005U CPU @ 2.00GHz	Intel®Core™ i5-8300H CPU @ 2.30GHz
1	12dltvqp	67.27	35.43
2	@12#Abc12.ndei*&%	67.71	35.14
3	%$#$^$%*&%(^*^*&%$@&^$% *^*&#*(&)(+)	67.75	35.99
4	ABNYGFTRVVHGFGFGFGCGHD FDDDDGFDCCFDD	67.05	35.47
5	1248948974654654687878978746878 7987	67.45	35.90
6	Abbdcgufrhurfghvdnbvcfdsndcgsdgc hgs	68.01	35.12
7	Abbdcgufrhurfghvdnbvcfdsndcgsdgc hgsdcnvdbcmbdgbwdbnbvdv	67.51	36.41
8	1556787128798600364712149535487 878794554547845457874211647244	134.34	75.11
9	^^%$#^%(+_+)(&*(^$%@!$#!$%@ $^*&*)+*)(&)*^*^//–/*(^%$$%^$# $%@####$%#$^&%%%&*^^^	134.58	75.98
10	HGFDDRTHHJAQERYTUIJOPJOI YIUTJYGHVBGCFGSTWRDFNV BCGFDRETFHJGDETYRYUGG GJYGJHGHGGJGGUYFGFDRYU TTTIUITGUII	134.22	77.28

to be certain libraries or functions built for basic qubit additions, XOR, AND, NOT, OR, and other operations that are currently available in a classical computer. But the possibility of executing an encryption technique on a quantum computer from a classical computer opens the possibility of future improvements and the development of better and stronger encryption techniques. When compared to classical computers, quantum computers have the potential to achieve quadratic or exponential gains in the speed at which they solve problems. This is made possible by a phenomenon parallelism, which arises as a result of superposition. Quantum computing, on the other hand, does not offer such a boost in speed for each and every issue, and researchers are currently trying to figure out in what circumstances it can be deployed most effectively [15].

Figures 10, 11, 12, 13, and 14 demonstrate, in probabilities, the performance measurement of quantum circuits built to perform various operations. These quantum circuits are implemented in classical and quantum computers using Qiskit. In this instance, the plot histogram function is utilized in order to evaluate the effectiveness of the circuits that are intended to carry out the quantum SHA-1 algorithm. The

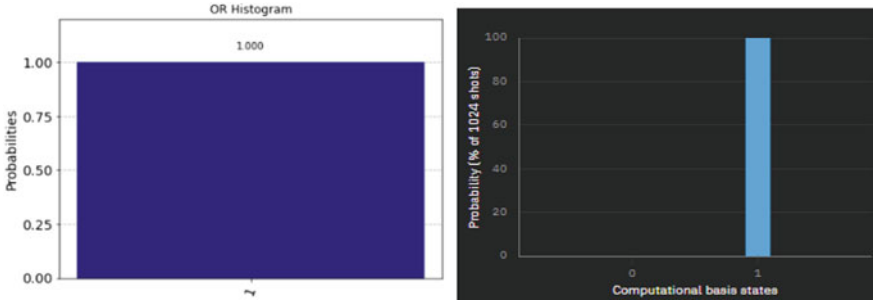


Fig. 10 Performance measurement of OR circuit

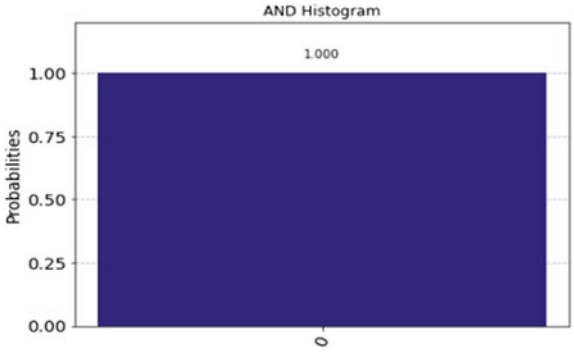


Fig. 11 Performance measurement of AND

measurements of qubits are plotted on a histogram in the order of their frequencies. Because of this, the histogram is a great option for visually representing the findings from the QASM simulator. Histogram visualization can be generated using the plot histogram () function of the qiskit.visualization package. The function accepts data for plotting histograms in the form of a Python dictionary. The various measurement values serve as keys in the dictionary, and the frequencies that correlate to those measurements serve as values.

7 Conclusion and Future Work

In this study, the comparison and contrast of SHA-1 with its quantum counterpart are shown. The IBM QASM simulator is used in order to carry out the implementation of the quantum SHA-1 algorithm on a local computer. For scientific computing, Anaconda is used, while Jupyter is used for interfacing with Qiskit. As a whole, the suggested algorithm takes between 35 and 67 s to complete a single run. The lengthy process is caused by the need to convert between classical and quantum states and back again. In the future, when the quantum computer is fully functional,

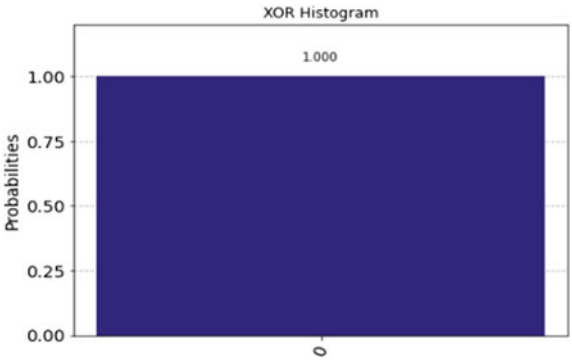


Fig. 12 Performance measurement of XOR

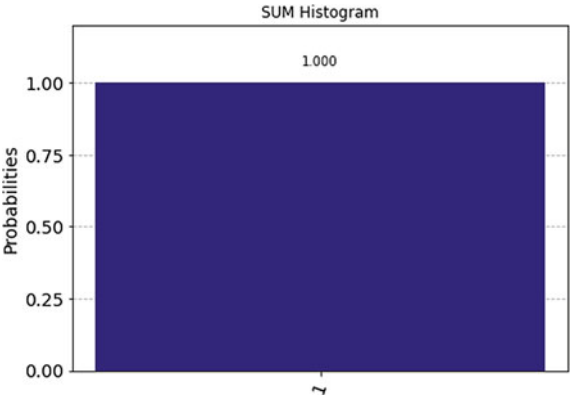


Fig. 13 Performance measurement of SUM

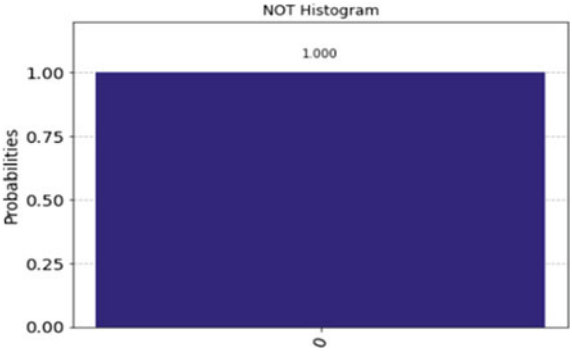


Fig. 14 Performance measurement of NOT

this algorithm will run efficiently. Future research could focus on further optimizing the proposed algorithm to reduce the overall execution time.

References

1. Kasgar AK, Agrawal J, Shahu S (2012) New modified 256-bit MD5 algorithm with SHA compression function. IJCA (0975–8887) 42(12):47–51
2. Schneier B (2005) Schneier on security: cryptanalysis of SHA-1. Schneier Secur
3. NIST.gov—Computer Security Division—Computer Security Resource Center. Archived from the original on 2011-06-25
4. Gaetan L, Thomas P (2020) SHA-1 is a shambles first chosen-prefix collision on SHA-1 and application to the PGP web of trust (PDF). Cryptology ePrint Archive
5. Goodin D (Retrieved 2019-05-29) Microsoft to retire support for SHA-1 certificates in the next 4 months. Ars Technica
6. Stevens M, Bursztein E, Karpman P, Alberttini A, Markov Y, Katz J et al (2017, February 23) The first collison for full SHA-1. In: Advances in cryptology—CRYPTO 2017, vol 10401, pp 570–596. . https://doi.org/10.1007/978-3-319-63688-7_19
7. Barker E (2020) Recommendation for key management. Comput Secur 56. https://doi.org/10.6028/NIST.SP.800-57pt1r5
8. Fernandez-Carames TM, Fraga-Lamas P (2020) Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. IEEE Access 8:21091–21116. https://doi.org/10.1109/ACCESS.2020.2968985
9. Mailloux LO, Lewis II CD, Riggs C, Grimaila MR (2016) Post-quantum cryptography: what advancements in quantum computing mean for IT professionals. In: IT professional, vol 18, no 5, pp 42–47, Sept–Oct 2016. https://doi.org/10.1109/MITP.2016.77
10. Zeydan E, Turk Y, Aksoy B, Ozturk SB (2022) Recent advances in post-quantum cryptography for networks: a survey. In: Seventh international conference on mobile and secure services (MobiSecServ), pp 1–8. https://doi.org/10.1109/MobiSecServ50855.2022.9727214
11. Borges F, Reis PR, Pereira D (2020) A comparison of security and its performance for key agreements in post-quantum cryptography. IEEE Access 8:142413–142422. https://doi.org/10.1109/ACCESS.2020.3013250
12. Ekert AK (1991) Quantum cryptography based on bell's theorem. Phys Rev Lett 67:661–663
13. Nicolas G, Grégoire R, Wolfgang T, Hugo Z (2002) Quantum cryptography. Rev Mod Phys 74(1):145–195
14. Haffner H, Roos CF, Blatt R (2008) Quantum computing with trapped ions. Phys Rep 469(4):155–203
15. Havenstein C, Thomas D, Chandrasekaran S (2018) Comparisons of performance between quantum and classical machine learning. SMU Data Sci Rev l(4):11

Deep Learning-Based Automatic Speaker Recognition Using Self-Organized Feature Mapping

K. Preethi and C. V. P. R. Prasad

Abstract Automatic speaker recognition (ASR) plays the major role in many applications including forensics, dictionary learning, voice verification, biometric systems, and so on. The performance of these application depends on efficiency of ASR system. However, the conventional ASR systems were developed using standard machine learning algorithms, which resulted in low recognition performance. Therefore, this work is focused on development of deep learning-based ASR system. Initially, voice features are extracted using Mel-frequency cepstral coefficients (MFCC), which analyzed the spectral properties of various voice samples. Then, self-organized feature map (SOFM) is applied to reduce the number of available features, which selects the best features using Euclidian similarity between features. Further, deep learning convolutional neural network (DLCNN) model is used to train the features and forms the feature database. Finally, a test voice sample is applied to the trained DLCNN model, which recognizes the speaker detail. The simulations carried out on Anaconda (TensorFlow) showed that the proposed ASR-Net system resulted in superior recognition performance as compared to conventional systems.

Keywords Automatic speaker recognition · Deep learning convolutional neural network · Mel-frequency cepstral coefficients · Self-organized feature map

1 Introduction

The most expressive biometric trait is speech, which is used in a variety of automated applications in addition to being used as a communication tool. The frequent phenomena connected to speech processing include identity identification, emotion detection, attentiveness recognition, etc. [1]. Speech processing is adopted by many

K. Preethi · C. V. P. R. Prasad (✉)
Department of Computer Science and Engineering, Malla Reddy Engineering College for Women (UGC Autonomous), Hyderabad, Telangana State, India
e-mail: prasadcvpr@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_10

applications to achieve the globalized communication independent of the language restriction. Speech processing is applied as an integrated feature to various online and offline systems to verify the presence of the user. The instant feedback through automatic answering machines also uses the speech processing methods to interact with customers or users. The categorization of all these applications can be summarized as the speech recognition, ASR or speech transition system. Each of these systems collectively comes under the pattern recognition in which speech signal is processed in different aspects to answer related queries. Any of speech processing systems [2] need to acquire and process the speech characteristics using the traditional as well as the scientific approach. The traditional phenomenon suggests dealing the speech as the dictations or word processing. The word separation, pause between words, and pitches of spoken words are the common features used by most of the speech processing system to take the adaptive decisions.

The speech systems are also affected by the clarity of speaker pronunciation, vocabulary used by the speaker, speech-disturbing factors, and the environmental constraints. The acoustic methods influenced by the language [3] model is also used to predict the words and the speech itself. The technical concerns and factors are also applied to recognize the speech more appropriately and accurately. In this chapter, different kind of speech processing systems, various issues, and the standard feature generation methods are discussed. Each contributing factor that can influence the speech system is identified and discussed in this chapter along with its significance. The most critical mismatch regarding the data concern is the difference between the simulated and real data. Let a dataset is generated by the English native speakers, but in real time, the application is used by a non-native speaker. In such case, the speech accent and signal features will completely mismatch [4]. The cross-talk situation, noise vector, breathing sound, and fan noise can affect the quality of speech signal during real time. Such kind of interferences and noise increases the difference between the dataset speech signal and the real time captured speech signal [5]. Higher the mismatch between the ground truth data and noise signal data exists, more error in speech and ASR will occur. The major contributions of this work are as follows:

- To implement MFCC for extracting the spectral properties from various voice samples.
- To implement SOFM for reduce the number of available features, which selects the best features using Euclidian similarity between features.
- To implement DLCNN for training and testing of features, which recognizes the speaker details.

Rest of the article is organized as follows: Sect. 2 deals with literature survey, Sect. 3 deals with the proposed deep-ASR implementation, Sect. 4 deals with analysis of results with performance comparison, and Sect. 5 concludes the article with possible future directions.

2 Literature Survey

In [6], authors generated and trained a wider vocabulary set using DNN acoustic model to improve the adaptation of speech recognition. The unified model was integrated by configuring the DNN under specification of wide vocabulary set to handle the intuitions to gain the architectural improvement. The uncertainty weight scheme using hybrid DNN-HMM method was proposed to improve the speech recognition while modeling the uncertainty variance. In [7], authors implemented the nonlinear activation function with approximation, which was applied to generate the loglikelihood scores. The combined model reduced the computational cost and propagates the effective features for maximizing the accuracy of speech recognition [8]. The feature gap evaluation was applied to generate the uncertainty weight under multi-noise conditions. A Bayesian modeling framework with word segmentation in acoustic speech modeling was introduced to map the features to the sequence frames. In [9], authors used the larger vocabulary dataset driven from multi-speakers to improve the classification result. The unsupervised classifier was applied with ground truth measures to reduce the error rate. The speaker-dependent classifier was applied to improve the quality of cluster and word segmentation. The overall system outperforms in multi-speaker and genre-specific speech recognition. An improved HMM adaptation model was defined for adaptive speech recognition from reduced frame rate [10]. The connecting method was defined to achieve the state transition to avoid the probabilistic error in speech mapping. The combined model was defined to rectify the problem and to improve the effectiveness of the speech recognition system. The insertion-prone alignment method was introduced by the author to improve accuracy of the speech recognition system.

In [11], authors had proposed the Highway Deep Neural Network (HDNN) for training the existing acoustic model to facilitate it through gate functions. The depth-gated feed forward method flowed the information [12] between multiple layers with adaption of model parameters. The model was integrated with cross-entropy training to reduce the error and to generate more probabilistic results. The feature transformation and behavior adaptation were also included in information flow generation to improve the accuracy of a speech recognition system. In [12], authors implemented the speech accuracy estimation by estimating the error class. The substitution error, innervations error, and deflection errors were observed by the author to perform probabilistic evaluation on inter-word detection [13]. The refined and accuracy measure on the conditional random field was implied to recognize the relevancy of recognized words. The correlation coefficient-based root mean square error was analyzed to improve the efficiency and reliability of speech recognition.

A very deep convolution neural network (CNN) architecture was configured with different filters, pooling and the input feature set to reduce the data dimension and to improve the accuracy of speech recognition [14]. The architecture was adaptive with pooling, padding, and feature selection methods to improve robustness of ASR. The convergence speed and noise robustness were achieved by this proposed architecture. In [15], authors implemented the graph-based semi-supervised learning method for

speech recognition. The method used the weight constraints and labels for data point specification. The graph specific evaluation and its recording were integrated in two subsequent frameworks to extract most relevant features from neighborhood analysis [16]. The continuous feature-based frame-adaptive method improves the accuracy of phonetic classification. A filter bank learning-based unsupervised auditory was integrated with convolution restricted Boltzmann machine (RBM) method to improve the classification results [17]. The contrastive diverse (CD) was defined to improve the learning method and to improve the feature generation. The sub-band filter-based nonlinear filter was applied to reconstruct the speech and to reduce the residual error for effective speech recognition. A novel joint training framework [18] for speech separation and recognition was defined for corrupted speech. The deep neural network-based acoustic model was defined for weight assignment. The sequence training-based generated features were processed on the back-propagated network to improve the robustness and accuracy in noise integrated speech dataset.

In [19], authors proposed a factor-aware framework for adaptation to neural network for improving the speech recognition. The computational deep model was defined with functional module to improve the classification accuracy. The factor-aware training method [20] was defined to process the distant microphone-based system to generate the multi-factor-aware training of speech. The reduced feature processor improved the performance and complexity with low error rate in classification of speech. In [21], authors proposed a word recognition method based on language adaptation. The structures of the word with frequency and contextual measures were adapted to achieve the existence of speech aspects. The fine-grain context sharing with structural flexibility was analyzed to improve the reliability and accuracy of speech recognition. In [22], authors proposed the multitask learning (MTL) model for improving the speech recognition with low resource adaption. The additional language resource was utilized by the author for improving the performance of a speech recognition system. The heuristic model was defined to generate the effective speech features for multiple languages to improve the mapping results.

In [23], authors introduced the neural network architecture to perform multi-channel filtering and to use oracle knowledge for speech recognition. The computation on frequency decomposition and adaptive variant was defined as the spatial filter to improve the classification rate [24, 25]. The speech processing in real time suffers from various impurities, noise, and interferences. The noise estimation adaptive methods were applied at preprocessing, feature processing, and classification stage improving the speech recognition. A hybrid model using hidden-Markov model (HMM)-based Bayesian feature enhancement (BFE) was proposed to achieve effective noise robust speech recognition [72]. The designed system was capable to handle the speech acquired from multi-microphones and in different noisy environments.

3 Proposed Methodology

Figure 1 shows the proposed deep learning-based ASR system. Initially, voice features are extracted using MFCC, which analyzed the spectral properties of various voice samples. Then, MFCC is applied to reduce the number of available features, which selects the best features using Euclidian similarity between features. Here, the features in each voice sample from a cluster that corresponds to the map neuron, especially the first feature from that cluster; blank gaps indicate the absence of a neuron in that place. This is predicted because sometimes, individual features vary enough from one another for the MFCC to assign them to distinct neurons. This may be decreased by either lowering the number of neurons or by determining the exact distance between individual values (which is partly obfuscated here since the map does not show the actual distances in order to display the voices properly). The U-matrix is a matrix representing the Euclidean distance between the weights of two nearby neurons. This enables to see the depth of the boundaries between various clusters in addition to their location and extracts the features. The dimensionality reduces with the use of SOFM, which is one of the machine learning techniques. For determining a projection of source data into the same count or lesser dimensions, simple operations of matrices from the linear algebra and statistics have used. SOFM can represent the projection method in which the features or m-columns have projected into the subspace by m or lesser columns in addition to the preserving of source data. J is resulted for a source feature matrix I with a size of $n*m$, and J refers to the projection of I. In the initial step, the mean value computes for each column and these values are centered through the subtraction of mean value of column.

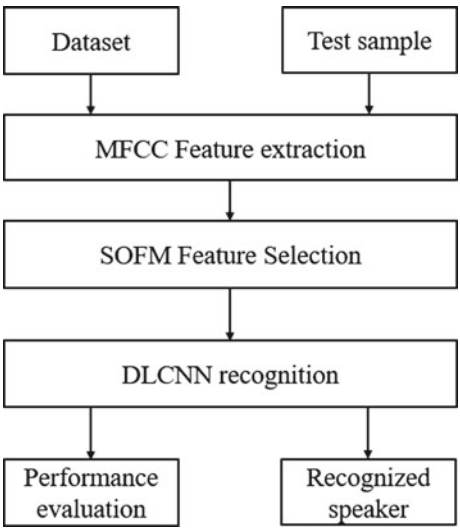


Fig. 1 Proposed ASR-net system

Then, the centered matrix covariance has determined. Finally, the eigenvalue decomposition computes for every covariance matrix which provides the list of eigenvectors or eigenvalues. For the reduced subspace of J, the eigenvectors contain the components or direction and their peak amplitudes represent using the eigenvectors. By making the descending order of eigenvalues, these vectors can be sorted out for ranking of elements of the new subspace for I. k eigenvectors have chosen in general that refer to the principal components or features. Further, DLCNN model is used to train the features and forms the feature database.

3.1 MFCC Feature Extraction

The performance of any classification process is strongly impacted by the features that are acquired from these procedures since feature extraction is regarded as a crucial phase in any sort of deep learning. Furthermore, the main step of any intelligent system is feature extraction, which eliminates redundant data and leaves just the inherent value of the real original material. The important information is therefore confirmed by feature extraction. A voice signal's qualities in the frequency domain are described by its spectral feature set. To get this feature set, the DFT is applied to the signal. This is so that a voice signal may be represented in high-dimensional space with fine spectral characteristics using DFT of the signal. The produced DFT spectrum is then compressed into a smaller feature set that represents the voice signal and is used to tasks involving speech. These representations, which are employed in the synthesis recognition process, may take the shape of MFCCs, LPCs, etc. It helps provide the prosodic characteristics more information, which is extremely helpful.

The earliest step of extraction is pre-emphasis. It is the process of increasing high-frequency energy. The reason behind this is because lower frequencies in the vocal spectrum have more energy than higher frequencies. The nature of the glottal pulse is what causes this phenomenon, known as spectral tilt. Increasing high-frequency energy provides the acoustic model with additional information, which enhances phone recognition performance. Figure 2 shows the proposed MFCC block diagram for feature extraction.

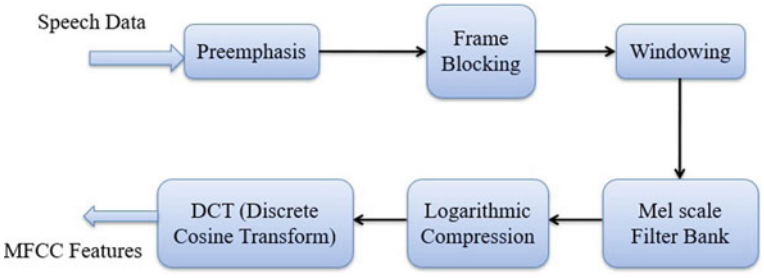


Fig. 2 MFCC operation diagram

Following steps shows the MFCC feature extraction process.

Step 1: Each frame of the supplied voice signal lasts for around 20 ms. The interval between each frame is normally 5–10 ms long.
Step 2: To ensure the signal's continuity, the aforementioned frames are multiplied using a Hamming window. The use of Hamming windows prevents the Gibbs phenomenon. To ensure continuity at the start and stop of each frame and to prevent rash changes at the end point, the Hamming window is multiplied to each frame of the signal. Additionally, each frame is given a Hamming window to group the nearest frequency components together.
Step 3: Applying a Mel-scale filter bank to a DFT power spectrum produces the Mel spectrum. To get data values, Mel-filter focusses more on the important portion of the spectrum. A collection of triangular band pass filters called the Mel-filter bank resembles the human-hearing system. There are overlapping filters in the filter bank. Each output of a filter represents the total energy of a set of frequency bands. This method simulates the human ear's higher sensitivity to lower frequencies. Another crucial aspect to be gathered is the energy contained inside the frame. Calculate the logarithm of the square of the Mel-filter bank's output. The way people react to signals is logarithmically. When energy levels are high, humans are less sensitive to modest energy shifts than when energy levels are low. The dynamic range of values is compressed by the logarithm.
Step 4: Smoothing and Mel-scaling (pull to right). Below 1 kHz, the Mel scale is roughly linear, while beyond 1 kHz, it is logarithmic.
Step 5: Calculate the logarithm of the square magnitude of Mel-filter bank's output.
Step 6: DCT is a further step in MFCC that transforms frequency domain signals into time domain signals and reduces data redundancy, which may ignore the signal's minor temporal fluctuations. Mel-cepstrum is produced by doing DCT on the Mel-logarithm. spectrums. The number of feature dimensions is decreased by using DCT. It lessens the spectral correlation between the coefficients of the filter banks. For every statistical classifier, low dimensionality and 17 uncorrelated features are ideal. The energy is not captured by the cepstral coefficients. Adding an energy feature is thus required. As a result, one energy coefficient and twelve Mel frequency cepstral coefficients are retrieved. The term "basic features" often refers to these thirteen characteristics.
Step 7: Acquire MFCC capabilities. Using the equation, the MFCC is created by transforming the frequency into the cepstral coefficients and the cepstral coefficients back into the frequency.

$$mel(f) = 2595 \times \log 10\left(1 + \frac{f}{700}\right) \tag{1}$$

where f stands for frequency in hertz (Hz). The procedure used to calculate the MFCC. The following equation is used to estimate the MFCC characteristics.

$$C_n = \sum_{n=1}^{K} (\log S_k) \left| n(K - \frac{1}{2}) \right|_K \quad \text{where, } n = 1, 2, \ldots K \tag{2}$$

Since *Cn* indicates the mean value of the input speech signal, which contains no important speech-related information, it is not included in the DCT in this instance. Here, *K* stands for the number of Mel cepstral coefficients. An acoustic vector made up of MFCC is generated for each of the overlapping frames of speech (about 20 ms). The qualities of speech are both represented by and recognized by this collection of coefficients.

3.2 Feature Selection with SOFM

In speech technology, SOFM is a highly potent and well-liked approach for feature extraction. The fundamental strategy in this case is to forecast the current speech sample based on the linear combination of the previous speech samples. It is widely used in the field of voice research due to its ease of implementation, speedy calculation, and accuracy in mathematics. It is used to choose the vowels, consonants, syllables, phonemes, isolated words, etc., in speech. In this digital approach, an analog signal is encoded, and the specific signal value is predicted from the previous value with the use of a linear function. The fundamental reason the SOFM model is chosen is because a speech sample, $S(n)$, may be created at any moment by linearly mixing the previous speech samples, i.e.,

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \cdots + a_n s(n-p) \tag{3}$$

where it is anticipated that the coefficients are $a1$, $a2$, $a3$, …, anwill not change during the course of the speech analysis frame. The aforementioned equation may thus be changed into an equality by including the excitation term Gu(n) as shown below:

$$s(n) = \mathrm{Gu}(n) + \sum_{i=1} a_i s(n-1) \tag{4}$$

The relationship shown below may be obtained by translating Eq. (4) into the Z domain:

$$S(n) = Gu + \sum_{i=1}^{p} a_i \cdot z^{-1} s(z) \tag{5}$$

As demonstrated below, the relation may be changed into a transfer function.

$$H(z) = \frac{s(z)}{\mathrm{Gu}(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-1}} = \frac{1}{A(z)} \tag{6}$$

In this study, SOFM-based feature vectors as well as other crucial metrics with phonetic data are employed. In order to extract a feature vector with cepstral weights, each frame of a continuous speech stream is processed. Then, sampling and quantization are performed using an analog-to-digital converter. Pre-emphasizing the voice signal results in spectrally flattening of the signal. A digital filter of the first order is used for this. The digital filter's transfer function is provided by:

$$H(z) = 1 - az^{-1}, \quad \text{for } 0 \leq a \leq 1.0 \tag{7}$$

Here, a signal with continuous speech is regarded as one frame. The Hamming window is a feature of Windows that is used to get around Gibbs phenomenon's shortcomings. This function multiplies the frames. It is calculated as:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi m}{N-1}\right), \quad for\, 0 \leq n \leq N \tag{8}$$

Here, N stands for the total number of samples in the block. The signal's frames are now all automatically linked to produce:

$$r_f(m) = \sum_{n=0}^{N-m} \tilde{x}_f(n)\tilde{x}(n+m), \quad m = 0, 1, 2, \ldots, p \tag{9}$$

The optimal auto-correlated value in this case is shown by $r_f(m)$.

3.3 DLCNN Classification

Figure 3 shows how the DLCNN architecture is used to identify the speaker. Convolution and max-pooling were used three times each in this method; the results of the convolution and max-pooling layers were fused into a single feature layer. Final weight maps for the speech features may then be generated by using the SoftMax output layer and the focus map generation layer together. Each layer action of the proposed DLCNN architecture is described in the following manner:

Convolution layer: This type of layer is often utilized to estimate the yield of nerve cells in the input feature linked to a local region. The coefficients of weight are created using 64 filters in this case. Individual neuron conducts the operation of dot product betwixt the linked regions, pixels, or linked regions. The final dimension of layer CONV 1 is $\{20 \times 8 \times 64\}$ where height and width of input is 256 and the filter proportion is 64 with kernels of 3×3 size.

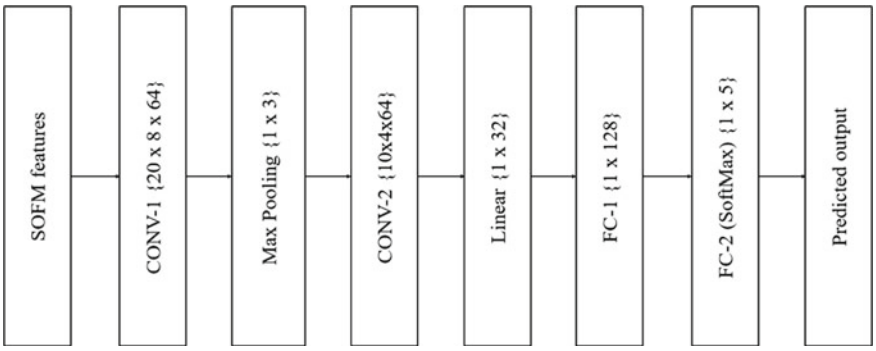


Fig. 3 Proposed DLCNN model

Table 1 Layer sizes of DLCNN model

Layer	Height	Width	Features	Stride height	Stride width	Parameters (k)
Conv1	20	8	64	1	3	10.2
Conv2	10	4	64	1	1	164.8
Lin	–	–	32	–	–	65.5
FC1	–	–	128	–	–	4.1
SoftMax	–	–	5	–	–	0.7

Max-pooling Layer: This layer involves the operation of down-sampling operation subjected on input feature. Hence, the sampling by factor 2 is implied on both width and height of the feature, which yields a layer sized $\{1 \times 3 \times 128\}$, where width and height of feature are 128 and filter size 128, respectively.

Linear layer: This layer is used to concatenate the input features. Therefore, the layer size remains similar $\{1 \times 32 \times 2\}$ where 2 is representation of filter size, whereas height and width of feature is represented by 32.

SoftMax output layer: The SoftMax output layer probability is employed, which reduces inefficiencies during the speaker recognition operation development. The error falls betwixt 1 and 0, as this function reduces speaker recognition inefficiencies. Finally, the SoftMax classifier results the recognized person. Table 1 presents the layers of SoftMax output layer.

4 Results and Discussion

This section gives the detailed analysis of simulation results. Further, the performance of proposed ASR system is compared with the state-of-the-art approaches using same dataset.

4.1 Dataset

The dataset contains the voice samples from five different persons. From each person, nine samples are collected with multiple phrases. Totally, 45 phrases are collected in entire dataset. Further, 80% of dataset is used for training, 20% of dataset is used for testing the ASR system.

4.2 Subjective Performance

Figure 4 shows the original audio with 2.5 s of time. Figure 5 shows the frequency of original audio, which is having the frequency of −3 k to + 3 k. Figure 6 shows the spectrogram of MFCC. Figure 6 shows the spectral features of MFCC, which contain eighth-order feature, cepstral feature, and log-MFCC features. Figure 7 shows the recognized person details.

Table 2 presents the performance of proposed ASR system with conventional approaches such as HDNN [62], CNN [64], and DBN [107]. The proposed DLCNN-based ASR resulted in superior performance for all metrics.

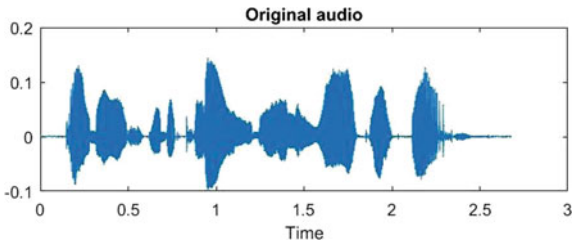


Fig. 4 Original audio

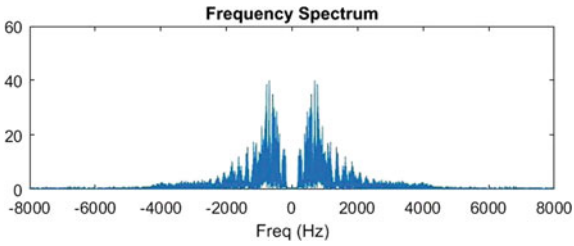


Fig. 5 Frequency spectrum

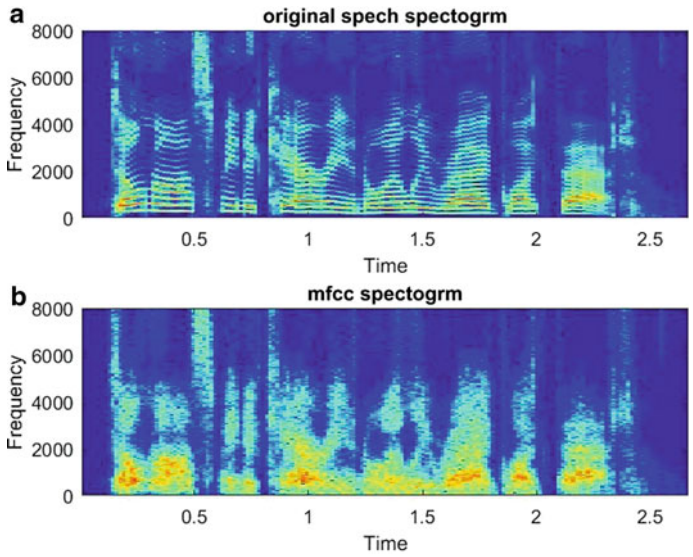


Fig. 6 Spectrogram **a** original speech, **b** MFCC

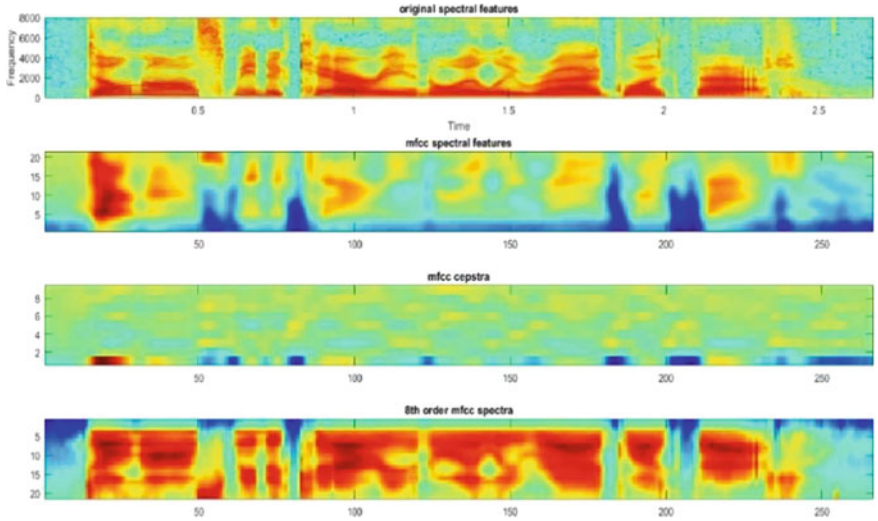


Fig. 7 Spectral features of MFCC

Table 2 Performance comparison

Metric	HDNN	CNN	DBN	Proposed ASR
Accuracy (%)	92.550	94.750	94.200	99.98
Sensitivity (%)	90.307	91.267	95.715	99.973
Specificity (%)	94.770	94.104	95.669	99.83
F-measure (%)	93.466	92.825	95.257	99.67
Precision (%)	93.262	94.428	90.688	99.29
MCC (%)	92.947	95.241	90.583	99.45
Dice (%)	90.762	90.624	92.727	99.37
Jaccard (%)	93.565	94.364	91.614	99.46

5 Conclusion

This work is mainly focused on implementation of DLCNN-based ASR system. Here, MFCC was initially used to extract voice features, which examined the spectral characteristics of various voice samples. Then, SOFM is applied to decrease the number of available features and choose the best features, using Euclidean similarity between features. The feature database is created using the DLCNN model, which is also used to train the features. The trained DLCNN model then processes a test voice sample to identify the speaker detail. The simulations run on Anaconda (TensorFlow) demonstrated that the proposed ASR system produced better recognition results than traditional systems. This work can be extended to implement with transfer learning-based ASR approach. Here, advanced filters are suggested for preprocessing, which eliminates any unwanted noise elements or artifacts in speech signals.

References

1. Ali A, Gravino C (2019) A systematic literature review of software effort prediction using machine learning methods. J Softw Evol Process 31(10):e2211
2. De Lima TA, Speech C (2019) A survey on automatic speech recognition systems for Portuguese language and its variations. Comput Speech Lang 62:101055
3. Claus F, Rosales HG, Petrick R, Hain HU (2015) A survey about databases of children's speech a survey about databases of children's speech Dresden University of Technology, Chair for System Theory and Speech Technology. INTERSPEECH. 2015, pp. 2410–2414. Available online: https://www.isca-speech.org/archive_v0/archive_papers/interspeech_2013/i13_2410.pdf. Accessed on 15 Mar 2021.
4. HTK Speech Recognition Toolkit. Available online: http://htk.eng.cam.ac.uk/. Accessed on 2 Sept 2020
5. Overview of the CMUSphinx Toolkit. Available online: https://cmusphinx.github.io/wiki/tutorialoverview/. Accessed on 2 Sept 2020
6. Povey D, Ghoshal A, Boulianne G (2011) The Kaldi speech recognition toolkit. IEEE Signal Process Soc, pp 1–4. Available online: http://kaldi.sf.net/. Accessed on 19 July 2020

7. Open-Source Large Vocabulary CSR Engine Julius. Available online: http://julius.osdn.jp/en_index.php. Accessed on 2 Sept 2020
8. Sunil Y, Prasanna SRM, Sinha R (2016) Children's speech recognition under mismatched condition: a review. IETE J Educ 57:96–108
9. Bhardwaj V, Kadyan V (2020) Deep neural network trained punjabi children speech recognition system using Kaldi toolkit. In :Proceedings of the 2020 IEEE 5th international conference on computing communication and automation (ICCCA), Greater Noida, India, 30–31 Oct 2020, pp 374–378
10. Claus F, Rosales HG, Petrick R, Hain H (2013) A survey about ASR for children. ISCA Arch, pp 26–30. Availableonline: https://www.isca-speech.org/archive_v0/slate_2013/papers/sl13_026.pdf. Accessed on 5 July 2021
11. Kathania HK, Kadiri SR, Alku P, Kurimo M (2021) Spectral modification for recognition of children's speech under mismatched conditions. In: Proceedings of the 23rd Nordic conference on computational linguistics (NoDaLiDa); Linköping University Electronic Press, Linköping, Sweden, pp 94–100. Available online: https://aclanthology.org/2021.nodalida-main.10. Accessed on 5 Sept 2021
12. Madhavi MC, Patil HA (2019) Vocal tract length normalization using a Gaussian mixture model framework for query-by-example spoken term detection. Comput Speech Lang 58:175–202
13. Kathania HK, Kadiri SR, Alku P, Kurimo M (2021) A formant modification method for improved ASR of children's speech. Speech Commun 136:98–106
14. Tsao Y, Lai YH (2016) Generalized maximum a posteriori spectral amplitude estimation for speech enhancement. Speech Commun 76:112–126
15. Bhardwaj V, Kukreja V (2021) Effect of pitch enhancement in Punjabi children's speech recognition system under disparate acoustic conditions. Appl Acoust 177:107918
16. Bhardwaj V, Kukreja V, Singh A (2021) Usage of prosody modification and acoustic adaptation for robust automatic speech recognition (ASR) system. Rev d'Intell Artif 35:235–242
17. Takaki S, Kim S, Yamagishi J (2016) Speaker adaptation of various components in deep neural network based speech synthesis. In: Speech synthesis workshop, pp 153–159. Available online: https://206.189.82.22/archive_v0/SSW_2016/pdfs/ssw9_PS2-5_Takaki.pdf. Accessed on 15 Apr 2021
18. Kathania HK, Kadiri SR, Alku P, Kurimo M (2021) Using data augmentation and time-scale modification to improve asr of children's speech in noisy environments. Appl Sci 11:8420
19. Kaur H, Bhardwaj V, Kadyan V (2021) Punjabi children speech recognition system under mismatch conditions using discriminative techniques. In: Innovations in computer science and engineering. Springer, Singapore, pp 195–202
20. Klejch O, Fainberg J, Bell P, Renals S (2019) Speaker adaptive training using model agnostic meta-learning. In: Proceedings of the 2019 IEEE automatic speech recognition and understanding workshop (ASRU), Sentosa, Singapore, 14–18 Dec 2019, pp 881–888
21. Bhardwaj V, Bala S, Kadyan V, Kukreja V (2020) Development of robust automatic speech recognition system for children's using Kaldi toolkit. In: Proceedings of the second international conference on inventive research in computing applications (ICIRCA-2020), Coimbatore, India, 15–17 July 2020, pp 10–13
22. Bala S, Kadyan V, Bhardwaj V (2021) Bottleneck feature extraction in punjabi adult speech recognition system. In: Innovations in computer science and engineering. Springer, Singapore, pp 493–501
23. Shivakumar PG, Georgiou P (2020) Transfer learning from adult to children for speech recognition: evaluation, analysis and recommendations. Comput Speech Lang 63:101077
24. Shahnawazuddin S, Bandarupalli TS, Chakravarthy R (2020) Improving automatic speech recognition by classifying adult and child speakers into separate groups using speech rate rhythmicity parameter. In: Proceedings of the international conference on signal processing and communications (SPCOM), Bangalore, India, 28 Aug 2020, pp 1–5
25. Kathania HK, Kadiri SR, Alku P, Kurimo M (2020) Study of formant modification for children ASR. In: Proceedings of the international conference on acoustics, speech and signal processing (ICASSP), Virtual Barcelona, 4–8 May 2020, pp 7424–7428

Machine Learning-Based Path Loss Estimation Model for a 2.4 GHz ZigBee Network

Prashanth Ragam, Guntha Karthik, B. N. Jagadesh, and Sankati Jyothi

Abstract Wireless sensor networks (WSNs) and Internet of Things (IoT) have received remarkable attention from the past few years in various applications. Modeling of path loss (PL) for the deployment of a developed WSN system is a crucial task owing to the time-consuming and elegant operation. However, radiofrequency (RF) engineers adopted either deterministic or stochastic empirical models to estimate the PL. In general, empirical models utilize predefined influenced parameters including path loss (dB), path loss exponent (Υ), and other significant parameters. Although, empirical models differ significantly from original measurement due to consideration of different terrains. In this study, an endeavor has been made to develop a machine learning-based model to estimate the path loss for a standard ZigBee communication network operating on a 2.4 GHz carrier frequency deployed in an urban area. An experimental setup was designed and tested in line-of-sight (LOS) and non-line-of-sight (NLOS) conditions to collect the influence parameters such as received signal strength indicator (RSSI), frequency, distance, and transmitter antenna gain. Besides that, environmental parameters such as temperature and humidity are also included. In this context, a three-layer, feed-forward backpropagation multilayer perception neural network (BPNN) machine learning and Log-Distance empirical models were employed to estimate the PL. The obtained results reveal that the BPNN model noticeably enhanced the coefficient of determination (R^2) and reduced root mean square error (RMSE) compared with the empirical

P. Ragam (✉) · B. N. Jagadesh
School of Computer Science and Engineering, VIT-AP University, Inavolu Beside AP Secretariat, Amaravathi, Andharpradesh, India
e-mail: prashanth.rajam@vitap.ac.in

G. Karthik
Department of ECE, Stanely College of Engineering and Technology for Women, Hyderabad, Telangana, India
e-mail: gkarthik@stanley.edu.in

S. Jyothi
Software Development Cell, VIT-AP University, Inavolu Beside AP Secretariat, Amaravathi, Andharpradesh, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_11

model. The R^2 and RMSE metrics were obtained as 0.97220 and 0.03630 in the NLOS scenario as well as 0.99820 and 0.00773 in the LOS environment, respectively.

Keywords ZigBee · Path loss · BPNN · Log-distance · Average RSSI

1 Introduction

The technologies developed based on IoT and WSN felicitous are a viable choice for distributed monitoring using smart sensors that can function in specified locations and in a variety of geographical and physical situations. The main motto of WSN and IoT is to convey real-time data including air pollution, temperature, unwanted gases, atmospheric humidity, water quality, and pH values through sensor units. Most of the cases require a distinct radio frequency (RF) module such as Bluetooth, ZigBee, LoRa, and NB-IoT to convey the information accumulated by the sensor nodes. Consequently, RF modules play an important role in WSN and IoT-based applications. During wireless propagation, path loss represents the difference between transmitted and received power that can be widely adopted to estimate signal attenuation. Currently, various researchers and scientists are working to enhance wireless network coverage range by improving path loss models depending on electromagnetic propagation [1]. Usually, path loss (PL) has several models specific to particular scenarios. These models consist of three types such as empirical, semi-deterministic, and deterministic models [2]. Among, empirical also known as conventional models are widely adopted because of simple implementation and legible mathematical expression over other PL models. However, empirical PL models including Log-Distance (LD), Cost-231 Hata, Ericcson, and SUI models are restricted by environmental locations like semi-urban, rural, and urban. Hence, they may not suitable for all-terrain. Moreover, these models were considered few affected parameters only while predicting the PL. But, in ground reality, path loss is influenced by environmental parameters too. So that, it is essential to introduce advanced modeling methods to estimate the path loss accurately in a complex environment scenario.

In most IoT applications such as agriculture, smart cities, opencast mines, when the wireless signal is propagated from the sensor node to the receiver node, it crosses via various obstacles (buildings, plants, dense crops). Therefore, an adequate clearance location would not be ensured, which leads to signal attenuation, scattering, and signal reflection. In this instance, the strength of the communication link will be reduced, especially, during the signal transmitted via dense crops and high-rise buildings. Because of this reason, the quality of communication links and variables should be taken while IoT and WSN systems are deployed. The performance of the propagation signal of WSNs is dependent on the working environment directly. Henceforth, to evaluate the propagation behavior, a precise path loss model is needed in the wireless communication network to reduce minimal pocket loss between transmitter and receiver node as well as enhance the long-lasting lifetime. For this reason, this challenge has inspired us to develop an intelligent machine learning-based path

loss model between the ZigBee router node (varies with distance) and the coordinator node (constant on the ground).

The contribution of this study can be summarized as follows:

- An experimental setup (hardware) was designed with ZigBee technology. It was installed in the urban environment area to measure the RSSI.
- From the collected RSSI values, two-path loss models such as Log-Distance and three-layer multilayer back-propagation neural network models were formulated by considering influenced parameters such as distance, antenna gain, frequency, humidity, temperature, and RSSI.
- The evaluation metrics such as R^2, RMSE, mean absolute difference (MAD), and mean square error (MSE) were obtained from the analysist to ensure the performance of developed models.

2 Previous Works

In general, RSSI coverage, coverage range, and the distance were considered to estimate the required path loss. In 2017, Pan et al. [3] conducted a test using a 2.45 GHz RF module in a cornfield environment and estimated the PL using a log-normal shadowing model (LNSM). The obtained results show that R^2 between measured and predicted PL is 0.937 and 0.9888. Egi and Otero [4] were created a machine learning and 3D-point-cloud-based model to predict the signal power path loss (SPLL) by considering distance as an influencing parameter. The results ensure that the proposed model performed with 2.4% mean absolute percentage error (MAPE). Jawad et al. [2] were formulated hybrid models such as exponential-based particle swarm optimization (EXP-PSO) and polynomial particle swarm optimization (POLY-PSO). The simulation results prove that the proposed model has a received signal level (RPL) of 4.26%, whereas empirical models having errors within the range of 6.29–16.9%. Jo et al. [5] were developed a combined machine learning framework such as artificial neural network (ANN), Gaussian process, and principal component analysis (PCA) for modeling the path loss. They observed that the proposed combined machine learning model provides more accuracy over other conventional linear model. The author [6] tested the ZigBee WSN and IEEE 802.15.4 communication networks at 2.4 GHz carrier frequency at distinct terrains to estimate the PL. The regression results provide a high R^2 (in the range of 0.915–0.983) and the least RMSE of 2.5. From the previous literature review, multiple researchers were developed machine learning models as well as adopted conventional empirical models to estimate the path loss. However, they were considered limited influenced parameters such as distance and carrier frequency only. The path loss is affected by other parameters including environmental conditions too.

3 Evaluating Experiment

Connectivity is a primary building component in WSN and IoT systems. The popular short-range communication protocols in IoT and WSN are ZigBee, Bluetooth, and EnOcean. In this section, an energy-efficient ZigBee module (XBee series 2 (S2C), DIGI) along with accessories was selected as an experimental setup (see Figs. 1 and 2). Free software such as the X-CTU platform (DIGI) was used to configure the ZigBee modules. The coordinator (sink or transmitter) node was interfaced to a PC using the RS-232 USB cable; on the other side, the router (receiver) node was connected to a power supply (Intex power pack). Here, the X-CTU software tool is used to configure the modules as well as monitor packet loss and RSSI levels. A coverage range test was conducted in two significant use scenarios, namely LOS (outdoor-to-outdoor) and NLOS (indoor-to-outdoor) in an urban region near Bagath Nagar, Hyderabad metropolitan city, India, to understand the capability of the ZigBee communication network. The deployment region has a geographic coordinate such as latitude of 17°46′46.14″ and longitude of 78°42′83.01″. The concerned area comprises high-rise dwellings, vegetations, and roadside trees on two sides of roads (see Fig. 3). In each scenario, the coordinator node was kept static (point A), while the receiver node was non-stationary. At every deployment case, ten readings were captured, and an average of the RSSI and pocket loss was performed for further analysis. During this measurement campaign test, sixty and hundred RSSI readings were recorded for both NLOS and LOS scenarios, respectively. The specifications of XBee S2C modules are depicted in Table 1. The relationship between distance and average received power at the receiver of different locations is carried out in a specified area as shown in Figs. 4 and 5, respectively.

As stated earlier, the path loss is not only affected by traditional parameters such as distance and operating frequency but also influenced by other parameters including environmental conditions (temperature and humidity). Thus, recorded temperature and humidity values at each case during the measurement campaign are summarized in Table 2.

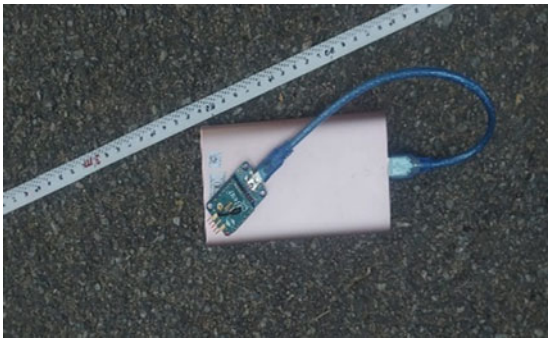
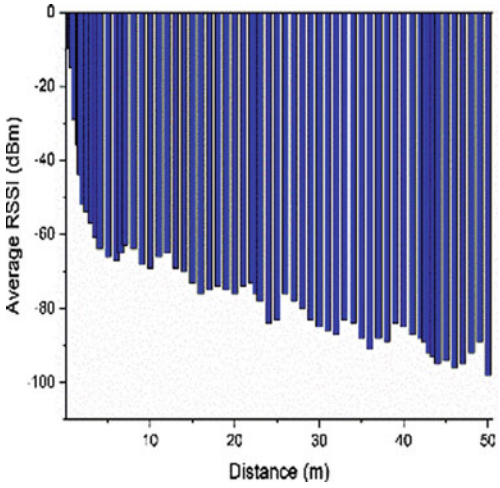


Fig. 1 XBee S2C router node

Fig. 2 XBee S2C coordinator

Fig. 3 Experiment coverage plan

Fig. 4 Average RSSI versus distance in NLOS case



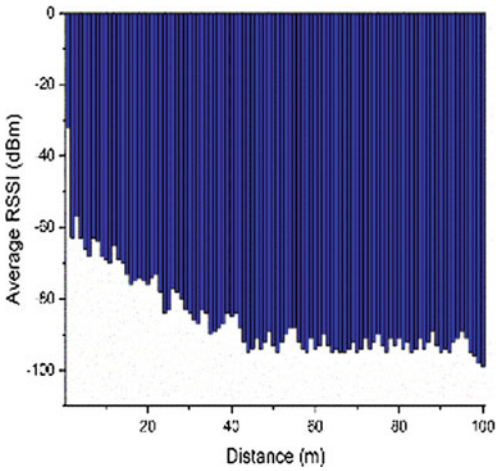


Fig. 5 Average RSSI versus distance in LOS case

Table 1 Hardware specification of the router and coordinator nodes

Hardware module	Specification
Transmitted power	8 dBm
Antenna gain	2.1 dBi
Carrier frequency	ISM 2.4 GHz
Receiver sensitivity	−102 dBm

Table 2 Recorded influenced parameters

Parameter	LOS		NLOS	
	Min.	Max.	Min.	Max.
Distance (m)	1	100	0.5	50
Temperature (°C)	31	37	31	37.5
Humidity (%)	17	28	17	30
Average RSSI (dBm)	−98	−10	−98	−10

Note Min.—minimum value; Max.—maximum value

4 Empirical Path Loss Modeling

In this research work, an experimental model has been deployed as well as the path loss of ZigBee wireless signal operating at 2.4 GHz was determined experimentally at Bagath Nagar, Hyderabad. A notable word in estimating the wireless propagation coverage range is PL describes a signal attenuation in decibels (dB).Usually, the PL at varied locations from the transmitter and receiver nodes in any WSN is expressed as follows [7]:

$$\mathrm{PL}_{\mathrm{measured}} = P_{\mathrm{t}} - P_{\mathrm{r}} + G_{\mathrm{t}} + G_{\mathrm{r}}. \tag{1}$$

Here, P_{r} is the power of the signal at various locations from the transmitter (dBm), P_{t} is the power of the transmitter (dBm), G_{t} is a gain of transmitter antenna (dBi), and G_{r} is gain of receiver antenna (dBi). The defined values for G_{t} and G_{r} are equal to 2.1-dBi. In general, the wireless communication propagation models are categorized into three types such as free space model, two-ray-reflection ground model, as well as Log-Distance model. The Log-Distance propagation model can able to estimate the PL of a communication signal confronts within a building or densely populated areas over distance. Henceforth, this study carried out an experimental test in an urban region that comprises high-rise buildings and trees. Consequently, applied a first-order Log-Distance (LD) propagation model [8] to find the predicted path loss and formally expressed it as:

$$\mathrm{PL}_{\mathrm{predicted}} = \mathrm{PL}_0 + 10\Upsilon \log_{10}\left(\frac{r}{r_0}\right) + X_\sigma. \tag{2}$$

where r_0 is the specified near-earth reference distance. X is a zero-mean Gaussian random variable (dB) and σ is the standard deviation. If the devices are kept static, ignore the effects of X_σ.Υ is a path loss exponent that is based on the surrounding or propagation region. Here, it has been taken as three. PL_0 is the path loss at a near-earth reference distance r_0 and is defined as [9]:

$$\mathrm{PL}_0 = 20\log 10(f_{\mathrm{MHz}} + 20\log 10(r_0)) - 28. \tag{3}$$

To estimate the PL at a reference distance of 1 m (r_0) using Eq. 3 and plugging it into Eq. 2 results in:

$$PL_{\mathrm{predicted}} = 20\log 10(f_{\mathrm{MHz}}) + 10\eta \log 10(r) - 28 + X_\sigma. \tag{4}$$

where, f_{MHz} is the operation frequency of propagation model.

5 Machine Learning Modeling

Machine learning (ML) has emerged with high-speed performance big data computing to provide new opportunities in various areas such as agriculture, mining, and healthcare. Usually, ML methodologies require a learning process using the objective to learn from "experience" to execute a specific function. ML consists of data that includes examples. Generally, an individual example is denoted by a set of attributes, also well-known as parameters, features, or variables. Here, feature may be nominal, numeric, binary, or ordinal. The designed ML algorithm's performance can be evaluated by numerous performance metric which is useful for improving the ML model. The ML tasks can be classified into various categories based on learning

type (supervised/unsupervised), learning models such as regression, clustering, classification to implement the specified task. An artificial neural network (ANN) is a twig of artificial intelligence that can be originated from the neural system of the human brain. It has a powerful computing system comprised of interconnected adaptive simple processing elements (known as neurons). These neurons are associated with the weighted vector to compose the ANN architecture to do the massive parallel computation for data processing [10].

Usually, multi-layer perceptron (MLP) is the wide popular adopted algorithm. It is a class of feed-forward ANNs and consists of at least three layers (nodes) specified as input, hidden, and output layers. Every layer has a weight (w) matrix, a bias (b) vector, and an output vector. Except for input nodes, each layer can possess a various number of neurons that use a nonlinear activation function. Every intermediate layer's response has been given as the input response of the subsequent layer. The fundamental purpose of the training process in a neural network is to suppress the measured error. In this study, considered a back-propagation (BP) training algorithm to achieve accurate estimation. It consists of four stages such as initialization of weights, feed-forward, back-propagation of errors, and updating of weights and biases [11]. A MATLAB 2018b software was utilized to implement the BPNN model. A three-layer BPNN model was designed with fifteen hidden neurons in the middle layer. Here, the input node has six neurons (input parameters), and one output neuron (parameter) in the output layer. A Levenberg–Marquardt back-propagation-based training function was adopted. The database can be divided into 70% for training and 30% for testing in both scenarios (LOS & NLOS). The following algorithm was used to develop the BPNN model and as follows:

Algorithm: Multi-Layer Back-Propagation network (BPNN)
Input: Collected dataset Output: Predicted PL data set
Step 1: Normalize and split the data for training and testing; Select the number of layers (m) and neurons (n_i)
Step 2: Initial values of weight vector $W_{i,j}^{m}$ and bias vector $b_{i,j}^{m}$
Step 3: Determine neural network response using $a_{i,j}^{m} = \varphi\left(\left(w_{i,j}^{1}\right) * X_k + \text{bias}\right)$
Step 4: Calculate neural output: $Y_j = \varphi\left(\left(w_{i,j}^{2}\right) * a_{i,j}^{m} + \text{bias}\right)$
Step 5: Determine the error between actual and predicted output: $e_j(n) = d(n) - Y_j(n)$
Step 6: If obtained error is high, update the weights and bias: $w_{i,j}^{1}(\text{new}) = w_{i,j}^{1}(\text{old}) + \eta s_J^1$; $w_{i,j}^{2}(\text{new}) = w_{i,j}^{2}(\text{old}) + \eta s_J^2\left(a_{i,j}^{m}\right)$ Repeat the aforementioned steps until reach desired error **Note**: i and j are number (1, 2, 3 … n_i); φ is the transfer function; $w_{i,j}^{1}$—weight associated with the neuron; $d(n)$ is actual output; S_j is a sensitivity function

Table 3 List of performance metrics

Name	Equation
MAD	$\frac{\sum_{i=1}^{n} \lvert A_i - P_i \rvert}{n}$
MSE	$\frac{\sum_{i=0}^{n} \lvert A_i - P_i \rvert^2}{n}$
RMSE	$\sqrt{\frac{\sum_{i=0}^{n} \lvert A_i - P_i \rvert^2}{n}}$

Note A_i, P_i are measured and predicted values

6 Model Performance

Different applied ML algorithm performance can be evaluated by various standard statistical performance evaluation metrics. The widely adopted statistical metrics are coefficient of determination (R^2), mean absolute deviation (MAD), mean square error (MSE), and root mean square error (RMSE). Table 3 depicts adopted statistical performance metrics.

7 Outcomes and Discussion

In this section, the PL estimation of ZigBee network can be done using the machine learning model (BPNN) and empirical model. The outcomes of both models are presented. Further, the observed R^2 for Log-Distance model is 0.93401 and 0.90126 in NLOS and LOS cases, respectively (See Figs. 6 and 7). Similarly, obtained R^2 for the BPNN model is 0.97220 (NLOS) and 0.99820 (LOS), respectively, as depicted in Figs. 8 and 9. Tables 4 and 5 summarize the performance of developed models using evaluation metrics such as R^2, MAD, MSE, and RMSE. Here, maximum R^2 and minimum RMSE were obtained for the BPNN model in both cases.

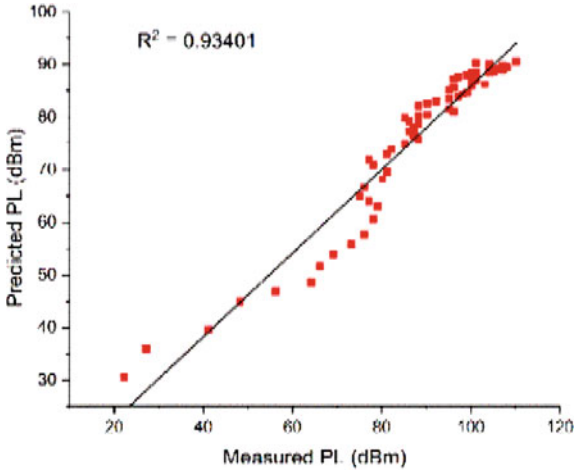


Fig. 6 Measured PL versus predicted PL using LD model in LOS

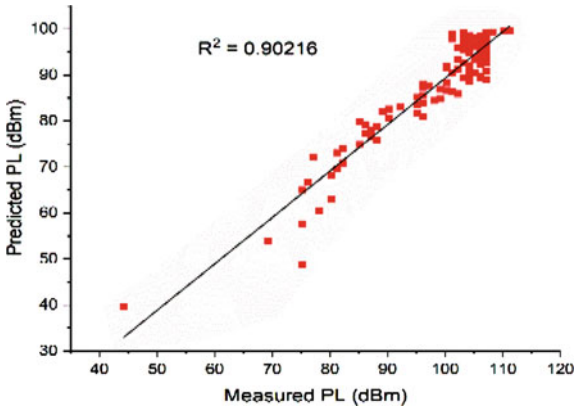


Fig. 7 Measured PL versus predicted PL using LD model in NLOS

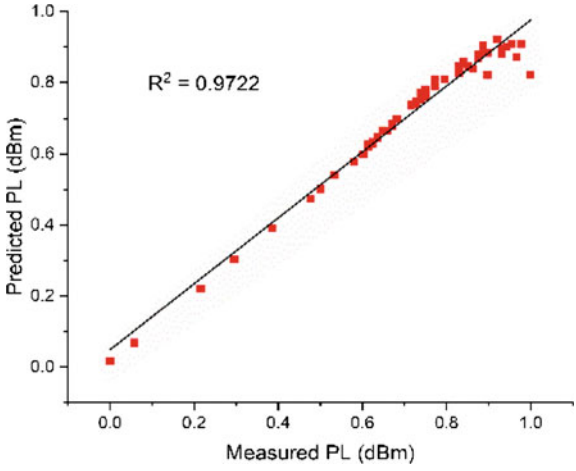


Fig. 8 Measured PL versus predicted PL using BPNN model in NLOS

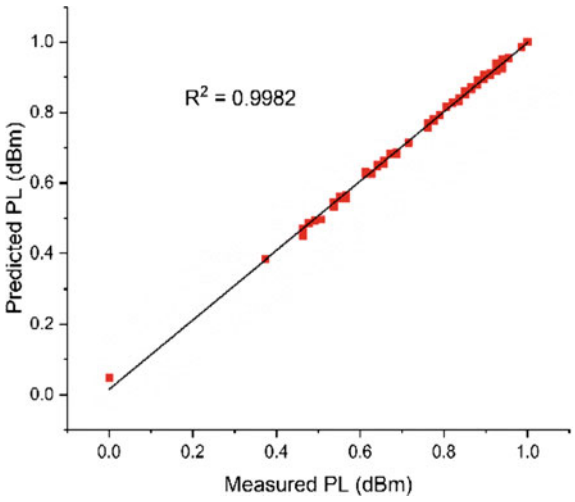


Fig. 9 Measured PL versus predicted PL using BPNN model in LOS

Table 4 Performance of developed models in NLOS case

Model	R^2	MAD	MSE	RMSE
ANN	0.97220	0.02184	0.00131	0.03630
Log-distance	0.93401	0.05183	0.00580	0.07622

Table 5 Performance of developed models in LOS case

Model	R^2	MAD	MSE	RMSE
ANN	0.99820	0.00439	0.00005	0.00773
Log-distance	0.90216	0.04901	0.00447	0.06688

8 Conclusion

In this article, a machine learning algorithm is included along with data measurement, empirical, and regression analysis to estimate the path loss of the ZigBee network more precisely. To achieve this, a ZigBee-assisted test setup operating on 2.4 GHz carrier frequency has been established, and a measurement campaign was performed for data collection in an urban area. In this context, two typical propagation scenarios such as NLOS and LOS were evaluated to employ their adversity to transmission reception, and accordingly, their impact on the performance of the ZigBee network. During the experiment, recorded the average RSSI, distance, temperature, humidity, transmitter antenna gain, and carrier frequency at distinct locations. It is observed that the Log-Distance empirical model suits to represent the relationship between measured and predicted PL with allowable R^2 of 0.93401 and 0.90216 and RMSE of 0.07622 and 0.6688 in both cases. However, the proposed machine learning

model such as BPNN provides significant R^2 (0.97220 and 0.99820) and less RMSE (0.03630 and 0.00773) for NLOS and LOS, respectively. Hence, the proposed BPNN ensures more accurate path loss estimation compared with empirical model.

References

1. Popola SI, Atavero AA, Arausi OD, Matthews VO (2018) Path loss dataset for modeling radio wave propagation in smart campus environment. Data Brief 17:1062–1073
2. Jawad HM, Jawad AM, Nardin R, Gharghan SK, Abdullah NF, Ismail M, Abu-AlShaeer MJ (2019) Accurate empirical path-loss model based on particle swarm optimization for wireless sensor networks in smart agriculture. IEEE Sens J 20:552–561
3. Pan H, Shi Y, Wang X, Li T (2017) Modeling wireless sensor networks radio frequency signal loss in corn environment. Multimed Tools Appl 76(19):19479–19490
4. Egi Y, Otero K (2019) Machine-learning and 3D point-cloud based signal power path loss model for the deployment of wireless communication systems. IEEE Access 7:42507–42517
5. Jo HS, Park C, Lee E, Choi HK, Park J (2020) Path loss prediction based on machine learning techniques: principal component analysis, artificial neural network, and Gaussian process. Sensors 20(7):1–23
6. Correia FP, Alencar MS, Carvalho FB, Lopes WT, Leal BG (2013) Propagation analysis in precision agriculture environment using XBee devices. In: 2013 SBMO/IEEE MTT-S international microwave & optoelectronics conference (IMOC). IEEE, Brazil, pp 1–5
7. Wang D, Song L, Kong X, Zhang Z (2012) Near-ground path loss measurements and modeling for wireless sensor networks at 2.4 GHz. Int J Distrib Sens Netw 8(8):969712
8. Sarma AD, Pandit SNN, Prasad MVSN (2000) Modelling of path loss using adaptive propagation technique for land mobile CM and MM wave communication systems. IETE Tech Rev 17(1–2):37–41
9. Ragam P, Nimaje DS, Yadav D, Karthik G (2019) Performance evaluation of LoRa LPWAN technology for IoT-based blast-induced ground vibration system. J Meas Eng 7(3):119–133
10. Nguyen H, Bui XN, Nguyen-Thoi T, Ragam P, Moayedi H (2019) Toward a state-of-the-art of fly-rock prediction technology in open-pit mines using EANNs model. Appl Sci 9(21):4554
11. Ragam P, Nimaje DS (2018) Evaluation and prediction of blast-induced peak particle velocity using artificial neural network: a case study. Noise Vib Worldwide 49(3):111–119

Comparative Analysis of CNN Models with Vision Transformer on Lung Infection Classification

G. S. S. V. Badrish, K. G. N. Prabhanjali, and A. Raghuvira Pratap

Abstract Lung infections are the most frequently observed medical conditions that affect the respiratory system. These infections can have meager effects like cough and if ignored can lead to the cause of death. Hence, a classification model that helps in the early detection of lung infections can help in avoiding further complications. This paper focuses on techniques to classify lung infections based on different convolution neural network model architectures in comparison with advanced deep learning techniques like Vision Transformer trained on chest X-ray dataset. One of the most powerful imaging techniques for diagnosing lung infections is chest X-ray. So, our model is built on chest X-rays collected from 30,805 individuals that were classified into 15 labels (including no finding). The dataset consists of 112,120 samples of images that are utilized for developing the model for predicting and analyzing lung infections by using InceptionV3, ResNet50, VGG16, InceptionResNet, and Vision Transformer. Each model is evaluated on the basis of mean accuracy error (MAE), binary accuracy, and loss. Among all the models, InceptionResNet has obtained a best accuracy of 93.33%. This study signifies that Vision Transformers are yet to be developed in order to catch up with CNN models.

Keywords Lung infections · Chest X-ray · Vision transformer · InceptionV3 · ResNet50 · InceptionResNet · MobileNet

G. S. S. V. Badrish · K. G. N. Prabhanjali (✉) · A. Raghuvira Pratap
Department of Computer Science and Engineering, V R Siddhartha Engineering College, Vijayawada, India
e-mail: prabhanjalikaturi@gmail.com

A. Raghuvira Pratap
e-mail: pratapadimulam@vrsec.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_12

1 Introduction

The COVID-19 had affected many people's lives from the year 2019. It had affected about 570 million people worldwide. It was proven that due to the occurrence of COVID-19, there is a chance of getting many lung infections, which can in turn cause severe side effects for the people which might lead to death. It can be easy to cure these lung infections if they can be detected early. So, our aim is to predict the lung infections that can be caused to a person using chest X-rays as input with highest accuracy possible. This paper focuses on the comparison of different types CNN models with Vision Transformer for classifying the lung infections that can be caused to a person. Human diagnosis can be a difficult task for the doctors as it takes some time for them to categorize which infection it is and also there are some countries in which there are more population and less medical facilities. In this case, it might be an uphill task for a doctor to diagnose the disease. So, we developed a model that can classify the different lung infections based on the X-ray images of patients. Here, we proposed an automated system for classifying the lung infections using transfer learning of various CNN architectures. A deep learning technique called transfer learning makes use of the previously acquired knowledge and skills in solving a new problem; this is nothing but using an existing architecture to create new model by adding some new layers like dense layer, Global Pooling layer, etc. for achieving better accuracy for the model on a certain dataset by using a specific algorithm. Vision Transformer is a self-attention-based architecture, which is inspired from its applications in natural language processing (NLP) for replacing convolutional neural networks in the classification of computer vision problems.

1.1 Advanced Technologies as a Solution

Dosovitskiy et al. [1] stated that transformer is a technique that is generally being used in natural language processing. Vision Transformer (ViT) tries to replicate the transformers technique in NLP on computer vision problems. It is one of the best alternative approaches to convolution neural network. Initially, the input image is separated into patches of specified dimensions. The number of patches produced will depend on the image resolution, patch size, and specified stride. Once the patches are generated, each patch is encoded with its corresponding position, and then these patches will be flattened to pass as input to the state-of-the-art transform encoder. Then this model will be pretrained on the dataset. Vision Transformers are good in removing inductive bias that is generally observed in convolutional neural networks. It is observed that Vision Transformers perform equally with CNN on small datasets and give better performance than CNN on large datasets.

1.2 Existing Models and Limitations

There are many existing models available in convolution neural network (CNN) that uses transfer learning for training the model and can automatically learn and extract features from the images based on training. All these models are used to classify the lung infections based on a particular algorithm and specify the output (accuracy). Some of them also used various image segmentation techniques for increasing the accuracy of their defined model and compared both models with and without applying segmentation. Some of them used the Inf-Net for extracting the low-level features, and the CT scans are fed into the two-layered convolutional neural networks for improving the representation of the boundaries. Some of them also used the ultrasound imagery of patients for classifying pneumonia affected, COVID-19 infected, and healthy people.

Drawbacks

- The applications take only limited chest X-ray images to detect the diseases.
- Only limited number of diseases are available in the models.

1.3 Contributions of Proposed System

- To analyze the existing problems that are in detecting the lung infections.
- To build a CNN-based model that can classify lung infections.
- To apply several variants of deep convolutional neural networks by using transfer learning and comparing them.
- To perform comparative analysis on the ViT model in comparison with CNN models.

2 Related Work

Souid et al. [2] suggested a system called "Classification and Predictions of Lung Diseases from Chest X-rays Using MobileNetV2." He researched that these days, the effective and widely used medical imaging procedure is the thoracic radiography, which is generally termed as the chest X-rays. On NIH Chest X-ray 14 database, a modified MobileNetV2 algorithm using transfer learning is applied, for the classification and prediction of the lung pathologies that were developed in the frontal thoracic. From 30,805 unique patients, about 112 front-to-back and back-to-front thoracic images were collected and placed in chest X-ray14 database. The metric that was used for evaluating the models is AUC, i.e., integral value of receiver operating characteristic curve. The accuracy that they have achieved is above 90% combined with 81% AUC score.

Rahman et al. [3] proposed a model named "Reliable Tuberculosis Detection Using Chest X-Ray with Deep Learning, Segmentation and Visualization." He inferenced that tuberculosis is a chronic lung disease that might lead to death. Hence, it is necessary to detect the presence of tuberculosis in the early stage to avoid further complications. Using image processing, image segmentation, data augmentation, and deep learning classification techniques, a model is designed for diagnosing tuberculosis with chest X-ray as input. A new database is created for training the model, from the several existing databases. The database consists of about 3500 unaffected X-rays and 3500 TB-infected patients X-rays. Nine different types of deep convolutional neural networks, namely Inception V3, VGG19, MobileNet, ResNet, ChexNet, DenseNet, and SqueezeNet are used with pretrained initial weights and tested for the binary classification of TB. Without using segmentation, the ChexNet had produced an overall accuracy of 96.47%, whereas with using segmentation, 98.6% accuracy is observed for DenseNet.

Zuo et al. [4] suggested a model called "Multi-Resolution CNN and Knowledge Transfer for Candidate Classification in Lung Nodule Detection." He proposed a system that can be used by doctors for diagnosing the lung cancer in the early stages by making use of lung nodule detection automation. As there can also exist the multiple nodules and also varying features of various resolutions of lung nodules, so a multi-resolution convolutional neural network (CNN) is used for feature extraction by using deep layers in the network for classification of lung nodule candidates. The suggested model consists of three stages for its execution. First, A CNN model which is capable of performing edge detection is taken, and its weights are transferred to build a new model capable of classifying images, i.e., multi-resolution model. After that, the knowledge is transformed to improvise the classification performance by taking side-outputs into consideration. Finally, for training and testing the classifier, the samples production as well as data enhancement is performed. The three different sizes of test samples are considered (26 * 26, 36 * 36, and 48 * 48) for testing the multi-resolution CNN and recorded an accuracy of over 92.81% by the three models.

Fan et al. [5] suggested a model on "Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images." He proposed that generally the difficulty in classification can be due to the fact that the infection characteristics contain high variance and due to the difference in intensities between infections. For this issue, a novel COVID-19 lung infection segmentation deep neural network (Inf-Net) is suggested to automatically locate infected regions from chest CT scan images. Based on the architecture of the Inf-Net, for the extraction of the low-level features, the CT images are passed into two-layered convolution neural network and added with a edge attention module for externally improving the representation of objective region boundaries.

Ohata et al. [6] suggested a model called "Automatic Detection of COVID-19 Infection Using Chest X-Ray Images Through Transfer Learning." Here, they used the chest X-ray scans; they detected the COVID-19 infections by using the model that is developed by using the convolutional neural networks, which is trained on relatively small sample consisting of 194 X-ray images collected from patients. For extracting the most important features of images of CT scans, they used transfer

learning technique which uses the transfer knowledge on the initial trained dataset developed by using convolution neural networks.

Diaz-Escobar et al. [7] suggested a model called "Deep-learning based detection of COVID-19 using lung ultrasound imagery." Here, for the detection of COVID-19 and diagnosing pulmonary conditions of patients, a lung ultrasound imagery model using deep learning is adapted with some of the pretrained deep learning architectures like VGG, Inception, Xception, and ResNet. A publicly available POCUS dataset is used for training the model. It consists of about 3326 ultrasound imagery of patients who are infected with COVID-19, pneumonia, and also healthy people.

Anthimopoulos et al. [8] suggested a model called "Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network." He proposed that in the diagnosis of interstitial lung diseases using the aid of computers, automated tissue characterization is identified as the one of the most important components. Generally, CAD system consists of three stages for the lung CT scan assessment, namely segmentation of lungs, quantification of lung diseases, and differential diagnosis. Here, about 14,696 images which are derived from 120 CT scans are trained for evaluating the convolutional neural network. They used a convolutional neural network that consists of five activations which were then forwarded to average pooling layer, and finally, it was fed into three successive dense layers for achieving better accuracy.

Jelodar et al. [9] made use of transformers and LSTM on textual data collected from social media to analyze how COVID-19 affected the mental health of individuals. Although in this paper, the transformers are used on textual data which can be taken as inspiration for implementation of Vision Transformers.

Yaseliani et al. [10] suggested an ensemble model consisting of support vector machine and linear regression for the classification of pneumonia. The model also made use of radial basis function and trained with chest X-rays over several epochs. This suggested model has achieved an accuracy of 98.55%. Although the model had provided a higher accuracy, it can be made more robust by making use of deep learning algorithms.

Zhang et al. [11] had performed a comparative analysis for the classification of pneumonia using various CNN models. The findings suggested that EfficientNet model has given a better performance than any other model. The model made use of ultrasound lung imagery for training. The disadvantage in this study is that the model is trained on limited dataset.

Boban et al. [12] had performed a comparative analysis for the classification of lung diseases using various ML models. The results show that SVM has given an accuracy of 70.45%, whereas accuracy provided by KNN is 99.2%. The model can be improved by making use of ensemble models and deep learning models.

3 Proposed Model

3.1 Dataset Description

The dataset for the lung infections classification is taken from Kaggle, which is named as NIH Chest X-ray Dataset. This dataset comprises of 112,120 chest X-rays which were collected from 30,805 distinct patients and were labeled with 14 types of lung infections. The labels are extracted from raw text using NLP; hence, there might be some discrepancies, but the labels are determined to be more than 90% accurate. The label frequency of NIH Chest Dataset has been given in Fig. 1.

Figure 2 describes about the different types of lung infections that are available in the dataset. This NIH dataset contains 15 classes (14 diseases, and one for "no findings").

The proposed system architecture shown in Fig. 3 is used to predict the lung infections that are caused to a patient post the recovery from COVID-19. The data is pre-processed and is modeled by using the above algorithms like basic CNN, Inception V3, ResNet, and VGG16. The model with the high accuracy is applied on the post-COVID lung dataset and applied on the model, to achieve the result.

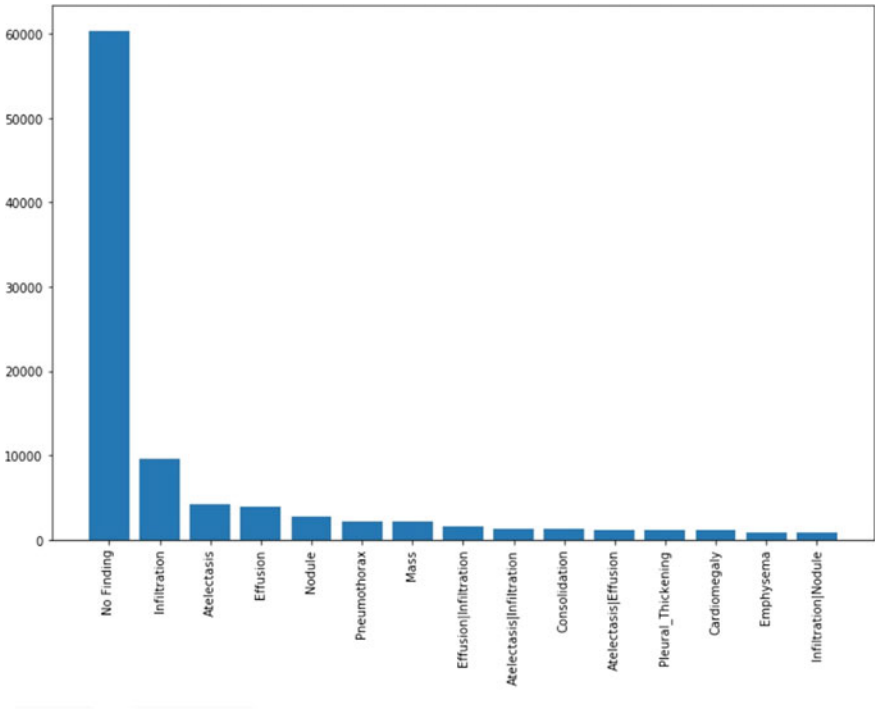


Fig. 1 NIH chest X-ray dataset labels frequency

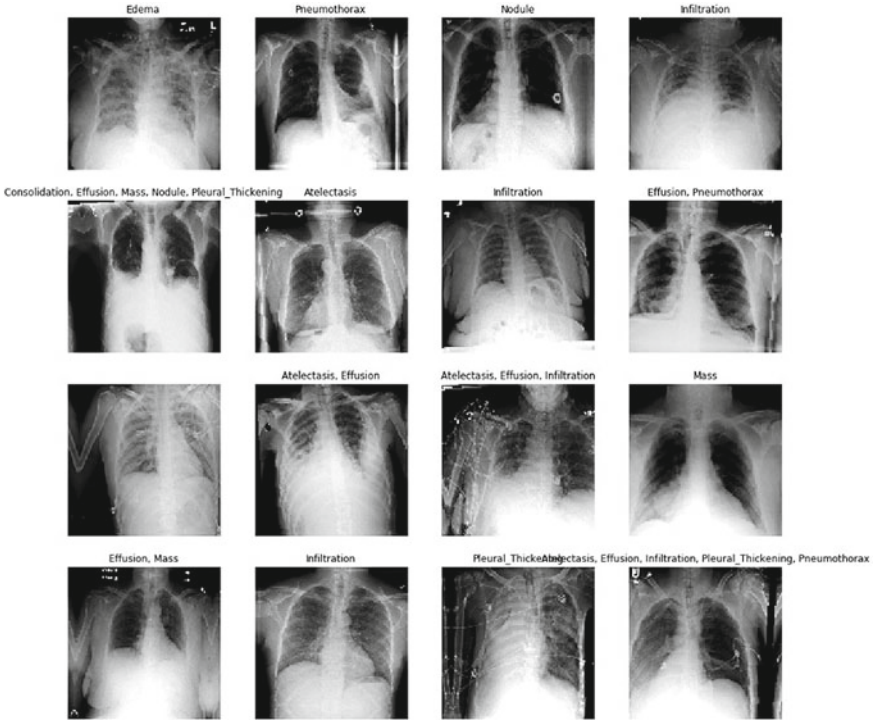


Fig. 2 Different types of lung infections

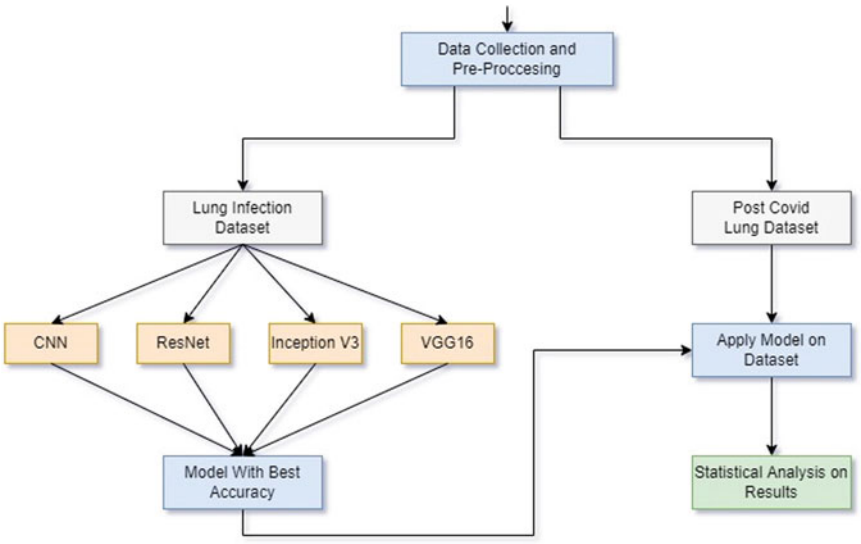


Fig. 3 Proposed system architecture

A. **Pre-processing**

The performing of pre-processing the images with the help of keras data generator class was happened in this step. Input image is transformed into a default image size, later image directories are loaded, and every image is converted in to the array. The next step in image pre-processing is fit_transform of label binarize which is used on the label list.

The dataset is segregated into training data and validation data in which the training data consists of both the input (chest X-rays) and its corresponding class labels and the validation data consists of images which are fed into the neural network as input and the output is compared against its corresponding label in validation data.

The image data pre-processing techniques such as samplewise_center, samplewise_std_normalization, horizontal_flip, vertical_flip, width_shift_range, rotation_range, shear_range, fill_mode, zoom_range are used.

A. **Building the Model**

1. ***Creating the Model***

Generally, there are many variant models for the convolutional neural networks, but one has to choose the model that suites the problem description and gives higher accuracy. Pretrained convolutional models can be obtained from the concept of transfer learning, and these can be used to train on our data by adding some additional trainable layers.

2. ***Building the Architecture of Model***

The pretrained model is added with some other new layers such as Global Pooling layers, dropout layers, dense layers etc. to increase the accuracy of the model. Each layer is associated with an activation function such as relu, softmax, sigmoid based on necessity of problem.

3. ***Training and Testing the Data***

The model is built on 89,696 images which are categorized into 15 labels, and the validation contains 22,424 images. During the training process, the optimization algorithm used is RMSprop which comes under adaptive learning rate method. The model is trained with 0.001 as learning rate. The activation functions used in the model are sigmoid and relu. The model is trained using Batch Gradient Descent with 1000 epochs and 128 steps in each epoch. The metrics used for evaluating the model are accuracy, mean absolute error (MAE), and loss. Here, the loss function used is categorical cross-entropy. True positives (TP) are the cases that the model classified as true labels accurately. False negative (FN) corresponds to the instances that the model classified as false labels accurately. False positives (FP) represent the cases that the model classified as true labels inaccurately. True negatives (TN) represent are the cases that the model classified as false labels inaccurately. The equations for accuracy and MAE are presented on (1)–(2), respectively.

$$\text{Accuracy } (\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \times 100 \tag{1}$$

$$\text{MAE} = (1/n) * \sum |y_i - x_i|. \tag{2}$$

4. ***Saving the Model***

After training and testing, the model is evaluated with the accuracy of 93.33%. Later, the model is saved, and the model is dumped into the application using tensor flow very easily.

Algorithm: CNN-based model
Step-1: Take input dataset
Step-2: Perform data pre-processing on datasets using keras
Step-3: Divide the dataset into training and validation sets
Step-4: Import a variant of CNN using Transfer learning
Step-5: Add a Global Pooling layer
Step-6: Add the Normalization layer to the model
Step-7: Add the flatten layer to the model
Step-8: Add the dense layer to the model
Step-9: Add softmax & compilation function to the model
Step-10: Fit the model with the dataset
Step-11: Return the accuracy and roc curve as the output
Step-12: End

Algorithm: Vision Transformers
Step-1: Take the input dataset
Step-2: Perform data pre-processing on datasets using keras
Step-3: Divide the dataset into training and validation sets
Step-4: divide each image into patches
Step-5: Flatten all the patches of an image
Step-6: Add positional encodings to the patches
Step-7: Feed this to the state-of-the-art transformer encoder as input
Step-8: Add multilayer perceptron to the above architecture
Step-9: Pre-train the model using the given dataset
Step-10: Return accuracy and roc curves as the output
Step-11: End

Algorithm: Lung Infection detection
Step-1: Take the Chest X-ray image
Step-2: Convert the image into a numpy array
Step-3: Input = numpy array of image
Step-4: if (Input image infection is detected)
Output = label of the detected infection
Step-5: else
Output = not detected
Step-6: End

4 Results and Analysis

The detailed analysis of different models used for classifying the lung infections is given in Table 1. The metrics used for evaluating the model are loss, binary accuracy, and mean absolute error (MAE).

From the shown table and graph (Fig. 4), the summary of the performance of the different models is analyzed. It is determined that the InceptionResNetV2 model is possessing high accuracy (93.33%) when compared to all the other models. Although Vision Transformer model is very promising as an alternative to convolutional neural network models, its performance on the given dataset (80.90%) is far from better, and it is outperformed by CNN models such as InceptionResNetV2, MobileNet, VGG16.

Figure 5 shows the lung infection classification of an chest X-ray image. Here, the true labels for the above image are consolidation and infiltration which were correctly classified, i.e., true positives.

Table 1 Summary of models used

Model	Loss	Binary_accuracy	MAE
InceptionResNetV2	0.298	93.33	0.1548
MobileNet	0.342	87.55	0.1721
VGG16	2.305	84.28	0.1572
ResNet50	0.515	87.21	8.1572
InceptionV3	5.886	63.18	0.3682
Vision transformer	5.678	80.90	37.994

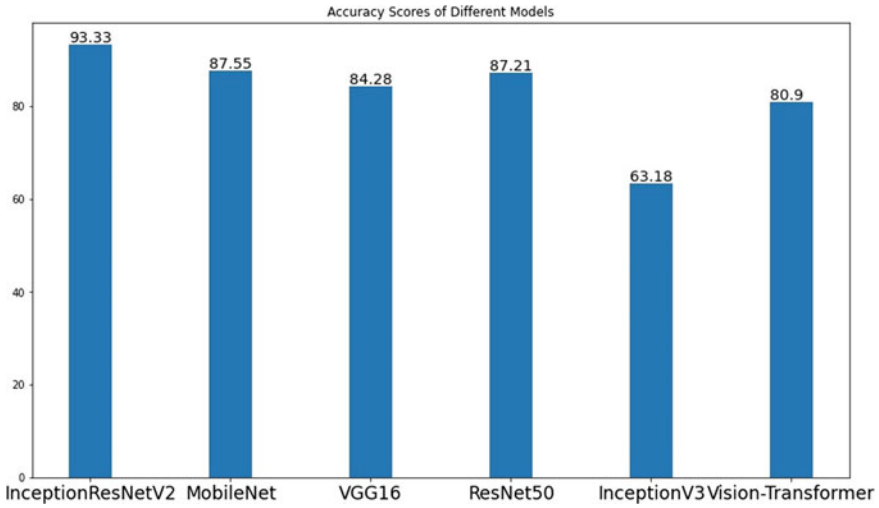


Fig. 4 Accuracy scores of different models

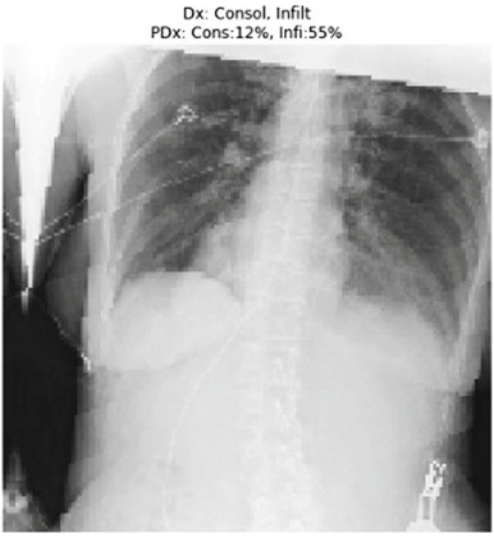


Fig. 5 Classification output

Figure 6 shows the lung infection classification of an chest X-ray image with false positive. Here, the true labels for the above image are atelectasis and cardiomegaly. But the model's output is atelectasis, cardiomegaly, and infiltration in which atelectasis and cardiomegaly are true positives and infiltration is false positive.

In Fig. 7, Dx and PDx describe the actual and predicted frequency percentages of corresponding labels. From the figure, we can observe that Dx and PDx values

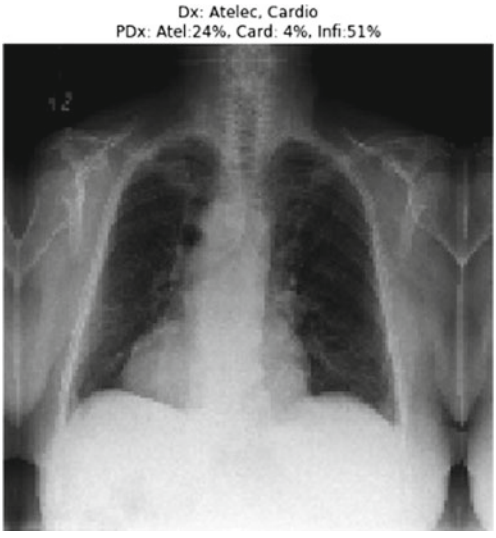


Fig. 6 Classification output with false positive

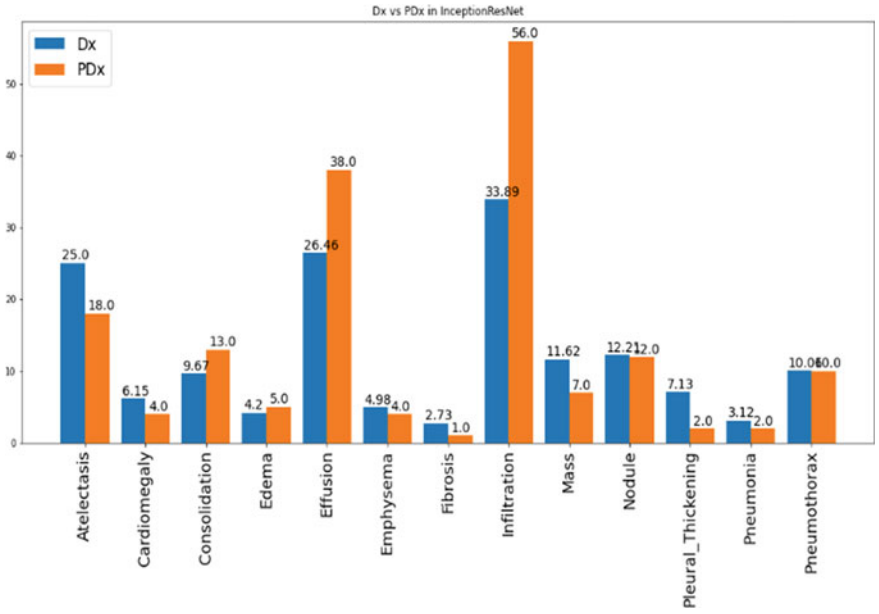


Fig. 7 Actual and predicted frequency percentage of labels in InceptionResNet

of atelectasis are 25% and 18.92%, respectively; as Dx value is greater than corresponding PDx value, this label may constitute to true negatives. Similarly, the Dx and PDx values of infiltration are 33.89% and 56.01%, respectively; here, the Dx value is less than the PDx value. Hence, this label may constitute to false positives. For the labels such as pneumothorax, emphysema, and nodule, the Dx and PDx values are almost similar which represent that these labels constitute to lower true negatives and false positives in the model's prediction.

Figure 8 shows the actual and predicted percentage of labels in Vision Transformer. We can observe that the InceptionResNet has predicted more accurately when compared to the Vision Transformer. For instance, take a look at PDx values of atelectasis which are 18.0%, 16.0%, respectively, for InceptionResNet, ViT for a Dx value of 25.0%. Here, the PDx value of InceptionResNet is much closer to Dx value than that of ViT. Similarly, this trend can be observed for almost every label; i.e., for 12 out of 14 labels, InceptionResNet has much smaller gap between Dx and PDx than ViT.

Figure 9 describes about the comparison of the Dx-PDx differences between InceptionResnNet and ViT. The smaller the value of difference, the better the model performed. From the bar plot, we can clearly observe that the difference value is significantly smaller for InceptionResNet except for effusion and Pleural_Thickening. Hence, we can safely infer that InceptionResNet provides better diagnosis in classification than Vision Transformer.

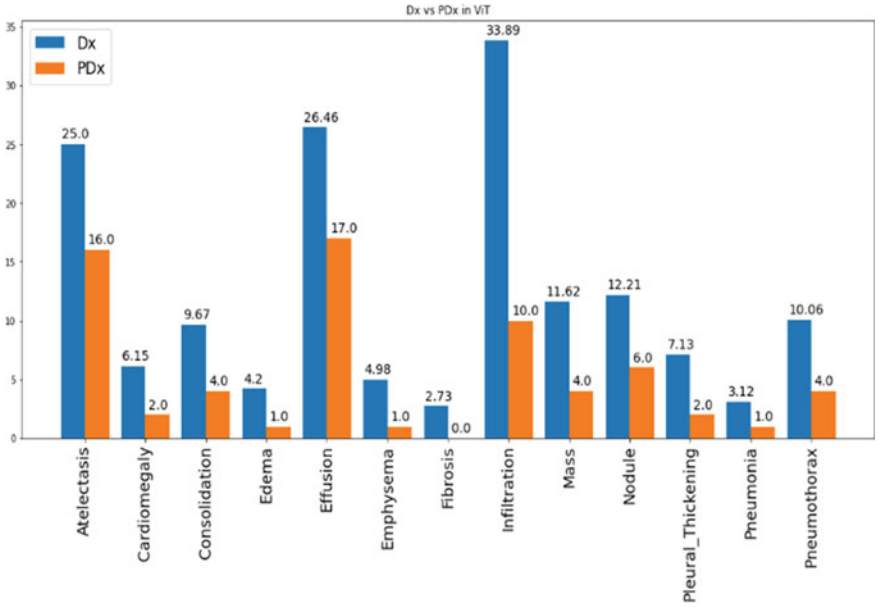


Fig. 8 Actual and predicted frequency percentage of labels in ViT

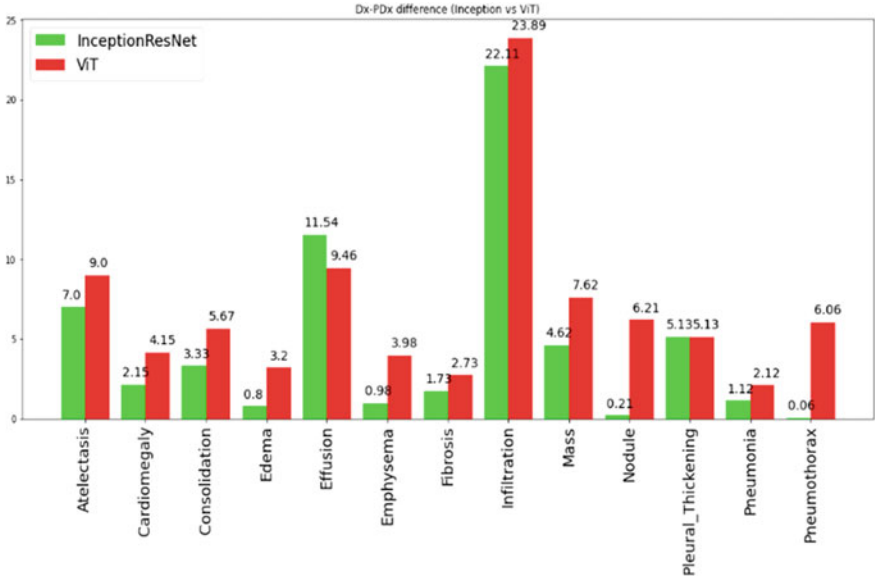


Fig. 9 Comparison of Dx-PDx differences between InceptionResNet and ViT

Figure 10 describes about the ROC curve for the InceptionResNetV2. Generally, an ROC curve is used for plotting change in true positives against the change in false positives at different classification thresholds. It is observed that edema is having an higher AUC of 0.75, and cardiomegaly is having an AUC of 0.57. Hence, these two lung infections are more accurately classified by the convolutional neural network models while validation.

Figure 11 shows the ROC curve for ViT in the classification of lung infections from the plot; it can be clearly observed that the highest AUC value is 0.58 for atelectasis, and the minimum AUC value is 0.43 for pneumothorax. These value are clearly much lesser than those that we have observed in InceptionResnetV2 model. Hence, we can easily infer that the ViT model has much higher false positive rate than InceptionResNet.

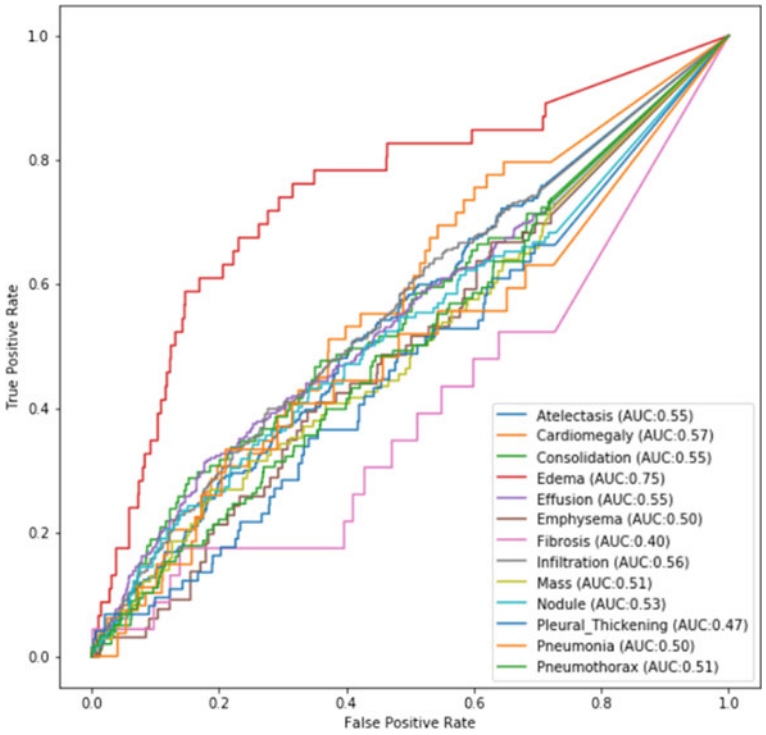


Fig. 10 ROC curve for InceptionResNetV2

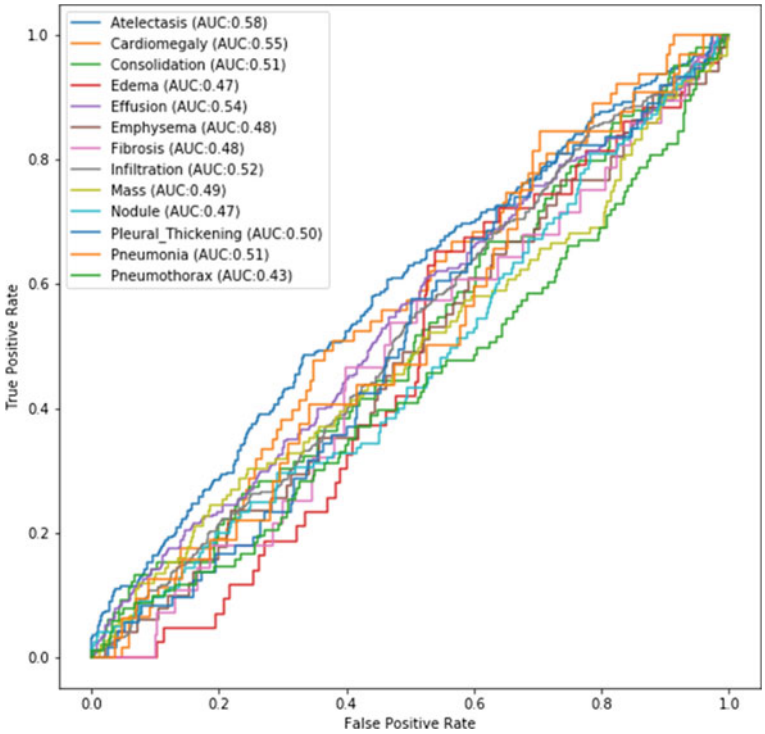


Fig. 11 ROC curve for ViT

5 Conclusion and Future Work

Deep learning is a modern and sophisticated method in the realm of image pattern recognition and the detection of any kind of diseases. In this paper, the proposed model uses CNN as compared to Vision Transformer to effectively identify the different types of lung diseases based on frontal X-ray images of chest. The suggested model is evaluated on 22,424 frontal X-ray images of chest, i.e., 20% of total dataset, and based on the comparison of the CNN models and Vision Transformers, the InceptionResNetV2 achieved the highest accuracy score of 93.33%. In the future, the suggested model can be extended to predict and analyze the lung infections that are caused to a patient post the recovery from COVID-19, and a mobile application can also be developed using the model that facilitates the users to simply upload their chest X-ray for identifying lung infections.

References

1. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: transformers for image recognition at scale. ArXiv, abs/2010.11929.
2. Souid A, Sakli N, Sakli H (2021) Classification and predictions of lung diseases from chest X-rays using MobileNet V2. Appl Sci 11:2751. https://doi.org/10.3390/app11062751
3. Rahman T et al (2020) Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access 8:191586–191601. https://doi.org/10.1109/ACCESS.2020.3031384
4. Zuo W, Zhou F, Li Z, Wang L (2019) Multi-resolution CNN and knowledge transfer for candidate classification in lung nodule detection. IEEE Access 7:32510–32521. https://doi.org/10.1109/ACCESS.2019.2903587
5. Fan D-P et al (2020) Inf-Net: automatic COVID-19 lung infection segmentation From CT images. IEEE Trans Med Imaging 39(8):2626–2637. https://doi.org/10.1109/TMI.2020.2996645
6. Ohata EF, Bezerra GM, das Chagas JVS, Lira Neto AV, Albuquerque AB, de Albuquerque VHC, Reboucas Filho PP (2020) Automatic detection of COVID-19 infection using chest X-ray images through transfer learning. IEEE/CAA J Autom Sinica, 1–10 (2020). https://doi.org/10.1109/jas.2020.1003393
7. Diaz-Escobar J, Ordoñez-Guillen NE, Villarreal-Reyes S, Galaviz-Mosqueda A, Kober V, Rivera-Rodriguez R et al. (2021) Deep-learningbased detection of COVID-19 using lungultrasound imagery. PLoS ONE 16(8):e0255886. https://doi.org/10.1371/journal.pone.0255886
8. Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S (2016) Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. IEEE Trans Med Imaging 35(5):1207–1216. https://doi.org/10.1109/TMI.2016.2535865
9. Jelodar H, Wang Y, Orji R, Huang S (2020) Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. IEEE J Biomed Health Inform 24(10):2733–2742. https://doi.org/10.1109/JBHI.2020.3001216
10. Yaseliani M, Hamadani AZ, Maghsoodi AI, Mosavi A (2022) Pneumonia detection proposing a hybrid deep convolutional neural network based on two parallel visual geometry group architectures and machine learning classifiers. IEEE Access 10:62110–62128. https://doi.org/10.1109/ACCESS.2022.3182498
11. Zhang J et al (220) Detection and classification of pneumonia from lung ultrasound images. In: 2020 5th international conference on communication, image and signal processing (CCISP), pp 294–298 (2020). https://doi.org/10.1109/CCISP51026.2020.9273469
12. Boban BM, Megalingam RK (2020) Lung diseases classification based on machine learning algorithms and performance evaluation. In: 2020 international conference on communication and signal processing (ICCSP), pp 0315–0320.https://doi.org/10.1109/ICCSP48568.2020.9182324

Classification of Alzheimer's Disease Using Stacking-Based Ensemble and Transfer Learning

T. Madhumitha, M. Nikitha, P. Chinmayi Supraja, and K. Sitakumari

Abstract Dementia is collection of traits that are linked with a reduced memory power, thinking abilities or other cognition-related skills. Dementia exists in many different forms and can be caused by many different conditions. However, the most common cause of dementia is Alzheimer's. Alzheimer's can be described as one specific disease (Beer et al in The Merck manual of diagnosis and therapy. Merck Research Laboratories, 1999 [1]). In recent years, many deep learning approaches to classify brain images with respect to the Alzheimer's disease are being proposed, and a tremendous amount of research is being done in this area. However, these approaches are still not being used actively in the medical field due to apprehensions about their accuracy and due to a general lack of appropriate medical data. This paper aims at introducing an approach to classify the Alzheimer's MRI image data into four different stages. The approach produces efficient and accurate results and is designed to encourage the implementation of deep learning in day-to-day medicine without the need for much human involvement. In the proposed method, we use transfer learning to employ three pre-trained deep learning models to perform the task of classification and combine them through a stacked ensemble, which can then be used for the purpose of predictions. To carry out this approach, the Alzheimer's data set (four classes of images) from Kaggle, containing MRI brain images, has been used, and the proposed methodology has produced an accuracy of 97.8%. The results have been visualized and presented in this paper.

Keywords Alzheimer's disease · Dementia · Pre-trained model · Transfer learning · Ensemble

T. Madhumitha (✉) · M. Nikitha · P. Chinmayi Supraja · K. Sitakumari
Department of Information Technology, VR Siddhartha Engineering College, Vijayawada, India
e-mail: madhumithat88@gmail.com

K. Sitakumari
e-mail: k.sitakumari@vrsiddhartha.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_13

1 Introduction

Deep learning models have historically been well-established as supervised learning models. However, there is significant research into deep learning models that can compete using different types of machine learning techniques as well. One such type of deep learning model is an ensemble. Ensemble models can be powerful to solve some of the most grueling machine learning tasks. They provide a comprehensive architecture to tackle complex machine learning applications. They can especially be used in the healthcare sector as they have the capability to deal with large data sets with high efficiency. In this paper, we present how the capabilities of ensembles can be leveraged to classify the different stages of the Alzheimer's disease.

The Alzheimer's disease is a neurodegenerative disease which primarily occurs in the brain when the brain cells are deprecating, which destroys the memory and thinking skills, causes memory loss and hinders brain functioning [2]. Alzheimer's is the most common form of dementia, which mostly occurs in older people. This has no proper treatment and diagnosis. However, it has been identified that taking proper measures during the early stages of Alzheimer's can greatly help with preventing the disease from worsening cognitive abilities. Using image pre-processing techniques on MRI scans can increase the chances of detecting the presence of the Alzheimer's disease at an early stage. For early diagnosis, we must focus more on improving classification performances. Medical imaging is an important part of medicine. It is very useful in helping with diagnosis and treatment.

Conventionally, deep neural networks have been used to work with medical imaging data. These methods have shown great accuracies where there is an abundance of data. However, in cases where there is an imbalance of data classes, traditional CNNs show a great bias and produce inaccurate results. Using finely curated technologies can greatly increase accuracies.

Ensembles leverage the powerful learning capabilities of traditional CNNs and combine this knowledge to form a cohesive meta-learning solution.

Our proposed approach aims at identifying and classifying the Alzheimer's disease into its four early stages—non-demented, very mildly demented, mildly demented and moderately demented classes, by making use of transfer learning and stacked ensemble. The motto behind the solution is to encourage the use of deep learning methodologies in everyday medicine and make them more accessible. The proposed work has achieved high accuracies and projects a great scope for practical implementation. The parameters and computational costs of the proposed neural network have been optimized with maximized efficiency.

This paper showcases a study on the currently existing methodologies and introduces an ensemble-based solution to classify the stages of the Alzheimer's disease. The architecture and algorithms of the proposed solution have been presented, followed by an analysis of the results produced by our model.

1.1 Field Study

To obtain a better understanding of people with Alzheimer's, we visited Pinnamaneni Siddhartha Medical College and Hospital, interacted with the neurologists and radiologists, examined the MRI scans of Alzheimer's patients and talked to older people about their memory loss condition and reduced cognitive abilities.

2 Literature Study

Toshkhujaev et al. [3] deduced that, in dealing with neuroimaging data, labeling the different magnetic resonance imaging (MRI) images accurately is crucial to produce practically useful and precise results. This was done by accurately segmenting the data and performing volume measurement. This work primarily focuses on the preprocessing aspects and data preparation.

Bae et al. [4] make use of convolutional neural networks (CNN) to perform image classification on T1-weighted MRI scans. This paper puts forth the idea that, in order to make a widely applicable classification model, the model must be trained with data obtained from various populations. This ensures that the model does not become overfit to a certain niche population due to the lack of variety in the data.

The work of Yamanakkanavar et al. [5] strengthens the assertion that MRI images can be crucial in identifying changes cognitive capabilities. The paper provides an automated method to perform segmentation of MRI images. It points to the evidence that proper segmentation of the MRI data greatly improves the accuracy of Alzheimer's disease classification.

Zhang et al. [6] experimented with a 3D fully connected convolutional neural network (CNN) for the classification of AD. This CNN was used to obtain multiscale features of the MRI images. At each 3D layer, the features from the previous layers were combined to produce a comprehensive classification model.

Tian et al. [7] utilized a multi-stage pipeline for determining the stage of the Alzheimer's disease. The pipeline proposed in this paper includes selection of image quality, generating the vessel map of the data and finally predicting the AD stage.

Our proposed solution leverages the understandings of the above-mentioned works to maximize the accuracy of prediction of the AD stage.

3 Proposed Work

The proposed system uses three base models, which are pre-trained deep learning models, namely InceptionV3, Xception and ResNet50. The concept of transfer learning has been used to employ the pre-trained models in our proposed network.

Table 1 Data set structure

Dementia stage	No. of data samples
No dementia	3200
Very mild dementia	2240
Mild dementia	896
Moderate dementia	64

These models are then combined to create a stacked ensemble, in order to maximize the accuracy.

3.1 Data Set

The proposed model was run on 6400 magnetic resonance images of the brain corresponding to four different Alzheimer's disease stages. The data set was collected from Kaggle [8] (Table 1).

3.2 Pre-processing

The images were pre-processed by applying image augmentation. The augmentation techniques applied are zooming, brightness change, horizontal flip and re-scaling. Synthetic Minority Over-sampling Technique (SMOTE) was performed to balance the data of the minority classes.

A total of 12,800 images were obtained after the pre-processing stage (Fig. 1).

3.3 Design Methodology

The three base models were selected based on the number of parameters and the accuracies they produced on the ImageNet data set.

These models were initially trained individually on our data set in order to optimize their weights to fit the data. The following parameters were initialized in the training process:

Activation function: sigmoid
Optimizer: adam
Learning Rate: 0.0001
Loss: Categorical Cross Entropy.

Each model was run for 25 epochs to obtain the measures below (Table 2).

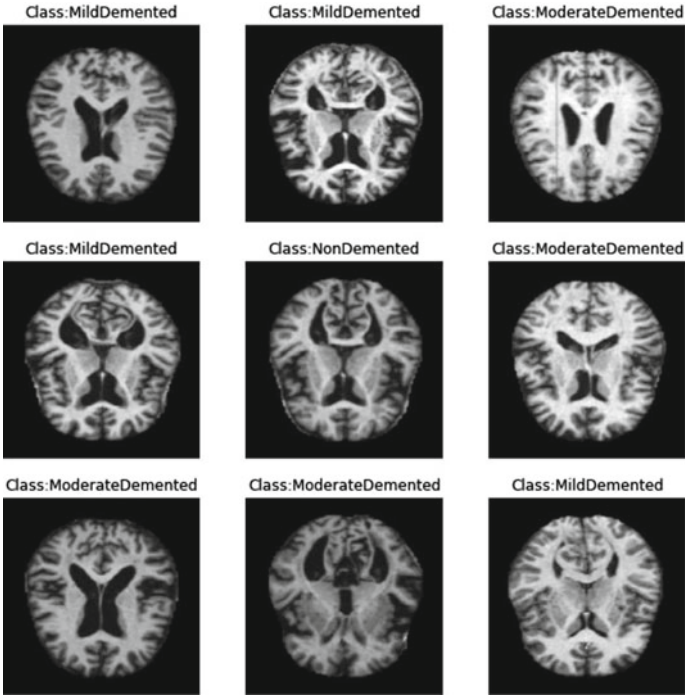


Fig. 1 Sample images from the data set

Table 2 Comparison of performances of base models

	InceptionV3	Xception	ResNet50
Testing accuracy	93.50	82.70	97.50
Validation accuracy	94.05	92.81	97.70
Training accuracy	98.88	99.44	99.90

3.4 Transfer Learning

A machine learning technique called transfer learning uses a model created for one task as the foundation for a model on another task [9]. In the proposed methodology, we used InceptionV3, Xception and ResNet50, three models which were pre-trained for the task of image classification on the ImageNet data set. The three models were then trained to fit our data set to classify the different stages of the Alzheimer's disease.

After fine-tuning of the base models is done by updating their weights, we used them to form a stacking-based ensemble. The main goal of using a stacking-based approach is to bring together a group of independently strong but diverse base learners

[10]. Even though the base models produced commendable accuracies, the ensemble was used to give the accuracy a boost.

Algorithm for Stacked Ensemble [11]

Input: $D = \{(a_i, b_i)|a_i \in A, b_i \in B|\}$**.**

Output: Ensemble H

Step 1: Train the base classifiers
For $b \leftarrow 1$ to B do
Train base-level classifier h_b on data set D
Step 2: Build fresh data set from D
For $i \leftarrow 1$ to m do
Build new data set that has $\left\{a_i^{\text{new}}, b_i\right\}$ where

$$a_i^{\text{new}} = \left\{h_j(a_i) \text{ for } j = 1 \text{ to } B\right\}$$

Step 3: Train the stacked ensemble h^{new} on the newly created data set
Step 4: Return $H(a) = h^{\text{new}}(h_1(a),\ h_2(a), \ldots, h_B(a))$

The architecture diagram of the proposed ensemble model has been presented in Fig. 2.

First, the three base classifiers were selected based on the number of parameters. These classifiers are diverse. Each classifier was then trained with the same data set. The weights of each classifier are updated with respect to the data set, and the network is fine-tuned. The outputs of all the base classifiers were recorded, and their performances were analyzed against each other. The predicted class label outputs of the first-level classifiers are seen as new features, and the original class labels are seen as the new data set labels.

Then, a stacked generalizer with an additional dense layer was developed and was used as a classifier at the second level. Stacked generalization introduces a method for merging the knowledge of multiple models. It was used as it provides an efficient

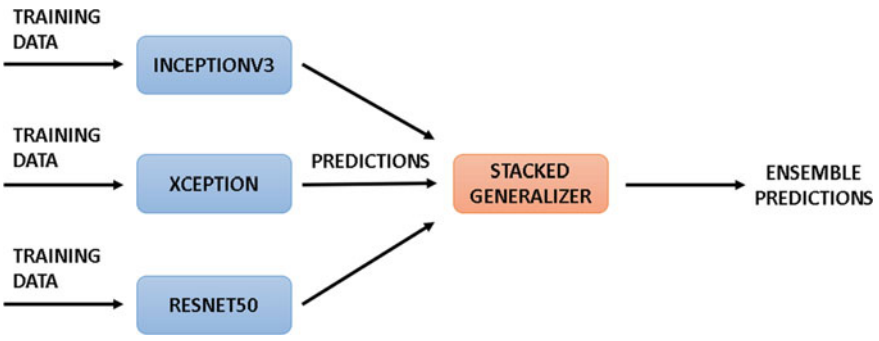


Fig. 2 Proposed ensemble architecture

way for meta-learning by combining the base models nonlinearly, as opposed to the "winner-takes-all" approach [12]. The following parameters were initialized for training the ensemble model:

Activation function: sigmoid
Optimizer: adam
Learning Rate: 0.001
Loss: Categorical Cross Entropy

The trained ensemble classifier produced an accuracy of 97.8%.

4 Results and Observations

The performances of all three base models and the ensemble are presented below.

4.1 Inception V3

The model InceptionV3 is observed to have the least starting point accuracy. But the model performs exceptionally well as the training progresses. The validation set loss is seen to fluctuate heavily during the entire training period (Fig. 3).

The confusion matrix for InceptionV3 is as presented in Fig. 4. The model performs well in distinguishing the NonDemented and VeryMildDemented stages but performs poorly while distinguishing the ModerateDemented data from the MildDemented data.

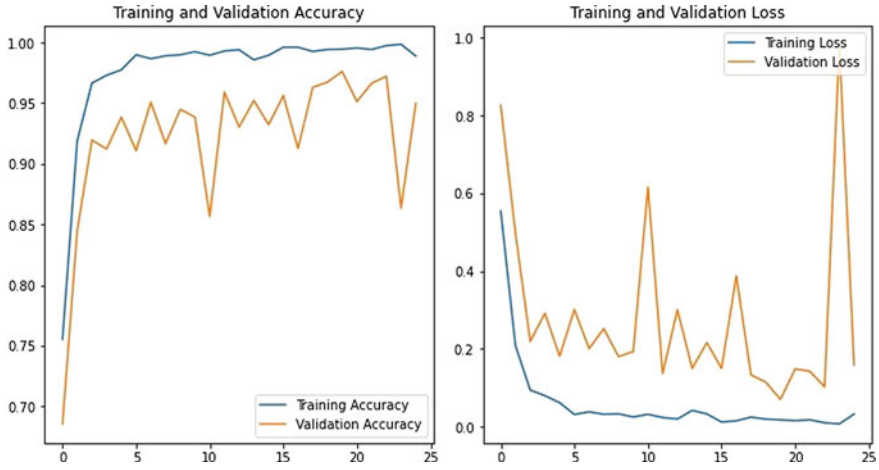


Fig. 3 InceptionV3—plot showing accuracy, loss of training and validation sets

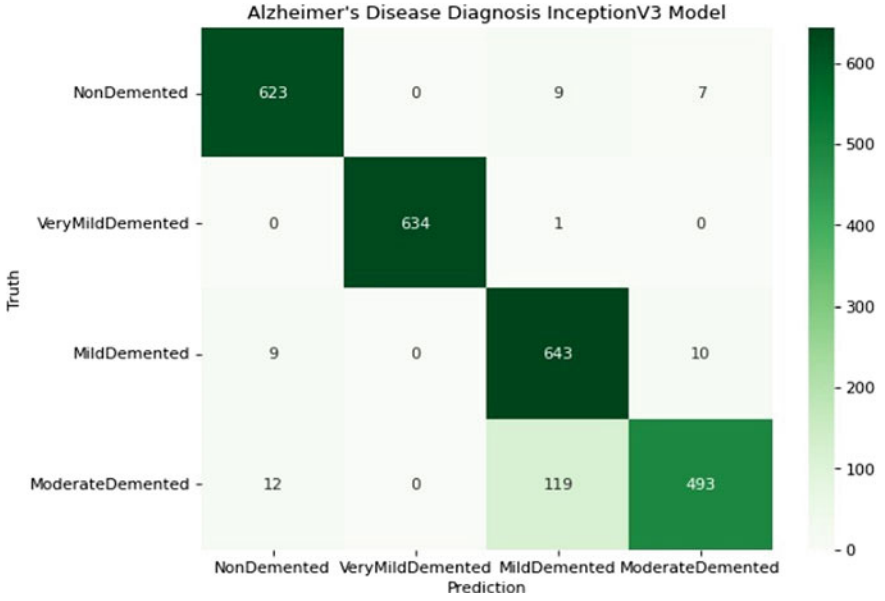


Fig. 4 InceptionV3—confusion matrix

With a testing accuracy of 93.5%, it has a precision of 94%, recall of 93% and an F1 score of 0.93.

4.2 Xception

For the model Xception, the starting point of the accuracy is much higher than Inception. The training curve is a steep plateau. The validation loss does not seem to fluctuate as much as InceptionV3 (Fig. 5).

Figure 6 presents the confusion matrix of Xception. From the confusion matrix presented, we can deduce the fact that the model performs poorly in distinguishing between the MildDemented class and the ModerateDemented class. The performance of InceptionV3 seems to be better than the performance of the model Xception.

Xception performs well for the VeryMildDemented and MildDemented classes but fails to categorize the other two classes well.

With a testing accuracy of 82.7%, it has a precision of 88%, recall of 83% and an F1 score of 0.83.

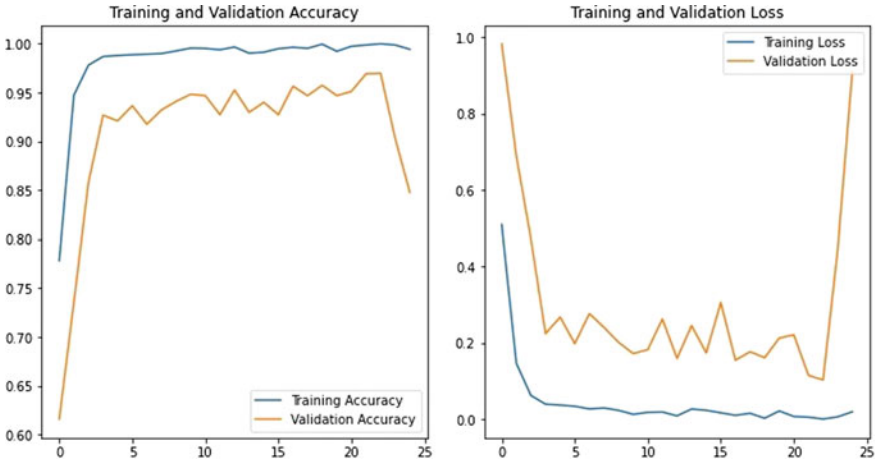


Fig. 5 Xception—plot showing accuracy, loss of training and validation sets

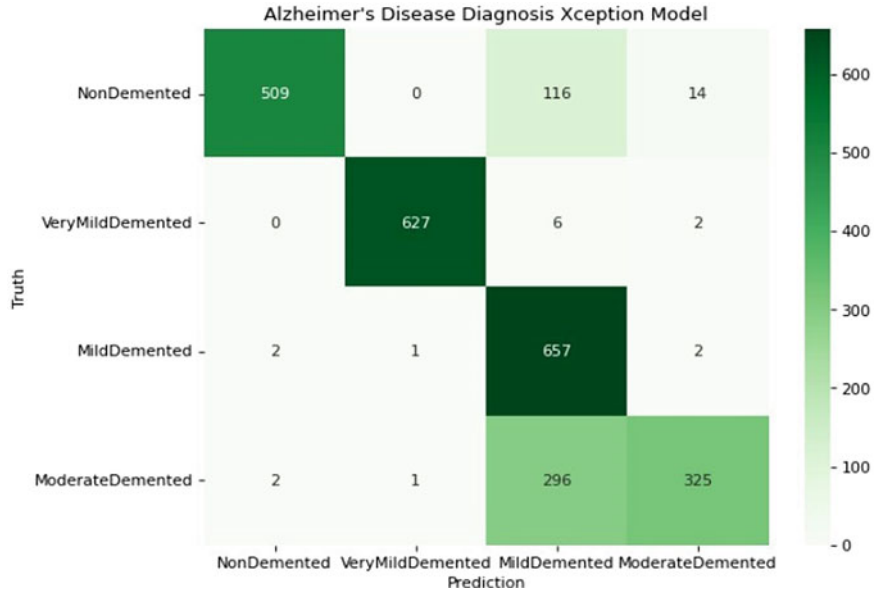


Fig. 6 Xception—confusion matrix

4.3 ResNet50

The performance curves of ResNet50 have been presented in Fig. 7.

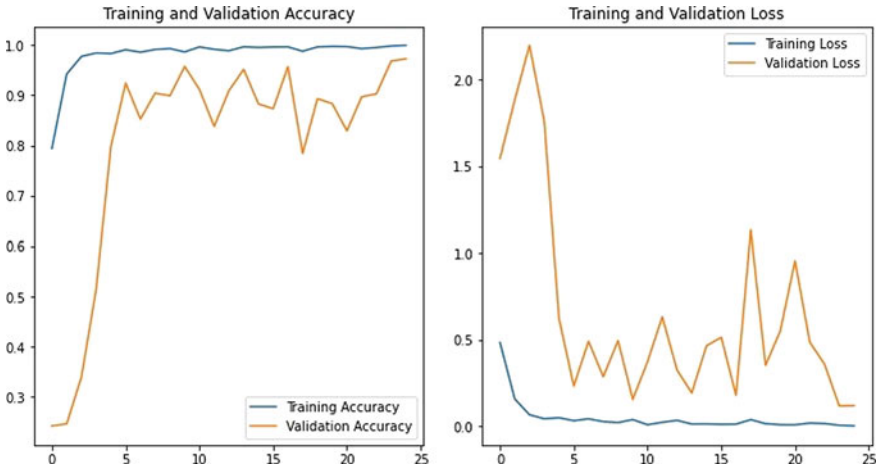


Fig. 7 ResNet50—plot showing accuracy, loss of training and validation sets

For ResNet50, the validation loss fluctuates heavily throughout the training period. But the model performs extremely well toward the end of the training period. Figure 8 presents the confusion matrix of the model ResNet50.

The base model ResNet50 shows the best performance compared to InceptionV3 and Xception. The model performs well at classifying all the classes in the data set.

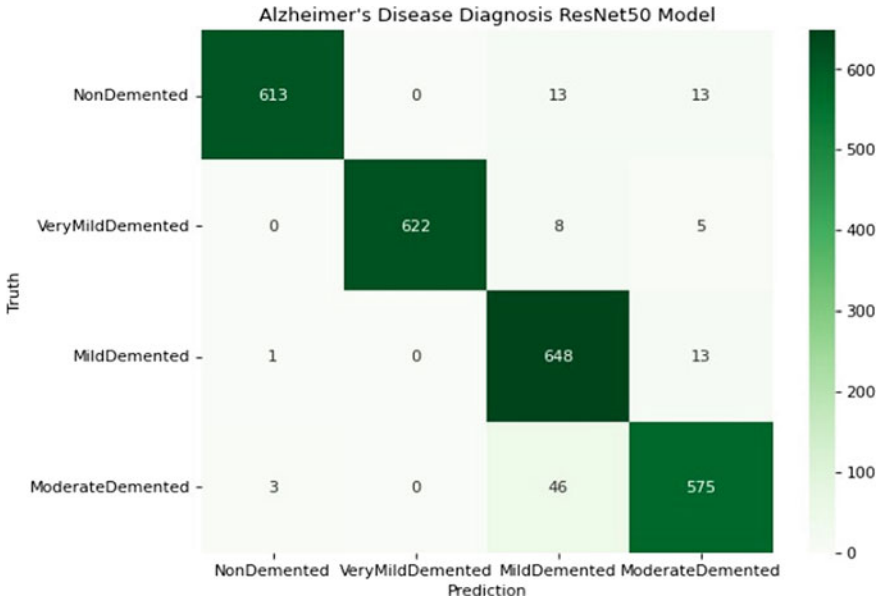


Fig. 8 ResNet50—confusion matrix

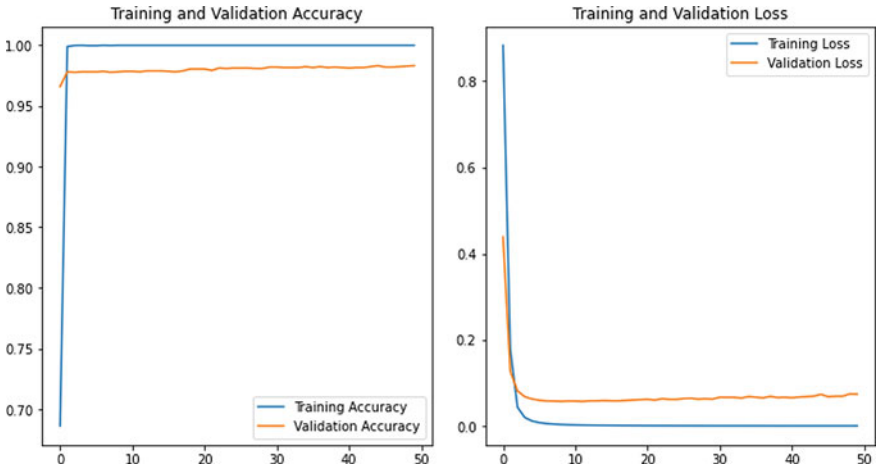


Fig. 9 Proposed ensemble model—plot showing accuracy, loss of training and validation sets

With a testing accuracy of 97.5%, the model has a precision of 96%, recall of 96% and an F1 score of 0.96.

4.4 Proposed Ensemble Model

The learning curve of the ensemble is observed to be extremely steep, and there is barely any fluctuation in the training and validation accuracies and losses. The ensemble model performs extremely well with both the training and the validation data (Fig. 9).

Figure 10 shows the confusion matrix of the proposed ensemble model.

The proposed ensemble model has been shown to give the best predictions when compared with each of the three base models.

With a testing accuracy of 97.8%, precision of 98%, recall of 98% and an F1 score of 0.98, the ensemble model outperformed the base models. The proposed model also outperforms all the other existing models on the taken data set.

Table 3 summarizes the performances of all the models.

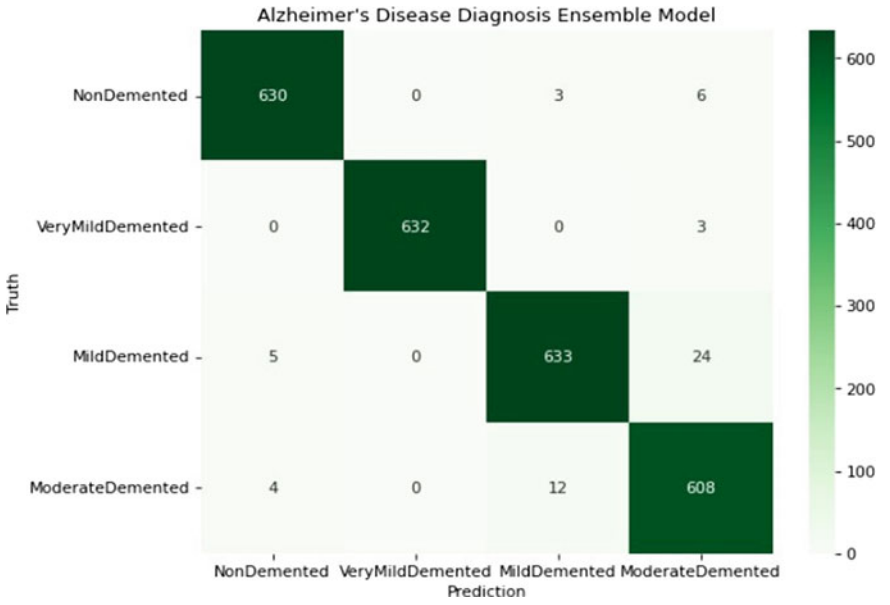


Fig. 10 Proposed ensemble model—confusion matrix

Table 3 Summary of performances of all models

Metric	InceptionV3	Xception	ResNet50	Ensemble
Precision	0.94	0.88	0.96	0.98
Recall	0.93	0.83	0.96	0.98
F1 score	0.93	0.83	0.96	0.98
Testing accuracy	93.50	82.70	97.5	97.8
Validation accuracy	94.05	92.81	97.70	98.24
Training accuracy	98.88	99.44	99.90	99.00

5 Conclusion

The proposed deep learning ensemble model provides a precise and accurate solution to classify the different stages of the Alzheimer's disease from magnetic resonance images of a person's brain. Early detection of AD can be helpful to prevent the disease from further affecting cognitive brain functions. As a large part of the elderly population is affected by the Alzheimer's disease, the proposed solution can serve its purpose and cater to a large demographic of people. The use of MRI images replaces the traditional way of Alzheimer's disease detection which involves the use of cerebrospinal fluid, which can introduce hazardous health complications to the patients. The proposed model can be employed in the medical field for everyday usage as it is more cost effective. The model also produces highly accurate results

and does not put the patient's health at risk. It can be made more accessible to the elderly. In rural areas, where advanced medical equipment or radiologists are not available, the model can come to great use.

The proposed model is also easy to implement as it only requires an MRI image of the brain of the person concerned, as opposed to traditional methods where retinal data or cerebrospinal fluid are required. It is also greatly efficient as it can easily identify and distinguish features in the MRI images which may not be visible to the human eye.

The model can be integrated with web or mobile applications, which then lets users to directly use their MRI images for AD detection. Modern tools and technologies like cloud computing can be used to deploy the model at low costs.

6 Future Work

In the future, we are planning on integrating this model with an app that lets the user upload an image and immediately displays the stage of Alzheimer's disease. We also would like to deploy the app at a low cost and make it more accessible to the people living in rural areas.

References

1. Beers MH et al (1999) The Merck manual of diagnosis and therapy. Merck Research Laboratories
2. Breijyeh Z, Karaman R (2020) Comprehensive review on Alzheimer's disease: causes and treatment. Molecules 25(24):5789
3. Toshkhujaev S et al (2020) Classification of Alzheimer's disease and mild cognitive impairment based on cortical and subcortical features from MRI T1 brain images utilizing four different types of datasets. J Healthc Eng 2020
4. Bae JB et al (2020) Identification of Alzheimer's disease using a convolutional neural network model based on T1-weighted magnetic resonance imaging. Sci Rep 10(1):1–10
5. Yamanakkanavar N, Choi JY, Lee B (2020) MRI segmentation and classification of human brain using deep learning for diagnosis of Alzheimer's disease: a survey. Sensors 20(11):3243
6. Zhang J et al (2021) A 3D densely connected convolution neural network with connection-wise attention mechanism for Alzheimer's disease classification. Mag Resonance Imaging 78:119–126
7. Tian J et al (2021) Modular machine learning for Alzheimer's disease classification from retinal vasculature. Sci Rep 11(1):1–11
8. [Online]. Available: https://www.kaggle.com/datasets/tourist55/alzheimersdataset-4-class-of-images
9. Sun Q et al (2019) Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition
10. Sollich P, Krogh A (1995) Learning with ensembles: how overfitting can be useful. In: Advances in neural information processing systems, vol 8
11. Zhou Z-H (2012) Ensemble methods: foundations and algorithms. CRC Press
12. Wolpert DH (1992) Stacked generalization. Neural Netw 5(2):241–259

Heart Device for Expectation of Coronary Illness Utilizing Internet of Things

P. Kumar, S. Vinod Kumar, and L. Priya

Abstract As of late, there has been a quick development of medical care administrations for giving remote correspondence media among specialist and patient through wearable innovations, which is alluded to as "telemedicine." The reason for the curio is to offer continuous checking of constant circumstances like cardiovascular breakdown, asthma, hypotension, hypertension, and so on that are situated a long way from clinical offices, like provincial areas or for a briefly not individual getting clinical consideration. Because of a change in way of life that influences all age gatherings, coronary illness turns into the primary driver of mortality in every single such case. Writing reports that practically 2.8 billion people overall pass away from coronary illness because of being overweight or corpulent, which ultimately influences circulatory strain swings, cholesterol levels, and in particular the effect of pressure chemicals on hidden heart sicknesses. Numerous wearable innovations screen commonplace heart working pointers like circulatory strain, glucose level, blood oxygen immersion, and ECG. The proposed framework might gather the fundamental information while barring commotion unsettling influences by at the same time checking a few boundaries integrated into wearable gadgets utilizing a solitary chip. It endeavors to keep its precision by limiting human collaboration. A constant demonstrative gadget involves biomedical sensors to gauge different boundaries in heart-weak patients who are somewhat found. Doctors can hold and view various accumulated perceptions sometime in the future for a precise determination. Consequently, the proposed structure will lay out an emotionally supportive network that will permit the specialist to electronically screen the patients' wellbeing boundaries while he is away. The actual impression of a specialist cardiologist would clearly be pivotal for approving the aftereffects of the determination.

Keywords Telemedicine · Heart disease · Internet of Things · Machine learning · Health care

P. Kumar (✉) · S. Vinod Kumar · L. Priya
Rajalakshmi Engineering College, Chennai, India
e-mail: kumar@rajalakshmi.edu.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_14

1 Introduction

Human prosperity is characterized by the World Health Organization (WHO) as a condition of by and large physical, mental, and social success [1]. The typical life expectancy of people has expanded because of advances in clinical and logical exploration and expanded consciousness of individual cleanliness and prosperity [2]. The consistently rising clinical expenses and consistently developing populace are creating a colossal clinical benefit unevenness regarding geological imperatives, monetary status, and way of life. Because of these distinctions, there is an enormous disengage between the people who are needing clinical consideration and the individuals who will pay to support that consideration.

We presently approach homes that have been extraordinarily improved to make our lives more helpful thanks to the expansion in web-associated gadgets. The Internet of Things (IoT) is a term used to depict this gathering of web-associated objects [1]. The Internet of Things is developing rapidly nowadays, and its impacts on day-to-day existence and conduct are turning out to be increasingly articulated. An organization interfaces truly genuine things, like vehicles, structures, and web-associated contraptions, and it has actual science, programming, sensors, actuators, and organization properties worked in with the goal that the articles might assemble and share information. It has been utilized in many fields, including those connected with conveyance and transportation, weather conditions determining, and individual and social circles [3]. It is right now very predictable that by 2030, the IoT will comprise of around one trillion articles [4]. Past examinations have shown how habitually coronary illness strikes the oddball as a result of his exhausting way of life and unhealthful eating routine. Each odd person needs to have their wellbeing status looked at something like once a year to know how their body parts, especially their hearts, are doing. This second needs a gadget or system by which we can essentially screen the key heart boundaries that are expected to know the situation with heart working on account of older individuals who are separated from everyone else at home and need steady consideration. It has been found that there are as of now a few contraptions that register or screen pulse, ECG, and different boundaries. A glucometer can likewise be utilized to quantify irregular sugar. These are one of a kind machines that figure different boundaries. In this way, we require a contraption that can quantify the heart's important bodily functions at the same time, in one area, and that the patient and specialist can use in a crisis. The web of things is a worldwide innovation that is being utilized to give more prominent open doors and administrations to sickness survivors as the pervasiveness of persistent diseases ascends, as indicated by a 10,000 foot perspective of the medical services industry. Immediately, IOT innovation offers a superior and more effective approach to remotely screen patients, utilizing a sensor network that assembles information on the cloud as well as preprocesses it utilizing AI.

Because of various elements, the market for medical services remote observing frameworks has developed hugely. After some time, the number of inhabitants in

older individuals has developed to the point that it is currently moderately considered normal in industrialized countries for old individuals to reside freely in their own homes. Furthermore, the Internet of Things [5], which is the possibility of a world that can be checked and changed and utilizes sensors and actuators on both living and nonliving things, makes these medical services remote observing frameworks in fact and financially useful. The populace is expected to be prepared to embrace these kinds of arrangements that catch individuals' private and delicate information continuously, including temperature, blood glucose, heartbeat, and heartbeat oximetry sensors, to specify a couple, because of the inescapable utilization of brilliant portable innovation. For instance, medical services individual analyzers, for example, shrewd beds consequently report who is dozing in them. Considerably more, they can give an account of the physiological conditions of different patients, which empowers genuine brilliant home prescription allocators to, for instance, consequently ready when medicine isn't taken. There are a few medical services remote observing frameworks that utilization different innovations to follow or potentially screen patients, biomedical hardware, as well as both inside emergency clinics and at their homes. Tragically, to the extent that we know, most of these frameworks are not right now adaptable enough to add new sensors while they are running. Moreover, it hasn't permitted normal individuals to promptly assemble impromptu admonitions utilizing the new sensors.

2 Related Work

Utilizing an IOT design, Yeh et al. [6] presented a body sensor organization (BSN). For the development of two interchange components between brilliant items, the nearby handling unit, and the backend BSN server, they gave solid crypto-natives to ensure transmission privacy and give element validation. They have utilized the Raspberry Pi stage to show that the proposed approach is serviceable and plausible. They have set up testbeds for security parts in light of IOT. Here, a nearby handling unit or wise portable item reenactment of the Raspberry Pi stage is utilized (LPU). A coronary illness observing framework that can send patients' substantial pointers to far off clinical applications continuously was established by Li et al. [7]. Pulse sensor hub examination utilizing implanted frameworks and the help of fitting designing was accentuated by Fouad et al. [8]. It put an accentuation on wellbeing, home recovery, appraisal of treatment viability, and early problem disclosure.

A three-level IOT design with an AI calculation for coronary illness early discovery was accounted for by Kumar et al. [9]. To store and inspect the tremendous volumes of data made by wearable development, they have proposed three-level plans. Level 1 is stressed over party data from various sensors. Level 2 uses Apache HBase to store huge proportions of data in the cloud, and Tier III uses Apache Mahout to make assumption models considering key backslide. To give a nodal assessment of cardiovascular sickness, it performs ROC assessment last. To watch the genuine earnestness level and decide patients in understanding to have the truth, Kumar et al.

[10] made applications for adaptable clinical consideration that are cloud and IoT based. Wearable and embedded devices are examples of IoT contraptions. These contraptions are used to collect clinical information from distant districts. To quantify the continuous condition, rehabilitative data accumulated by IoT devices associated with human bodies can be utilized. By predicting everyone, which has through and through influenced diabetes, related clinical information is made using the UCI Repository dataset and the medicinal sensors. Applying five wonderful steps of an actually proposed accumulating approach, similar to information amassing, information recovery, information complete, information division, and information joining, can safely save the made information.

Moreover driving sensible assessments on wearable clinical development is Haghi et al. [11]. These PDA applications and wearable advancement have now been really gotten together with telemedicine and telehealth to develop the clinical snare of things. In scholastic articles and exceptional activities, the author fundamentally pushes wearable advancement. Long-stretch prosperity checking is wanted to be given at a lower cost.

Significantly more exploration was finished on wearable gadget innovation and its purposes in the mining area by Mardonova et al. [12]. A wearable wellbeing the executives framework for diggers is likewise proposed, alongside other expected applications. This can work on the wellbeing of mining tasks. The creator has accurately ordered wearable advances as indicated by their elements, capabilities, and applications, including wearable cameras, wellness trackers, savvy garments, brilliant watches, and shrewd eyewear. Ecological sensors, biosensors, position-and-area following sensors, and other normal sensors have all been distinguished.

Li [7] has made a coronary illness-checking framework that can possibly hand-off patients' substantial pointers to far off clinical applications continuously. Its two parts are information transmission and information catch. It was made in the wake of talking with clinical specialists. There are different actually taking a look at choices, for instance, blood lipids, ECG, SpO2, beat rate, and others. The three layers of the plan are the application layer, the vehicle layer, and the sensor layer. While actually looking at the patient's limits, taking into account the testing repeat is critical. The best degrees of patients have been described for area in Modes I to IV, which rely upon network quality interest. With the guide of Cogent Engineering, the creator underscored pulse sensor hub examination utilizing inserted frameworks. They proposed and executed a pulse sensor hub in an implanted telemedicine framework intended for the medical care framework. It put an accentuation on security, home recovery, evaluation of treatment adequacy, and early issue revelation. The creator has likewise proposed different ZigBee, Wi-Fi, Bluetooth, and RFID correspondence conventions for use in remote sensor network applications.

Utilizing Raspberry Pi, Jaiswal et al. [13] made an IoT-Cloud-based structure for gathering patient information in a shrewd medical care framework. The development of the Internet of Things can possibly save lives in the field of human administrations by get-together information from different gadgets, seeing patient information, and constantly diagnosing. As well as furnishing specialists and heads with admittance to an extensive variety of data sources, involving IoT innovation in clinical benefits

likewise presents hindrances in getting heterogeneous IoT data, especially in the versatile organization district of continuous IoT application structures.

The proposed approach by Umer et al. [20] investigates profound learning calculations for sorting cardiovascular breakdown patients as alive or dead. The system utilizes Internet of Things (IoT) sensors to gather flags and communicate them to a cloud web server for handling. Profound learning calculations further cycle these signs to discover the soundness of the patients. A clinical expert who will give crisis help on the off chance that vital approaches patients' wellbeing records and can handle results. The Heart Failure Clinical Records vault at UCI gave the dataset to this review, which has 13 qualities. The Internet of Things was utilized by Ancy et al. [21] to foster an edge-based heart sickness expectation gadget. Temperature, beat rate, and accelerometer sensors were joined with a Raspberry Pi to make the model. The deliberate worth is tried utilizing a prepared AI model to grasp the patient's condition.

Free old enough, a new review uncovers a sensational expansion in those with clinical issues. Then again, in such conditions, specialists likewise expect help to follow a persistent number of patients rapidly. There is consistently a requirement for brilliant gadgets and shrewd innovation to assist clinical experts and even people with better therapies of such ongoing sorts of sicknesses since there aren't generally clinical offices accessible in a crisis. The writing shows numerous techniques for wellbeing checking set forth by different writers. Nonetheless, it has been found that AI is essentially a technique by which people can help machines to do what we believe they should do. Thus, the customized machines acted such that assisted the framework with accomplishing its planned reason.

3 Proposed Work

The proposed framework might gather the fundamental information while barring commotion unsettling influences by at the same time checking a few boundaries consolidated onto wearable gadgets utilizing a solitary chip. It endeavors to keep its precision by limiting human cooperation. An ongoing symptomatic gadget involves biomedical sensors to gauge various boundaries in heart-weak patients who are somewhat found. Doctors can hold and view n number of accumulated perceptions sometime in the not too distant future for exact conclusion.

The preparation information will be sharpened and broken down utilizing an AI way to deal with work on heart capability. The data will be moved between the specialist and patient through two fundamental points of interaction. To save the patient's life, this structure perceives basic heart issues progressively and creates cautions by SMS, Email, and so on in view of high and low edge esteems that lay out an ideal for specialists as well concerning capable enlisted relatives. This will improve the framework by adding a feeling of care and consideration that esteems the patient's wellbeing.

This strategy intends to impel the patient-expert data affiliation. This advancement helps with cutting down the cost by clearing out the need to buy various gadgets at various sticker costs. As this is a one-time adventure, hospitalization costs will moreover be cut down consequently. By growing minimal expense, easy-to-utilize apparatuses for deciding and assessing heart capability, the proposed exertion plans to apply designing skills to social issues. To make and pick appropriate sensors, especially for cholesterol, heart maturing, and stress chemicals. To forestall any damage to human existence and to oversee coronary illness for poor individuals in a protected environment, it is important to make a calculation that will set off a caution in the event of chance. The focal point of the recommended work starts with the consideration and observation of the patient who demands the assistance of clinical experts during the conclusion interaction.

The expected methodology has been isolated into two particular stages: preparing and testing, as displayed in Fig. 1. Preparing will gather certifiable review data as well as fictitious data like that saw as on the web. Concerning used to remove data, for example, data preprocessing, data cleanup, data assortment, exception finding, and data control. Background data, which is required during time testing, was once held as a record of the total stage data. Anyplace we apply unobtrusive quantities of wearable sensors as medical services gadgets, essential plans foster the IoT-based medical care conspire climate. Then, subsequent to interfacing every sensor to the Raspberry Pi, we started assembling information from the sensors utilizing a liberal methodology. Each one accumulated has developed into a worldwide record thanks to affiliation situated plan. We get ready realities for testing while likewise reading up for each test. By utilizing various classifiers, one may likewise anticipate expected results. Give the assessment exactness with both genuine and counterfeit framework excitement toward the end.

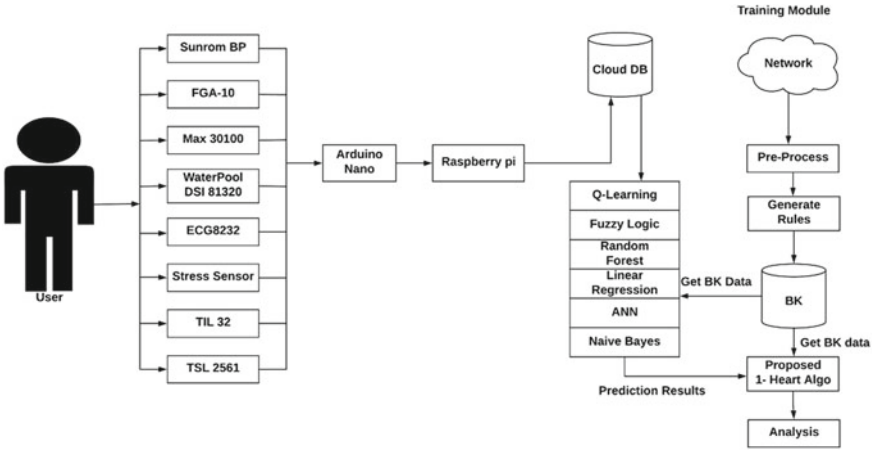


Fig. 1 Proposed system architecture

3.1 Proposed Modules Admin/Patient

Patients' data is required all through framework activity as well as authoritative information. The program will show the range of likely info values when the client dispatches it. During the main stage, the head fosters a couple of foundation strategies that are utilized to check the info streaming information delivered by the IoT climate. The background knowledge (BK) checks the ongoing occasion of the data set and works, when in doubt, establishment strategy.

3.2 Patient Profile

The patient modules give the manufactured information from profile data to the framework; it will initially make the patient profile with every essential boundary. For the stress level calculation, system asks a few questions to each profiler, these questions shown in Tables 1 and 2 describe the stress level with category for achieved score.

Table 1 Stress-level parameters

No.	Questions	Yes/ No
1	If the vehicle in front of me appears to be moving too slowly, I get upset or angry. I also get quite impatient when I have to wait in a line	
2	I have erratic moods, trouble making judgments, poor concentration, and poor recall	
3	Muscle aches and pains, especially in the neck, head, lower back, and shoulders, have increased	
4	My appetite has altered; I now either want to overeat or feel nauseous, and I may also skip meals	
5	My confidence and self-esteem aren't as high as I'd like them to be	
6	I lack motivation when I begin any new work. I don't feel well when I begin, and I don't want to complete any new tasks	
7	In any case, I'm not in a good mood	
8	Have trouble falling asleep, staying asleep, or both	
9	I can't make judgments quickly	
10	Fluctuations in blood pressure and heart rate	
	Total	

Table 2 Stress level

Stress level	0	1	2	3	4	5	6	7	8	9	10
Type	No	Slight			Reasonable		Severe		Very severe		Poorest

Patients answer each question with a yes or no according to the question table above. Each yes received a mark, and each no received a zero.

In the prosperity report for a specific patient, the strain regard is displayed according to the patient's continuous degree of tension. The strain values can move dependent upon the second and the incorporating conditions.

3.3 Vital Selection

A client or patient can pick the measurements that are fundamental in a given circumstance through a system called essential choice. The patient should choose the subtleties from the rundown of the multitude of boundaries. Pulse, pulse, cholesterol including LDL, HDL, and serum, stress pointer (mood), ECG including QT span and PR stretch, hemoglobin, oxygen immersion, and irregular sugar are among the indispensable signs that are given to the client.

3.4 I-Heart Calculation

The proposed technique offers an IoT forecast procedure in view of the machine learning (ML) system. The I-Heart calculation shows the qualities for the absolute body boundaries that still up in the air by a few AI methods. As indicated by the mix total rule forecast approach, the patient wellbeing report was produced utilizing the calculation underneath.

Input: Input values for all boundaries FullRecord < Double Value, String class> which contains all ascribed values like {BP, Heartrate, Cholesterol, Stress, Sugar, ECG, Oxygen immersion, Hemoglobin, CI} and so on. Strategy designs {$P1$, $P2$, Pn}

Output: Produce a sample report for a specific patient.

1. for every read Fullrecord, MinedElement[i][j] $\sum_{i=o,j=0}^{n}\left(a_{[i]}, a_{[j]} \ldots . a_{[n]}\right)$.
2. if MinedElement[j] similar to P[1], StandardLoc $= +1$.
   MainGrade1. Add $\leftarrow$ (StandardLoc).
3. if MinedElement[j] similar to P[2], TotLoc $= +1$.
   MainGrade2. Add $\leftarrow$ (TotLoc).
4. if MinedElement[j] similar to P[n] ComLoc $= +1$.
   MainGrade3. Add $\leftarrow$ (ComLoc).
5. for next.
6. Applying the formula below for the entire class list, compute the fitness factor for each class.

Table 3 CI grade calculation

No.	CI (L/min/m^2)	Gender	Class
1	If (CI < 4.2 OR CI > 2.5)	Both	Normal
2	If (CI < 2.4 OR CI > 4.3)		Below normal

$$f = \sum_{k=0}^{n} \frac{F(x)}{\text{Sum}F(x)}$$

$$\text{Load\_ PresentGrade } [w] = \frac{\text{MainGrade}(i)}{\text{Sumtest}} * 100$$

9. Arrange PresentGrade[w] in descending order
10. Mention PresentGrade[0] for last class for patient summary.
11. exit process.

3.5 Cardiac Index (CI)

In light of the patient's size, the cardiovascular record assesses the heart yield esteem displayed in Table 3. To find the cardiac index, divide the cardiac output by the person's body surface area (BSA). 2.5 to 4.2 L/min/m^2 is the typical reach for CI. The genuine model for the cardiovascular file (CI) and heart yield is displayed underneath (CO).

$$\text{Pulse} * 70) = \text{CO}$$
$$(\text{CO}/\text{BSA}) = \text{CI}.$$

A patient's CI would be 3.6 L/min/m^2 on the off chance that his cardiovascular result was 4.5 L/min and his BSA was 1.25 m^2. Another patient's CI would be 1.8 L/min/m^2 on the off chance that he had a cardiovascular result of 4.5 L/min yet a BSA of 2.5 m^2. Varieties might happen in both male and female sexual orientations.

3.6 Vascular Age Heart (VA)

Vascular Age Heart (VA) is the indicator suitable for patients with CVR factors is vascular age (VA), also known as heart age or CVR age. The age at which an individual of a similar sex as our patient would be assuming that individual had a similar outright gamble yet controlled risk factors is known as the VA of a patient with CVR factors. With this technique, the outright gamble is changed into an alternate thought that patients might grasp.

Various sensors are expected to catch the client's data to uncover the medicine's action and clinical side effects. As far as type, these sensors are too little to possibly

be worn. In view of the disclosures of hemoglobin and deoxyhemoglobin, a heartbeat oximeter sensor decides how much oxygen broke down in the client's blood. Such a sensor is valuable when the client's oxygenation is out of equilibrium and they need extra oxygen notwithstanding their escalated treatment. An electrocardiogram (ECG) sensor is utilized to get further cardiovascular information, including the pulse's and thump, any uncommon examples of invigorating action that might incline a patient toward sporadic heart thump conditions, and the area of the heart inside the chest area pit. Also, the ECG sensor shows signs of harm to different heart muscle tissues, including improved profundity of the heart muscle and essentially debilitated bloodstream to the heart muscle. The client's breathing rate is determined utilizing a nasal/oral wind current sensor to decide if the client needs respiratory help. A temperature sensor is utilized to decide the temperature of a specific body part. It might likewise be invigorating essentially to be put over the body part where temperature estimation is wanted. Alongside the over-exam sensors, which record the client's state of being, fall location and light sensors are likewise utilized. The light sensor gives the client the data to help the person in question control the encompassing light, permitting them to effectively explore around. For example, an estimation from this sensor could be utilized to turn on additional lights as the encompassing light level drops. An accelerometer is utilized by the fall revelation sensor to decide if a client has fallen out of the blue with the goal that an admonition sign can be shipped off raise the ideal concern.

As a rule, manual examination is utilized in most of medical clinics to assemble patient condition records. Persistent and standard patient observing is essential relying upon the patient's wellbeing. Outcomes incorporate manual pulse estimation; a long estimation time; low screen accuracy; trouble with programmed patient observing; a tedious method of interfacing various bits of gear; and trouble utilizing a thermometer to really look at a patient's internal heat level. The current body sensor organization, which offers completely mechanized and remote patient body checking, has been set up to dispose of every such restriction. Notwithstanding clinics, the medical services community can profit from the plan framework. The patient's body boundaries ought to be resolved utilizing the temperature, circulatory strain, and heartbeat rate sensors. The Raspberry Pi is independently connected to every sensor, which sends information about the body attributes to the thing talking. The client can see the body boundary information as a chart utilizing an internet browser on both a PC and a cell phone. At the point when BP and Pr ascend past the limit esteem, the signal will blare. It's attainable for the specialist to get mail containing patient body boundary data utilizing their discourse.

We expected numerous sensors to accumulate client information that addresses their action and clinical markers. For these sensors to be worn, they should be lightweight. In light of the estimation of hemoglobin and deoxyhemoglobin, a heartbeat oximeter sensor is utilized to compute how much oxygen broke down in the client's blood. Such a sensor is useful when the client's oxygenation is shaky, and they require extra oxygen or maybe escalated care. A few fundamental qualities are utilized for FGA10 and MAX30100. An electrocardiogram (ECG) sensor is utilized to assemble extra cardiovascular information, including the pulse's and beat, any

examples of unusual electrical action that could seriously jeopardize an individual for encountering strange heart cadence unsettling influences, and the place of the heart inside the chest pit. Furthermore, the ECG sensor gives indications of heart muscle injury in different regions, basically diminished blood supply to the heart muscle and thickening of the heart muscle. To assess whether a client needs respiratory help, a nasal/oral wind stream sensor is utilized to recognize the client's breathing rate. Any piece of the body can have its temperature estimated utilizing a temperature sensor, which is versatile and can be helpfully put over the area to be estimated. Light and fall discovery sensors are likewise utilized, notwithstanding the clinical sensor referenced over that screens the client's important bodily functions. The light sensor gives the client the data expected to change the surrounding lighting, making it simpler for them to move around. For example, data from this sensor can be utilized to turn on extra lights when the encompassing light level drops. To make advance notice signals and give the fundamental consideration, the fall discovery sensor, which is an accelerometer, is used to survey whether the client has fallen abruptly.

4 Performance of Proposed System

4.1 Cloud Database Administration and Online Service Deployment

The client data that has been assembled is shipped off a cloud server, which is responsible for permitting the openness of such data from any area over the Internet. Alongside executing a reasonable application program interface (API) and programming instruments that take into consideration admittance to and control of the information, the cloud server likewise carries out a wide scope of information the executives administrations, for example, information capacity, information investigation, and information representation. The center part of a cloud server is a sizable data set with sufficient space to hold huge volumes of information from a wide range of sensors over an extensive timeframe to follow framework client history. An enormous assortment of information examination instruments and APIs, including Google Sheets for information representation, is communicated with the data set. Both our planned Android applications utilizing information streams and dynamic website pages can be utilized to get information over the Internet.

The six existing AI calculations that are joined in the proposed I-Heart calculation are portrayed. To survey the general viability of the calculations used and displayed in Table 4, the accompanying boundaries were utilized:

Table 4 Testing parameters for the algorithm

S. No.	Properties/parameter
1	Data set
2	Size of dataset
3	Number of records for preparing and testing
4	Number of segments/attributes
5	Kind of dataset
6	Least threshold (lower)
7	Least threshold (upper)
8	Least selection rate for computing the accurately ordered results
9	Exactness of calculation
10	Mistake rate of algorithm
11	Running time of proposed calculation

4.2 Performance Analysis of Algorithm

The writing of a couple of current procedures has been utilized to break down calculations effectively while considering fleeting intricacy in contrast with the proposed approach. The exhibition evaluation of all calculations in a given climate is displayed in Table 5.

Table 5 Comparative analysis of system

Comparative factor	I-Heart (proposed)	NB	RF	ANN	Q-learning
Trustworthiness	Maximum	Moderate	Minimum	Moderate	Moderate
Safety	Moderate	Minimum	Moderate	Minimum	Moderate
Performance	Maximum	Maximum	Minimum	Moderate	Minimum
Portability	Indeed	No	Indeed	Indeed	Indeed (only a few conditions)
Configuration	Moderate	Minimum	Moderate	Minimum	Moderate
Compatibility	Profoundly compatible new working conditions	Moderate	Minimum	Minimum	Maximum

Table 6 Performance analysis

No.	Algorithm details	Accuracy in %	False ratio
1	Q-Learning [14]	93.7	7.32
2	Random forest [15]	94.2	3.99
3	Probabilistic fuzzy rule [16]	92.1	5.97
4	Naïve Bayes [17]	91.2	5.11
5	Linear regression [18]	93.21	9.03
6	Artificial neural network [19]	92.9	5.02
7	I-Heart algorithm (proposed)	98.11	1.98

4.3 Overall Performance Analysis

Table 6 outlines the general precision, everything being equal, including proposed I-Heart. It gives around 98.11% precision. The calculation Naïve Bayes gives the base precision than different calculations at 91.2%.

5 Conclusion and Future Scope

A brilliant gadget that tracks a few real vitals, for example, circulatory strain or sugar levels, is depicted ever. As per the situation, society should accomplish other things to safeguard individuals' lives and wellbeing in case of heart anomalies. Then, a moment reaction from a specialist, like a specialist, may assist with deflecting any troublesome results. This is a work made on the side of the people who live alone, are unattended, and need progressing oversight to make due. So the recommended I-Heart structure benefits society by offering a modest device to control the condition of coronary illness now. Furthermore, it makes a really inviting environment for them to handle the crisis issue effortlessly. Heart has been established for a wide friendly reason with the goal that those in need will actually want to get help with their earnest conditions. With a one-time venture, it considers long-lasting disease the board and treatment, cost decrease in medical problems, and constant remote observing. It likewise gives a mark of care to patients out of luck and makes a caution if there should be an occurrence of a crisis. Be that as it may, there is still opportunity to get better in the exploration on the grounds that the vast majority of the boundaries—including expansion file, blood vessel firmness, expansion tension, and others—are not considered while deciding the solidness of the heart corridors.

References

1. Čolaković A, Hadžialić M (2018) Internet of Things (IoT): a review of enabling technologies, challenges, and open research issues. Comput Netw 144:17–39. ISSN 1389-1286
2. Majumder S, Aghayi E, Noferesti M, Tehran HM, Mondal T, Pang Z, Deen MJ (2017) Smart home for elderly healthcare—recent advances and research challenges. Sensors (Basel) 11
3. Afzal B, Umair M, Shah GA, Ahmed E (2017) Enabling IoT platforms for social IoT applications: vision, feature mapping, and challenges. Future Gener Comput Syst. ISSN 0167-739X
4. Han SN, Crespi N (2017) Semantic service provisioning for smart objects: integrating IoT applications into the web. Future Gener Comput Syst 76:180–197. ISSN 0167-739X
5. Khutsoane O, Isong B, Abu-Mahfouz AM (2017) IoT devices and applications based on LoRa/LoRaWAN. In: IECON 2017—43rd annual conference of the IEEE Industrial Electronics Society, Beijing, pp 6107–6112
6. Yeh K-H (2016) A secure IOT-based healthcare system with body sensor network. IEEE Access
7. Li C, Hu X, Zhang L (2017) The IOT based heart disease monitoring system for pervasive healthcare service. Sci Direct Comput Sci 112
8. Fouad H, Farouk H (2017) Heart rate sensor node analysis for designing internet of things telemedicine embedded system. Cogent Eng 4:130–152
9. Kumar PM, Gandhi UD (2018) A novel three-tier internet of things architecture with machine learning algorithm for early detection of heart diseases. Comput Electr Eng, 222–235
10. Kumar PM, Lokesh S, Varatharajan R, Gokulnath C, Parthasarathy P (2018) Cloud and IoT based disease prediction and diagnosis system for healthcare using Fuzzy neural classifier. Future Gener Comput Syst
11. Haghi M, Thurow K, Stoll R (2017) Wearable devices in medical internet of things: scientific research and commercially available devices. Healthcare Informatics Research (HIR)
12. Mardonova M, Choi Y (2018) Review of wearable device technology and its applications to the mining industry. Energies 11
13. Jaiswal K, Sobhanayak S, Mohanta BK, Jena D (2017) IoT-cloud based framework for patient's data collection in smart healthcare system using raspberry-pi. In: 2017 international conference on electrical and computing technologies and applications (ICECTA), Ras Al Khaimah, pp 1–4
14. Chen G, Zhan Y, Chen Y, Xiao L, Wang Y, An N (2018) Reinforcement learning based power control for in-body sensors in WBANs against jamming. IEEE Access. 6:37403–37412
15. Anbarasi MS, Janani V (2017) Ensemble classifier with Random Forest algorithm to deal with imbalanced healthcare data. In: International conference on information communication and embedded systems (ICICES). IEEE, pp 1–7
16. Patil P, Mohsin S (2017) Fuzzy logic based health care system using wireless body area network. Int J Comput Appl, 80(12)
17. Kumar S et al (2018) Enhancement of health care using naive Bayes algorithm data mining of social media
18. Xu C, Chase JG, Rodgers GW (2015) Nonlinear regression based health monitoring of hysteretic structures under seismic excitation. Shock Vib
19. Woods JC, Walinjkar A (2017) ECG classification and prognostic approach towards personalized healthcare
20. Umer M, Sadiq S, Karamti H, Karamti W, Majeed R, Nappi M (2022) IoT based smart monitoring of patients' with acute heart failure. Sensors 2022. https://doi.org/10.3390/s22072431
21. Jenifer A, Jeba G, Paulraj L, Nithish Kumar K, Yuvaraj T, Alen G, Peter Rozario F (2022) Edge-based heart disease prediction device using internet of things. In: IEEE international conference on applied artificial intelligence and computing (ICAAIC)

Parallel Programming in the Hybrid Model on the HPC Clusters

Tomasz Rak

Abstract Support for an ever-expanding range of distributed and parallel strategies and support for performance analysis are constantly evolving. We propose message passing interface (MPI) and open multi-processing (OpenMP) standards that provide various software development tools covering a wide range of techniques not strictly limited to shared memory. The theoretical part focuses on a general overview of the current capabilities of MPI and OpenMP as application programming interfaces. The authors researched the high-performance computing (HPC) cluster environment, and this work provides results and representative examples. Also, in this study, it has been proven that to obtain favorable processing efficiency and write the code correctly—regardless of the programming paradigm used—it is necessary to know the basic concepts related to a given distributed and parallel architecture. Additionally, in this study, we investigated the need to adapt the tools used to solve a specific problem regarding improving processing efficiency. Finally, we presented that failure to apply a critical section may result in unpredictable results.

Keywords Performance engineering · High-performance computing · Cluster · MPI · OpenMP

1 Introduction

Parallel and distributed computing allow us to reduce the processing time of data. Dividing the program into separate parts makes it possible to process them on different machines or in different threads. The operation results always end in the main machine. Theoretically, the maximum acceleration is linear. However, it is not possible to achieve optimal acceleration. Amdahl's law [1] operates under the assumption that every large computing system consists of many separate parts. There are those

T. Rak (✉)
Department of Computer and Control Engineering, Rzeszow University of Technology, Powstancow Warszawy 12, 35-959 Rzeszow, Poland
e-mail: trak@kia.prz.edu.pl

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_15

that can be paralleled, and there are others that cannot be paralleled. Fragments mentioned last limit the achievable acceleration of the whole program. Additionally, thanks to the use of distributed computing, you could avoid any failure rate of the equipment. In this case, damage to one or more nodes does not stop the computation, but only extends the task processing time. High-performance computing (HPC) clusters are used for distributed computing [2, 3]. There are many different tools that could be used for cluster computing. Most recognizable being—the message passing interface (MPI) library and the open multi-processing (OpenMP) interface. These tools could be used together or separately, depending on the problem required for future solution. MPI and OpenMP could be used in well-known programming languages. The MPICH2 library was used, which implements the MPI protocol, and the GOMP library, which implements the OpenMP interface. MPI is the industry standard for distributed memory systems. OpenMP is the standard for parallel programming in shared memory systems. There are existing and discussed quantitative features for each of these solutions, and quantitative performance indicators are the best method to understand which tool to use for a specific application. As examples of problems for discussing these structures, the following were selected: matrix multiplication, shortest path search with the use of permutations, determination of π number with the Monte Carlo method, and a program for searching prime numbers. The results of simulations performed on a cluster of computers are presented. The tools used were analyzed, indicating which are appropriate for the group of tasks, and analyzing the characteristics and performance of the paradigms. We presented with the issues of performance engineering (PE), which covers techniques used in creating the architecture of the solution and the code. A distinction is made between system PE [4] and software PE [5].

1.1 Related Work

Parallel processing is not a new concept. This type of work is carried out in large computing centers [6]. Also, there are local data centers, usually with few resources. They serve the local needs of the unit in the field of research and are not available to external recipients. Today, no significant progress has been made in the building of efficient and optimal software for the use of contemporary architectures [7, 8]. Parallel computing is much more efficient and costs less. One of the possible solutions is the cluster with implemented, distributed, and parallel programming mechanisms [9]. Research work on the optimization of hybrid applications could be divided into three categories [10]. The first task is to improve existing standards and to determine how to implement the standards. The second one is responsible for most of the research on optimizing runtime environment functions to improve the performance of hybrid applications or improve handling in the context of MPI [11–13]. The third approach is to design resource managers and tools to facilitate and optimize the performance of hybrid applications. Complex problems often arise or problems that require processing large amounts of data. Therefore, on the one hand, effective

algorithms are being developed [14, 15]. On the other hand, as a result of continuous advances in parallel and distributed computing hardware and technology, it is possible to leverage the efficiency of computing resources to support them. The technology of HPC focuses on maximum use of the existing parallelism [16] using multi-core/multi-threaded processors and the use of network resources [17]. Various technologies have been developed to take advantage of hardware capabilities and successfully create distributed and parallel applications [18].

1.2 General Overview

This paper provides information of the performance about two parallel programming approaches: MPI and OpenMP. Studies have been conducted for several sets of problems in order to compare the two methods. Parallel programs were developed for each problem, and performance was also analyzed. The rest of the paper is structured as follows: Sect. 2 provides a brief overview of parallel computing models and architectures, and Sect. 3 provides a detailed overview of selected parallel programming frameworks. In Sect. 4, the parallel programs concerning the identified problems are discussed and the performance analysis is performed. Lastly, Sect. 5 provides a summary.

2 Parallel and Distributed HPC

Processing means that multiple threads or processes are executed concurrently if a single-core processor is available, otherwise they are executed simultaneously if a multi-core processor is available. Alternating execution happens when threads are switched in a quick succession, giving the impression that they are executed in a parallel way. There are two basic concepts associated with parallel and distributed computing: process and thread. Each time a computer program is launched, the operating system uses at least one process. The operating system assigns to each processor a unique process identifier (PID), memory, processing time, and access to input/output devices. Thread is part of a program that runs concurrently within a single process. The process can have multiple threads. Parallelism is a subset of concurrency, so a single-cored program, single-threaded processor, could be concurrent, but not parallel. The differences between process and thread are as follows: Threads are called by the system faster than the process and require less resources to run, and threads have a common address space in memory, which makes data exchange between them quicker, using pointers. Parallel calculations could be divided into the following:

– bits—they consist of increasing the length of the machine word, i.e., the basic unit of information processed by the processor in one cycle,

– instructions—it consists of grouping instructions and changing their execution order in such a way that they could be executed in parallels without changing the program result,
– data—it consists of iterative processing and focuses on the distribution of data between different computational units to minimize their interdependencies,
– tasks—it consists of assigning tasks between different processors, which shortens the program execution time.

Due to the equipment on which parallel calculations are performed, the following could be distinguished:

– single-processor multi-core computers—these are computers that contain one processor with many cores. These computers are commonly used in homes and offices.
– symmetric multiprocessor computers—these are computers that have many processors, which are the same, and each of them has many cores. Most of the time, such computers are used on servers or supercomputers. The servers use Intel Xeon or AMD Opteron processors. In high-speed, high-performance computing clusters (supercomputers), Intel Xeon, AMD Opteron, and high-speed hardware accelerators (GP/GPU) are predominantly used. Also, various server technologies are used (Intel, AMD, ARM, CPU, GPU), management systems data (disk, tape, flash memory), and network systems (Ethernet, low-latency Ethernet, Infiniband, Fiber Channel).

Parallel computations could be divided in terms of systems consisting of many machines into: clusters and grids. Appropriate hardware is not enough for parallel computing. Additionally, there is a need for appropriate algorithms. Parallel computing introduces additional issues, such as concurrent access to the resource and synchronization, which must be taken into account when designing the program.

2.1 Clusters

Cluster is a group of computers connected to each other for the purpose of performing a common task. The computers involved in the cluster are called nodes. The cluster has the following features: availability—node failure does not affect the operation of the entire cluster; scalability—it is possible to add another node at any time to obtain given computing power; performance—a cluster could consist of thousands of nodes; favorable price–performance ratio. There are three basic types of clusters: clusters of computers with high computing power (HPC)—the purpose of their operation is to increase the computing power. Computers perform subtasks that create one large task at the end; High-availability clusters—the purpose of these clusters is to ensure task continuity in case of node failure. Computers perform tasks that create one big task at the end; load balancing clusters—computers perform tasks from the available pool of tasks that have been sent to the cluster. One computer executes the entire task or may participate in the compounding subtask. Also, clusters could be divided based on

Table 1 Comparison of Ethernet, Myrinet, and InfiniBand

	Ethernet	InfiniBand	RoCE
Performance [Gbps]	100	100	100
Delay [μs]	5–50	<0.5	1–5

architecture: Single System Image (SSI)—the cluster is seen as a "single computer" by the user. Examples of such systems include: TruCluster, OpenSSI, MOSIX, and openMosix; Beowulf—Multiple Instruction, Multiple Data (MIMD)—this is a type of cluster that assumes obtaining the maximum computing power at the lowest cost. In this architecture, a cluster is a representative of HPC clusters. The cluster consists of particular elements: computing node—it is a computer that performs computing functions. This kind of computer may not have a hard disk, DVDROM, etc., which is not needed for the proper functioning of the operating system. The operating system could be loaded directly from the network or from ROM; managing node—this is the computer that starts tasks that are performed in the future; the node servicing the file system—this is a computer that has mass memories used by the other nodes; network—ensures communication between nodes. Typical networks such as Gigabit Ethernet or special networks such as Infiniband or RDMA over Converged Ethernet (RoCE) are used for this. In addition, RoCE offers the opportunity to use this technology in HPC systems by offering native support for both remote direct memory access (RDMA) computing applications and service applications. Table 1 compares Gigabit Ethernet, InfiniBand, and RoCE. Currently, there is a division into intra-cluster networks (e.g., OmniPath, Infiniband) and general-purpose networks (e.g., Ethernet) so it is necessary to build isolated network environments. The use of the RoCE protocol makes it possible to use applications using RDMA (most HPC applications) with an Ethernet network without the need for deep interference in the code of libraries or applications.

2.2 Grids

Grid technology is advanced enough to enable working infrastructure for scientific and commercial applications. The parallel programming paradigm for grids is an alternative to the well-known messaging paradigm in HPC clusters. However, implementing high-performance computing applications in grid infrastructures requires effective mapping, replication, scheduling, and commissioning, which could be an additional problem.

3 MPI Library and OpenMP Interface

On the control node, programs are launched, for example, by means of MPI, which distributes tasks to computing nodes. On each of the computational nodes, the task could be additionally divided into threads, e.g., using the OpenMP interface. Compute nodes could use a node that has disk space to read or write data. The computation that the cluster performs is distributed as tasks are allocated to different nodes. In addition, the computations the cluster performs are in parallel, since many instructions are executed at the same time. In addition, it is possible to transfer the calculations to the grid. There are pre-made MPI implementations for grids.

3.1 Message Passing Interface

MPI is a communication protocol between processes of parallel programs running on one or more computers. The first version of the protocol appeared in 1994. This standard is most often implemented in the form of libraries that could be used in various programming languages like Fortran, C, C++, Ada, Python, and Java. MPI is the dominant model currently used in clusters and supercomputers. There are three different versions: MPI 1, MPI 2, MPI 3. MPI 1 emphasizes messaging and has a static runtime environment. MPI 2 is better equipped with dynamic process management, memory operations management, and parallel I/O operations. MPI 3 introduces new collective operations that use non-blocking communication.

3.2 OpenMP

OpenMP is a cross-platform programming interface that allows to create parallel programs for multiprocessor systems with shared memory. OpenMP could be used in the programming languages C, C++, and Fortran. The interface consists of compiler directives, libraries, and environmental variables that affect the way the program runs. OpenMP implements multithreading where the main program is divided into several threads and performs a specific task with all those parts. Code fragments are tagged with appropriate preprocessor directives that branch the code into several new threads. Each thread has its own unique identifier. After finishing parallel processing, the threads return back to the main thread, which always has the identifier 0. Threads are allocated to the processor using special algorithms that take into account the load of individual processes of the entire machine. OpenMP is currently implemented in many commercial compilers.

4 Performance Study

The research was carried out on the performance cluster (HPC). The cluster consists of computing nodes connected to each other by a Gigabit Ethernet network (The cluster is located in the Department of Computer and Control Engineering of the Rzeszow University of Technology.). It runs on the Beowulf architecture with one master node. In addition to the operating system, software was also installed to prepare and run parallel and distributed applications. The following is the technical specification of the nodes: operating system: openSUSE 12.2 (Mantis x86_64); RAM: 1GB; CPU: Intel Core 2 CPU 64 bit, 2.13 GHz. A program has been written to perform the automated tests in the Python programming language, which automatically runs the specified number of times the given program in such a way that it starts the program, waits for its end, and then starts again (https://github.com/trak2023z/MPIcode). Each of the programs was written in the C++ programming language. Four applications were prepared for the study (The matrix multiplication program, the Monte Carlo determination of π number, and the prime lookup were written in such a way that it is easy to change the input data and test the effect of their size. The permutation shortest pathfinder program was written in a slightly different way than the others to not transfer too much data between processes. A matrix with the costs of connections between nodes has been permanently entered into the program code. The repository (https://github.com/trak2023z/MPIcode) contains the source code for all programs):

– matrix multiplication (P1),
– finding the shortest path using permutation (P2),
– determination of π number by the Monte Carlo method (P3),
– search for prime numbers (P4).

Each of the programs was compiled in two versions: MPI, and MPI from OpenMP. 100 repetitions were performed for the specified number of processes {2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 100, 150, 200} and {1, 2, 3} nodes for MPI alone and MPI with OpenMP. By default, OpenMP runs as many threads as the CPU has cores (2). Detailed results are presented for the first two applications. For the next two, the results of the analysis were discussed.

4.1 Matrix Multiplication Program

Matrix operations, presented in a field of mathematics called matrix algebra, are commonly used in a wide range of cases, including computer graphics. One of the basic activities among many others, often affecting their speed, is multiplication. The program was tested on matrices with 300×300 elements. The diagram (Fig. 1a) shows the minimum program execution time. As can be seen, the MPI's best time to execute is almost twice as good as the MPI from OpenMP. The MPIs with OpenMP

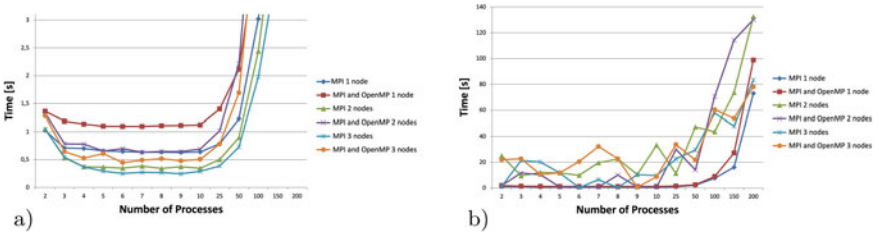


Fig. 1 Execution times of the matrix multiplication program (P1): **a** minimum, **b** maximum

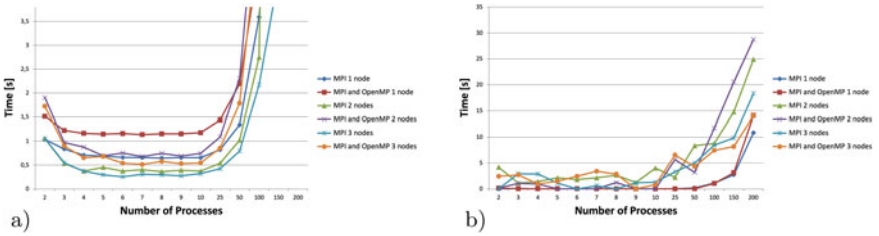


Fig. 2 Execution time for the matrix multiplication program (P1): **a** median, **b** standard deviation

were expected to have better results. The difference between the expected results and those obtained as a result of the test is due to the fact that the nodes in the cluster have only dual-core processors, which makes OpenMP not that effective. Synchronization between OpenMP threads resulted in increased computation time. Moreover, the computation time increases quite dramatically after exceeding 25 processes. This is due to the fact that the nodes have only one dual-core processor which must switch between processes in order to execute them in parallel. The graph (Fig. 1b) shows the maximum execution time of the program. It can be observed that the maximum execution times are many times greater than the minimum execution times. This happens because the cluster nodes used swap memory while the program was running.

The diagram (Fig. 2a) shows the median of program execution. The median is close to the values obtained in Fig. 1a. The graph (Fig. 2b) shows the standard deviation of the program execution time. The standard deviations are large in some cases. This may occur because the cluster nodes use swap memory. Also, the large standard deviation is influenced by delays in the transmission of messages related to a large number of messages. It can observe (the arithmetic mean) how MPI time on 3 nodes is approximately twice as fast as the time for 3 nodes using MPI and OpenMP.

4.2 Shortest Path Search Program with Permutation

The program was designed to find the shortest possible path between 10 points. The costs of connections between points have been presented using a matrix and have been permanently entered into the program to limit the amount of data transferred. The computational processes are given an initial permutation number and how many permutations to check. The permutation can be calculated knowing only the permutation index and what elements it consists of. It is known that the permutation of n elements is $n!$. The index passed to the algorithm takes the range $(0, n! - 1)$. All tables show the best time for each column, it has been bolded for more transparent view. Table 2 shows the minimum execution time of the program. As you can see, MPI together with OpenMP gives better results when fewer processes are running. As the number of processes running increases, MPI alone produces better results. Table 3 shows the maximum duration of program execution. The differences between the maximum and minimum times are not that big, which confirms the efficiency of the cluster. The program did not need such a large amount of memory to operate; therefore, the obtained results are similar. It can be seen that MPI and OpenMP always have better times where the number of processes is smaller. The difference between the median and the arithmetic mean is very small, which shows that the HPC cluster achieves the highest efficiency in this case. In some cases, standard deviations are large, which was caused by the use of swap memory during the operation of a specific case of the program.

Table 2 Minimum execution times for P2 program

Number of processes	MPI	MPI and OpenMP	MPI	MPI and OpenMP	MPI	MPI and OpenMP
	1 node	1 node	2 nodes	2 nodes	3 nodes	3 nodes
2	26.4476	22.7472	26.376	16.7828	26.4428	17.1245
3	18.7914	18.7109	13.2671	11.6007	13.2384	8.2419
4	17.7829	17.5746	9.4545	12.4316	8.875	8.2584
5	16.4546	16.9309	10.0503	9.4164	6.9449	9.2266
6	15.966	16.6181	10.5533	10.4606	7.6183	7.4251
7	15.4859	16.3859	8.8691	8.8622	6.5101	6.3152
8	15.3416	16.1473	9.2337	9.5807	6.0509	7.3654
9	14.9933	16.089	8.2811	8.5099	6.6342	6.5814
10	14.8345	15.998	8.9073	9.192	5.9183	5.9102
25	13.8875	15.495	7.299	7.9241	5.0377	5.4017
50	13.7873	**15.3953**	7.1239	7.9197	4.952	5.4502
100	13.8269	15.5632	6.9976	**7.9105**	**4.7475**	**5.3412**
150	**13.6046**	15.6275	**7.0498**	8.067	4.7645	5.4541
200	14.1202	16.3559	7.3483	8.4284	4.8542	5.5921

Table 3 Maximum execution times for P2 program

Number of processes	MPI	MPI and OpenMP	MPI	MPI and OpenMP	MPI	MPI and OpenMP
	1 node	1 node	2 nodes	2 nodes	3 nodes	3 nodes
2	27.8059	27.7805	26.8726	30.2247	35.4596	30.0826
3	26.2236	20.5182	13.4854	15.0807	13.7389	15.1038
4	18.266	18.3491	17.917	13.7096	9.2769	10.1046
5	17.8391	17.6732	13.4173	10.3809	13.4315	10.3121
6	16.1335	17.7954	10.8353	11.0585	10.8695	8.3371
7	16.3168	16.7009	9.2242	9.3075	9.0798	6.9155
8	15.624	16.5525	10.2406	10.1536	7.7937	13.0066
9	15.5481	16.4052	9.0637	8.8926	16.7568	17.646
10	15.0558	16.2958	9.1566	9.567	16.2937	16.8387
25	14.0705	15.6823	7.4671	8.0533	5.2715	5.5471
50	**13.935**	**15.5591**	7.3174	**8.0511**	5.154	5.5592
100	18.8326	19.0882	**7.2188**	8.1414	**4.9264**	**5.5222**
150	16.7958	17.4389	7.3776	8.3622	5.0088	5.7241
200	26.5847	38.2608	12.376	16.2739	5.1402	5.8769

4.3 Program for Determining π Number Using the Monte Carlo Method

The program was designed to determine the number π number using the Monte Carlo method. The program randomly picked 100,000,000 points and checked whether they belonged to the circle or not. To calculate the value of π number, all randomly selected points as well as those that belonged to the circle were selected. It can be seen that the minimum program execution times are much longer for MPI with OpenMP than with the implemented MPI library. The difference between the maximum and minimum times is not large. The program execution time decreases along with the number of nodes. The arithmetic mean and the median do not differ. It is similar for the standard deviation.

4.4 Program for Finding Prime Numbers

The program was designed to check whether or not numbers from 0 to 149,999 are prime numbers. It can be verified by the observation of minimum and maximum times, and the MPI with OpenMP gives better results than the MPI itself. The difference between the minimum and maximum time is even twice as large in some cases. The arithmetic mean is very close to the median. Standard deviations are not large in all cases.

4.5 Summary

Based on the analysis of the results of the programs' operation, several conclusions could be drawn. If the problem is "small" enough to be solved, and the computer resources such as cores and memory are sufficient, OpenMP is a good choice. When the data size is moderate and the problem is computationally intensive, MPI would be better. Also, MPI is a good choice when a program needs to run in parallel on a distributed structure with complex process coordination. OpenMP is the easiest to use because you only need to put a few directives in sequential code. MPI is relatively a bit more advanced to use when we can extract parts of the algorithm in the application that may be spread over many nodes.

5 Conclusions

In recent years, quantum computers have started to appear, but depending on the type of computation, they are either faster than traditional silicon computers or slower [19]. Therefore, the use of distributed and parallel methods continues to increase at all levels of processing, and with the development of accelerators such as GPUs, FPGAs, and other coprocessors, we are moving from desktops to clusters. MPI and OpenMP [17] are the most recognizable distributed and parallel programming solutions. Each of them is standard. MPI is a communication protocol between parallel program processes running on one or more computers. MPI is the dominant model currently used in clusters and supercomputers. The program must be specially prepared to use MPI. OpenMP is a cross-platform programming interface that allows you to create parallel programs for shared memory multiprocessor systems. OpenMP easily modifies a single-threaded program by using preprocessor directives so that the modified program will have fewer errors. OpenMP could be used together with MPI. To assess their performance in various configurations, several programs were implemented in the prepared server cluster. The performance of each of them was measured in terms of execution time. The aim was to adequately comprehend the usefulness of solutions in various situations. The presented are part of the performance engineering mechanisms. It is mainly about optimizing the performance of various types of applications and their verification. Most problems could be divided into n parts. The presented programs split the task into smaller parts and sent it to running processes. If the program was compiled with the OpenMP flag, then additionally, loops responsible for calculations in processes were paralleled in order to speed up the program. Note, however, that when applying multithreading, you need to apply a critical section to variables shared between threads to which you are writing. Failure to apply a critical section may result in unpredictable results. More complex programs that have a lot of read-only common data will give better results if we run fewer MPI processes with OpenMP because you will not have to transfer a lot of data over the network and there will be less memory consumption. The MPI solution from OpenMP only gave

better results for two programs: P2 and P4. For the other two programs (P1 and P3), MPI with OpenMP gave much worse results, even twice as long as the MPI itself. MPI from OpenMP works better for more advanced calculations or more complex algorithms. Parallelizing the *for* loop in OpenMP produces better results if the loop has fewer critical sections and fewer common variables to write data to. Parallel *for* loops, which have more lines of code, also produce better results. As the presented analyses show, there is a need to adapt the tools used to solve a specific problem with regard to improving processing efficiency. As part of the work, conclusions were presented that can be applied to other applications of this type on a local HPC cluster, as well as in relation to grid computing using various MPI implementations. Future research will focus on the MPI implementations for grids (MPICH-G2). MPICH-G2 allows us to connect multiple machines, potentially with different architectures, to run applications with the MPI library.

References

1. Amdahl GM (1967) Validity of the single processor approach to achieving large scale computing capabilities. In: AFIPS Spring joint computer conference, pp 483–485
2. González-Abad J, Lopez Garcia A, Kozlov V (2022) A container-based workflow for distributed training of deep learning algorithms in HPC clusters. Cluster Comput, 1–20. https://doi.org/10.1007/s10586-022-03798-7
3. Jianqi L, Hang L, Zhengyu T, Hua L (2020) Hybrid MPI and CUDA parallelization for CFD applications on multi-GPU HPC clusters. Sci Progr, 1–15. https://doi.org/10.1155/2020/8862123
4. Rak T (2017) Performance modeling using queueing Petri nets. In: Gaj P, Kwiecień A, Sawicki M (eds) Computer networks, CN 2017, Communications in computer and information science, vol 718. Springer, Cham. https://doi.org/10.1007/978-3-319-59767-6_26
5. Rak T (2015) Response time analysis of distributed web systems using QPNs. Math Probl Eng. https://doi.org/10.1155/2015/490835
6. Subodh K (2022) 2—Parallel programming models. https://doi.org/10.1017/9781009071314.003
7. Mustafa D (2022) A survey of performance tuning techniques and tools for parallel applications. IEEE Access 10:15036–15055. https://doi.org/10.1109/ACCESS.2022.3147846
8. Velarde Martínez A (2022) Parallelization of array method with hybrid programming: OpenMP and MPI. Appl Sci 12:7706. https://doi.org/10.3390/app12157706
9. Qiu Q, Lei Y, Wang D, Wang G (2021) An efficient hybrid MPI/OpenMP parallelization of the asynchronous ADMM algorithm. In: 2021 IEEE international conference on parallel and distributed processing with applications, big data and cloud computing, sustainable computing and communications, social computing and networking, pp 563–570. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00083
10. Abdullah A, Abdulraheem A, Fathy E (2021) Parallelization technique using hybrid programming model. Int J Adv Comput Sci Appl 12. https://doi.org/10.14569/IJACSA.2021.0120285
11. Li L (2022) Performance optimization of HPC applications in large-scale cluster systems. https://doi.org/10.1145/3489525.3511696
12. Saczek M, Wawrzak K, Tyliszczak A, Boguslawski A (2018) Hybrid MPI/OpenMP acceleration approach for high-order schemes for CFD. J Phys Conf Ser 1101:012031. https://doi.org/10.1088/1742-6596/1101/1/012031
13. Vallée GR, Bernholdt D (2018) Improving support of MPI+OpenMP applications. In: Proceedings of the EuroMPI conference. ACM, New York, NY, USA, Article 4, 2 p. 10.475/123_4

14. Sala K, Bellón J, Farré P, Teruel X, Perez JM, Peña AJ, Holmes D, Beltran V, Labarta J (2018) Improving the interoperability between MPI and task-based programming models. In: Proceedings of the 25th European MPI users' group meeting, Barcelona, Spain. https://doi.org/10.1145/3236367.3236382
15. Yang X, Chang X, Wang X, Li F, Ma J, Xin L, Chang H (2019) A new parallel scheduling algorithm based on MPI. In: 2018 15th international computer conference on wavelet active media technology and information processing, pp 228–231
16. Rak T, Schiffer Ł (2021) Own HPC cluster based on virtual operating system. In: Cognitive informatics and soft computing. Advances in intelligent systems and computing, vol 1317. Springer, Singapore. https://doi.org/10.1007/978-981-16-1056-1_37
17. Kwedlo W, Czochanski PJ (2019) A hybrid MPI/OpenMP parallelization of K-means algorithms accelerated using the triangle inequality. IEEE Access 7:42280–42297. https://doi.org/10.1109/ACCESS.2019.2907885
18. Gopalakrishnan A, Cabral MA, Erwin JP, Ganapathi RB (2019) Improved MPI multi-threaded performance using OFI scalable endpoints. In: 2019 IEEE symposium on high-performance interconnects, pp 36–39. https://doi.org/10.1109/HOTI.2019.00022
19. Hück A (2022) Compiler-aided type correctness of hybrid MPI-OpenMP applications. IT Professional 24(2):45–51. https://doi.org/10.1109/MITP.2021.3093949

An Extensive Study of Frequent Mining Algorithms for Colossal Patterns

T. Sreenivasula Reddy and R. Sathya

Abstract During the last decade of research, a lot of focus has been placed on the subject of frequent pattern mining (FPM). A profitable data set with a large sum of transactions and only a few items in each transaction has been used to develop numerous FPM algorithms. Because of the rise of bioinformatics, a new sort of data set called a high-dimensional data set has emerged, with fewer transactions but a greater sum of elements in each. The execution time of classical algorithms grows with deal length. High-dimensional data sets can't be processed by existing algorithms. But when applied to large data sets with a lot of dimensions, mining algorithms generate a huge amount of data, much of which is useless to scientists because of the little and medium-sized patterns they include. As a way to lessen the number of output patterns for mining patterns, colossal pattern mining is discussed. Since small and mid-sized patterns aren't mined, mining algorithms for enormous patterns run faster. In this work, an extensive study of colossal patterns, existing mining algorithms with its drawback is mentioned. The definitions of FPM, high utility mining and relation of colossal patterns with others are also explained. Pattern-Fusion is the first algorithm, which is developed for colossal patterns that is described briefly in this work.

Keywords Frequent pattern mining · Commercial data set · High-dimensional data set · Frequent colossal patterns · Data mining · Machine learning

T. S. Reddy (✉)
Research Scholar, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu 608002, India
e-mail: seenu4linux@gmail.com

R. Sathya
Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Annamalai Nagar, Chidambaram, TamilNadu 608002, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_16

1 Introduction

There are numerous applications for data mining, including the extraction of health computational biology, [1] detection of malicious online attacks, [2] web mining, [3] sentiment analysis and opinion mining in big data, [4] recommendation systems, [5] data warehousing, and [6] the use of data in decision-making. It is one of the most important practices for extracting from a data set that have occurred more than a user least threshold sum of transactions in the data set, and rule construction is the process of creating association rules using the patterns that have been extracted from the data set. A significant portion of research in association rule mining has been devoted to the FPM phase because of the enormous volume of computations, the lengthy execution time, and the enormous sum of memory it requires.

Various FPM algorithms in the literature have retrieved a variety of patterns. Patterns such as itemsets [7], sequences [8], and graphs [9] have been discovered from a variety of data sets. Novel approaches with humbler data constructions and more effective pruning algorithms have gradually superseded the initial motives of employing growth-based FP-based itemset mining methods due to their increasing complexity and vast number of created projected trees.

For efficient frequent itemset mining, several distributed tactics have been used [10]. Frequent itemset mining is an exponential problem, which means that mining methods produce a large number of extracted frequent itemsets as the outcome. The mining process may become less efficient as a result of the increased production. Research in the literature found that the best solutions to this problem were algorithms for closed pattern mining [11], maximum design mining [12], and colossal pattern mining [13]. The difficulty in mining frequent patterns is compounded by the fact that each pattern has an infinite number of sub-patterns, resulting in an enormous number of frequent patterns. Closed frequent pattern mining [14] and maximal frequent pattern mining (max-pattern) [15] have both been presented as solutions to this issue.

There are a limited number of common patterns in the whole pattern set, hence the term “closed pattern set” applies. A pattern is dubbed closed frequent if it occurs frequently in a data set but there is no super-pattern with the same support as it. There are fewer pattern super patterns than there are regular patterns in a database. In contrast to the more compact set of maximal frequent patterns, closed pattern mining tends to condense the set of frequently occurring patterns. As a result, maximally frequent patterns may not always include the full supporting material for their equivalent often occurring patterns.

The pattern mining challenge is space intensive for large data sets, even though closed pattern mining dramatically abridged the sum of processing and output capacity. Small and medium-sized patterns are frequently generated by mining algorithms, yet this data is often useless in many applications. Finding methods for mining that only extract huge patterns and ignore tiny and medium-sized patterns makes sense because many applications benefit from only large patterns. This difficulty can be approached in a new way by mining large patterns instead of little ones.

A bottom-up strategy to finding patterns is used by all previous techniques of pattern mining. These techniques begin with little designs and work their way up to larger ones. Small and medium-sized patterns, however, often lack useful information and can only be obtained from large-sized patterns, known as colossal patterns, in specific applications. It was in 2007 when the core Pattern-Fusion (core-fusion) technique, the first real approach for mining enormous patterns, was published [16].

1.1 Motivation

In the case of huge and very large data sets, the solutions to pattern mining problems [17] take a long time and are completely inefficient when trying to solve more complex problems. Many optimization and high-performance figuring practices have been industrialised to increase the performance of the pattern mining systems [18–21]. However, when working with large databases, these solutions are ineffective since only a small number of useful patterns are showed to the end user.

1.2 Problem Scope

In general, the sub-patterns that make up a colossal pattern are expected to appear at about the same frequency as the main pattern, therefore they can be identified by counting the number of supporters for each sub-pattern. There would be ncr number of common sub-patterns of size r for a colossal pattern of size n. As a result, in order to get to the massive patterns, we must first study an immense number of smaller patterns. In order to swiftly find large patterns, a strategy has been proposed to traverse the search area in jumps, ignoring most of the mid-sized patterns.

1.3 Structure of the Paper

It gives an introduction to FPM and talks about the problem of a huge pattern in Sect. 1. Pattern mining and colossal pattern relationships are explained in Sect. 2; this is the beginning of the text. Section 3 gives an impression of the algorithms for colossal patterns. Section 4 explains how to do colossal pattern mining from a list of frequently used items. It comes to an end in Sect. 5.

2 Pattern Mining Problems

A general definition of pattern mining is initially presented in this part, followed by an explanation of the connections between large-scale patterns and smaller ones.

Definition 1 (pattern) Reflect $I = 1, 2, \ldots, n$ as a set of items, and $T = t1, t2, \ldots, m$ as a set of transactions with the sum of transactions. We create the function *a*, which reads $p =$ for the item I in transaction t j. (i, j).

Definition 2 (pattern mining) A pattern mining task involves discovering all of the patterns L that are relevant to a given problem

$$L = \{p|\text{Interestingness}(T, I, p) \geq \gamma\} \tag{1}$$

To analyse a pattern p among a set and a set of items I, the measure of Interestingness(T, I, p) is used. Existing pattern mining issues can be summarised using these two definitions.

Definition 3 (frequent itemset mining (FIM)) As an postponement of the pattern mining issue, we create a FIM problem (Def. 2)

$$L = \{p|\text{Support}(T, I, p) \geq \gamma\} \tag{2}$$

As described in Def. 1, the collection of transactions in a Boolean database, T, I, and p are all defined as a Boolean database, and Support (T, I, p) is the sum of transactions in T covering the pattern _(T.I)/(|T|).

Definition 4 (weighted folder) A weighted database is defined by defining the function

$$\sigma(i, j) = \begin{cases} w_{\text{ij}} \; if \; i \in t_j \\ 0 \quad \text{otherwise} \end{cases} \tag{3}$$

W_{ij} refers to the weight of Item I, which is part of the transaction Tj.

Definition 5 (weighted itemset mining (WIM)) WIM problems are an expansion of the pattern mining problems (see Def. 2) by extending the

$$L = \{p|WS(T, I, p) \geq \gamma\} \tag{4}$$

It is possible to have WS (T, I) = WS (T, I) p ($j = 1$)(|T|)W (tj, Ip). W ($t\,j$, I, p) is the least weight of the elements of the pattern p in the transaction $t\,j$, and is a least weighted threshold for the weighted database established in Def. 3.

Definition 6 (uncertain database) Setting the function (see Def. 2) as is how an uncertain database is defined.

$$\sigma(i, j) = \begin{cases} \text{Prob}_{i,j} \text{ if } & i \in t_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Note that the transaction $t\,j$'s uncertainty value for I is Prob (i, j).

Definition 7 (uncertain itemset mining (UIM)) A UIM problem is defined as an postponement of the pattern mining problem (see Def. 2) by

$$L = \{p | US(T, I, p) \geq \gamma\} \tag{6}$$

With $\text{US}(T, I, p) = \sum_{j=1}^{|T|} \prod_{i \in p} \text{Prob}_{\text{ij}}$. In the indeterminate database specified by Def. 5.

Definition 8 (utility database) Utility databases are defined by setting the function (Def. 2)

$$\sigma(i, j) = \begin{cases} \text{iu}_{\text{ij}} \text{ if } i \in t_j \\ 0 \quad \text{otherwise} \end{cases} \tag{7}$$

It is important to keep in mind that eu denotes the item's external utility; this is the value of i's internal utility in the transaction (i).

Definition 9 (high utility itemset mining (HUIM)) The pattern mining problem (Def. 2) is an extension of this problem, which is called a "HUIM problem."

$$\begin{aligned} L &= \{p | U(T, I, p) \geq \gamma\}, \text{with } U(T, I, p) \\ &= \sum_{j=1}^{|T|} \sum_{i \in p} \text{iu}_{\text{ij}} \times \text{eu}(i) \end{aligned} \tag{8}$$

According to Definition 7, T is the set of transactions that can be found in the database defined by Def. 7.

Definition 10 (sequence database) Shoulder that there is a total order on things, such as $1 > 2 > 3 > n\ldots$ Sequences are an ordered list of $s = [I1, I2, \ldots, I\_(|s|)]$, where s is an itemet. The function (see Definition 2) is defined as $(i, j) = i$, if $I\ t\ j$ for each itemet Ii.

Definition 11 (sequential pattern mining (SPM)) Mining is extended to include the SPM problem (see definition 2) by:

$$L = \{p | \text{support}(T, I, p) \geq \gamma\} \tag{9}$$

If the sequence database's total transaction count (T) is less than the minimum support criterion (), then the answer is.

2.1 Relationship of Colossal Patterns to Other Patterns

Figure 1 shows how a colossal pattern develops over time. Minimum support should be 50. A and B are combined, and the frequency of AB lowers to 198 when AB is extended. As a result, pattern A appears 198 times, with pattern B appearing only once. Closed, but not massive, is the pattern AB. It is impossible to stop the pattern's growth at C, since the addition of D provides the same level of support as the original design. Closed but not enormous ABCD pattern. The smaller pattern will not be called a monstrous pattern until there is a considerable difference in frequency when it is extended. When G is added to ABCDF, the frequency drops from 193 to 190, and we no longer regard ABCDF to be a huge pattern, but rather ABCDFG. There is a considerable decrease in the frequency of ABCDFG when it is extended with Q, therefore this extension is not regarded as massive. (Seen in the figure marked with a cross.2) Because of this, ABCDFG and ABCDFGQ are classified as enormous patterns, respectively. As a result, a maximum pattern does not always have to be huge. Large-scale patterns are depicted in Fig. 1 ABCDEH and ABCDFG. The following observations can be made, based on the case above:

- There is no requirement that every closed frequent itemset is also a huge pattern, however this is not always the case.
- A gargantuan pattern does not always have to be an itemset with the highest frequency possible.

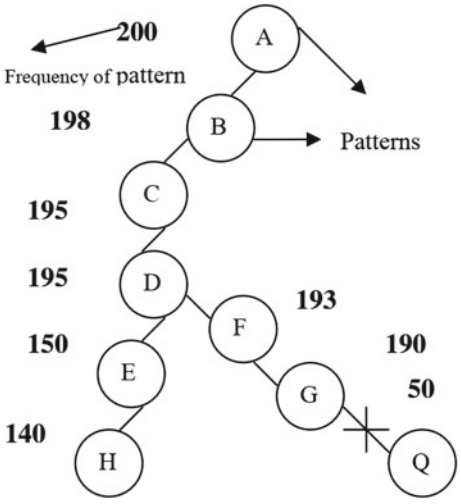


Fig. 1 Pattern tree growth process

3 Brief Explanation of Existing Techniques

3.1 Colossal Pattern Miner (CPM) Overview

Step-Wise Method

- There is an initial pool of 1 or 2 itemsets that we begin with (can be obtained using any frequent mining algorithm).
- We use any standard clustering algorithm to divide the patterns in the initial pool into groups based on their frequency. Frequency bands is what we call them.
- Choose a random seed pattern from the greatest frequency band and create a neighbourhood of -core seed patterns (depending on distance), startin–g with the present band and moving down through the other bands in declining order of their size.
- To remove a -core pattern () from its parent frequency band if its frequency is less than 1-, use the following procedure.
- Steps 3 and 4 should be repeated until all patterns have been picked or the desired number of neighbourhoods has been reached.
- To create one or more super patterns, we merge the patterns in each neighbourhood (colossal patterns).
- The new pool of super patterns is now the new initial pool, and the process repeats itself until we reach colossal patterns.

3.2 Bit-Wise Vertical Bottom-Up Colossal (BVBUC)

The enormous mining algorithm shown in Algorithm 1 is bit-wise vertical bottom-up. To begin with, it takes as input m, which is an index of the first row in the parent rowset, as well as the maximum number of rows in the bitmatrix, l, the level of processing a tree is now at, as well as S, the input rowset for the method.

This is a recursive algorithm with two major components. The algorithm initially determines if it has reached level minsup in the main if block. The method stops expanding the current branch if the level of three is the minimum level. This node's pattern and its support are written to the output file if the new gargantuan pattern is calculated and does not already exist in it. It is a basic function called pattern (S) that calculates the appropriate pattern of the rowset S by performing a "AND" operation on the bit vectors of the row-ids t–here are in S and selecting the items that have a

value of 1 in the result "AND" vector. The support of pattern (S) provided by function support is the sum of row ids in rowset S (S).

To begin, m is added to the S rowset, and then the else block builds the node's matching rowset. The algorithm will stop increasing this third branch if the pattern does not reach a user-specified threshold if it is not massive. Mining can continue only if the pattern (S) is large. This child node's rowsets will be generated by extending this node and creating its children for each row-id in the range of $m + 1$ to max if the branch of consistent child reaches the minsup level. If $p + m{-}a$ p is less than or equal to min sup, an algorithmic loop is in place. As a result of the closeness requirement method, we can now define a novel version of BVBUC that searches for closed, gigantic frequent patterns in data sets. BVBUC's closed version will be depicted in the algorithm's first component (the main if-block):

```
If(l = min sup)then
Begin
 If (Pattern(S)is colossal)then
  If(pattern(S)is not in file)then
   If(pattern(S)is closed)then
    Output(File, (Pattern(S), support(S)));
End
```

Only one condition has changed between the closed and primary versions of BVBUC, and that is the condition that tests for pattern proximity before adding it to output file. Since the algorithm has been altered, it can now mine closed, massively recurring patterns.

Algorithm 1. BVBUC Algorithm

```
Procedure BVBUC
Input
M: integer;// current, row id
Max: integer;// number of rows
1: integer;// current level (also number of row ids in this node)
S; string;// input rowset
Minsup; integer;// minimum support
Var
    i; integer;
Begin
   If(1 = minsup) then
Begin
  If (Pattern(S) is colossal) then
    If (pattern(s) is not in file) then
     Output (File, (Pattern(s), Support(S)));
End
Else
Begin
   S: = S + InToStr(m) + ";
   Begin
       For i:=m+1 to max do
         Begin
         BVBUC(I,max, 1+1,S, min sup);
         End
        End
      End
     End
End
```

3.3 LCCP: A Length Constraints with Algorithm for Mining Colossal Patterns

Here, two theorems are presented to establish the theoretical foundation of the suggested approach, which is cost effective in mining enormous patterns with length limits. Candidate patterns that do not meet the min-length restriction can be swiftly discarded using Theorem 1. For candidates with a max-length restriction, Theorem 2 has been found to be a time-saving tool.

Theorem 1 Nodes in the CP-tree that do not meet the min-length restriction are not allowed to have any children that do meet the constraint.

Theorem 1 states that if a node does not meet the least length requirement, it does not need to be expanded. As a result, mining operations can reduce the size of the exploration zone.

Theorem 2 Each child node in the CP-tree that meets the max-length criteria is considered to be a member of the parent node.

Rapid extraction of enormous patterns with length limits is made possible by the effective method LCCP (min-length and max-length). In order to produce and prune enormous pattern candidates, LCCP relies on PCP-Miner [22]. Using Theorems 1 and 2, it is possible to reject candidates who do not meet the min-length constraint and those who do not require the max-length condition to be tested.

4 From Frequent Itemset Mining to Colossal Itemset Mining

Itemsets that don't fulfil the minimum support are pruned from the search space. These prior row-enumeration algorithms, even using the support measure, failed to uncover many interesting closed patterns. Since the support-based strategy filters out patterns with low support but high confidence, this is the case. Support-based algorithms are discussed as alternatives in this part, which also examines approximately of the existing approaches.

4.1 Alternative User-Defined Threshold

MAXCONF [23] presents an algorithm for discovering interesting gene interactions from microarray data sets. In order to successfully narrow the search field, minsup is a measure of confidence rather than a requirement. For perturbation microarray data, which includes both common and rare (infrequent) correlations, this strategy was inspired by the view that support-based trimming is inappropriate. Multiple microarray data sets were used to evaluate MAXCONF and RERII [24], two improved versions of CARPENTER. For the purposes of categorising genes according to their molecular function, biological activity, and cell component, the extracted rules were subjected to the Gene Ontology's international standard for gene annotation (GO). In order for a rule to be considered biologically relevant, it needs be annotated with GO annotations. Support pruning is not ideal for mining gene expression data sets, according to this study, which found that MAXCONF rules are more biologically relevant than those identified by RERII. Though research has shown support-based algorithms remove many intriguing rules, the authors did not recognise the importance of pattern size. In association mining tasks, longer sequences are generally more essential than shorter sequences [25, 26].

4.2 Current Colossal Pattern Mining Process

First developed in an algorithm called Pattern-Fusion, the notion of colossal pattern was used to find an efficient approximation to gigantic patterns. In order to find the enormous pattern, Pattern-Fusion fused all of the smaller patterns into one, saving time and effort compared to manually traversing the pattern tree level by level. The Pattern-Fusion algorithm has been shown to be able to approximate large patterns in real data sets in several investigations.

Randomly picked sub-patterns are fused to form the enormous pattern, then the support is counted utilising individual database scans in Pattern-Fusion. However, Colossal Pattern Miner (CPM) [27] proposed a more intelligent technique to combine and separate the sub-patterns, which avoided the vast number of mid-sized patterns. To further reduce the number of database scans, vertical data format is used. Unfortunately, this technique has not been compared to Pattern-Fusion in terms of performance.

The data set is represented and compressed using a bit matrix by the BVBUC algorithm [28]. A bottom-up row list search tree is developed up to the minsup level since the greatest pattern of each branch is formed there. As a result, the time and memory requirements of most pattern mining algorithms are increased while searching for patterns with low minsup values. This is not the case in BVBUC, where when minsup drops, it also reduces the number of levels in the tree, removing branches that don't meet the minimum. However, as we'll see in a moment, this approach resulted in fewer purportedly mined itemsets. Another pruning strategy is to cease extending a node when the pattern has less elements than the minimum allowable quantity in a huge pattern. BVBUC beats both CPM and Pattern-Fusion in real data sets and microarray data sets, according to the authors.

DisClose [29] is an algorithm that first enumerates high cardinality itemsets and then constructs smaller itemsets to extract enormous closed itemsets. The row-enumeration tree is searched from the bottom-up like in BVBUC. Starting with a table that is transposed, the algorithm creates a compact row tree to hold the transposed table's itemsets. The CR-Tree is used during the search phase. Many rowset values are shared by a single node, making the suggested data structure compact and able to express itemsets with a minimum cardinality (mincard). Other methods employing a tree-based data structure, on the other hand, require a large amount of memory because of the length of the itemsets as well as the number of transactions that take place.

Colossal pattern mining using the -core ratio has been proposed using the newly presented algorithms CP-Miner and its upgraded counterpart, PCP-Miner [30], the pattern is a -core pattern. If there are no supersets in the database, then a -core pattern is considered gigantic. Instead of requiring the usage of a -core ratio, BVBUC proposes an acceptable minimum threshold and an acceptable minimum sum of itemsets in an itemset. However, in terms of runtime, they've outperformed BVBUC.

Using biological data sets, DPMine [13] proposes a new method for identifying enormous Colossal Pattern Sequences (CPS). In the DPT + tree, a vector

intersection operator is used to build enormous pattern sequences by identifying Doubleton Patterns. DPMine uses a new integrated data structure called "D-struct," which combines a doubleton data matrix with a one-dimensional array pair set to dynamically discover Doubleton Patterns from Biological data sets. DPT + trees are constructed using a bit-wise Top down Column enumeration tree. Constrained and predictable, D-main struct's memory can execute at a phenomenally high rate if memory is limited. It simply takes one scan of the database to detect enormous colossal pattern sequences thanks to the algorithm's construction. DPMine surpasses Colossal Pattern Miner (CPM) and BVBUC in a variety of biological data sets, according to an empirical investigation.

The decision-making process will be hampered because not all of the association's regulations will be generated. Most BVBUC enormous closed itemset support information is incorrect. This generates inaccurate association rules and has an impact on the decision-making process. Large cardinality itemsets, also known as enormous itemsets, are of particular interest to ARM because they may be used in applications using High-Dimensional Biological Data sets (HDBD) [31]. Colossal itemsets are significantly relevant and influential in many applications [32]. Naulaerts et al. [33] and Alves et al. [34] proved the importance of mining enormous item collections from high dimensional.

Since the time required to extract short and average-sized itemsets is exponential, [35] are inefficient for extracting FCCI from HDBD. The Pattern-Fusion approach was the first algorithm to come up with a large number of things [16]. In the Pattern-Fusion approach, the approximation of gigantic closed itemsets aids in the mining of big cardinality itemset. This bottleneck prevents the Pattern-Fusion approach from extracting a high number of FCCI. As a result of an insufficient collection of association rules, making decisions become more complicated. Even though HDBD contains a significant number of FCCI and frequently enormous itemsets, the BVBUC algorithm is unable to extract any significant number of these. Association rules influence decision-making in part because they are created. Even for the vast majority of mined FCCI, the BVBUC delivers incorrect support info. As a result, incorrect association rules are formed, which has a negative effect on decision-making.

The DisClose algorithm described by Zulkurnain et al. [36] mines FCCI from HDBD using a CompactRowtree (CR-tree). Before beginning the process of extracting FCCI from HDBD, the existing works do not have the ability to remove all unimportant characteristics and rows. As the number of rows enumerated in the mining search space grows exponentially, the algorithm becomes less efficient in mining FCCI. An efficient trimming methodology to minimise the row enumerated search space and an well-organised rowset testing mechanism are missing from current FCCI mining methods. The best way to extract the FCCI from HDBD is to traverse the row enumerated tree. There is an inherent imbalance in the row enumerated tree since the number of nodes in each row enumerated tree branch varies. Row enumerated trees must be distributed evenly across compute nodes in order to mine the FCCI efficiently. The compute nodes should be equally burdened when it comes to traversing.

Recently, a technique known as LCCP for mining massive patterns has been developed [25]. To begin, the issue of mining enormous patterns while enforcing length restrictions was raised. To swiftly determine if a huge pattern satisfies the length limitations, two new theorems were presented. Theorems based on these were used to develop a real algorithm for mining huge patterns with length limitations, eliminating those patterns that do not meet the length constraints in order to reduce mining durations. The min- and max-length limitations for mining gargantuan patterns were the focus of this paper. However, in order to mine the huge patterns, which necessitate the ideal selection of threshold values, it takes a long time to train the system.

According to this assessment, the greatest patterns may be found in the rowset, which has a support of 1. A criterion for mining large itemsets from high-dimensional data involving the item's support value is therefore unnecessary. As an alternative, a minimal permissible number of elements in a collection is known as the minimum cardinality of a collection.

5 Conclusion

To mine patterns in databases with a great number of characteristics and values, mining enormous patterns is utilised, although the number of occurrences in each database is limited. It is possible to extract gigantic patterns using efficient methods, but these methods cannot be applied to the case of constraint-based colossal pattern mining. It is the goal of this survey to investigate the extraction of enormous itemsets from biological data sets with high dimensionality. Massive mined itemsets of an average length do not include sufficient and useful info for making decisions. As a result, an enormous sum of time is spent mining a large number of short and medium-sized itemsets. The high-dimensional data set was created because of the increased interest in bioinformatics research and the abundance of data from a range of sources. These data sets have a great sum of features and a short number of rows. Bioinformatic applications, e.g. rely heavily on colossal pattern itemsets, which have a major impact on decision-making. From the enormous amount of information and knowledge that can be extracted, it is not an easy task. A high-dimensional data set's sequential and computationally expensive sequential mining algorithms include gigantic closed itemsets (CCI).

References

1. Xu J, Zhang Y, Zhang P, Mahmood A, Li Y, Khatoon S (2017) Data mining on ICU mortality prediction using early temporal data: a survey. Int J Inf Technol Decis Mak 16:117–159
2. Sohrabi MK, Karimi F (2018) Feature selection approach to detect spam in the facebook social network. Arabian J Sci 43:949–958

3. Kapusta J, Munk M, Drlik M (2018) Website structure improvement based on the combination of selected web structure and web usage mining methods. Int J Inf Technol Decis-Making. https://doi.org/10.1142/S0219622018500402
4. Hemmatian F, Sohrabi MK (2018) A survey on classification techniques for opinion mining and sentiment analysis. Artific Intell Rev. 10.1007/ s10462-017-9599-6
5. Sohrabi MK, Azgomi H (2017) Parallel set similarity join on big data based on localitysensitive hashing. Sci Comput Program 145:1–12
6. Liao S, Chang H (2016) A rough set-based association rule approach for a recommendation system for online consumers. Inf Process Manage 52:1142–1160
7. Sohrabi MK, Roshani R (2017) Frequent itemset mining using cellular learning automata. Comput Hum Behav 68:244–253
8. Huynh B, Vo B, Snasel V (2017) An efficient parallel method for mining frequent closed sequential patterns. IEEE Access 5:17392–17402
9. Cheng X, Su S, Xu S, Xiong L, Xiao K, Zhao M (2018) A two-phase algorithm for differentially private frequent subgraph mining. IEEE Trans Knowl Data Eng 30:1411–1425
10. Sohrabi MK, Taheri N (2018) A haoop-based parallel mining of frequent itemsets using N-Lists. J Chin Inst Eng 41:229–238
11. Rodríguez-González AY, Lezama F, Iglesias-Alvarez CA, Martínez-Trinidad JF, Carrasco-Ochoa JA, de Cote EM (2018) Closed frequent similar pattern mining: reducing the number of frequent similar patterns without information loss. Expert Syst Appl 96:271–283
12. Fasihy H, Shahraki MHN (2018) Incremental mining maximal frequent patterns from univariate uncertain data. Knowl-Based Syst 152:40–50
13. Prasanna K, Seetha M (2015) Efficient and accurate discovery of colossal pattern sequences from biological datasets: a doubleton pattern mining strategy (DPMine). Proc Comput Sci 54:412–421
14. Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Proceeding of the 7th international conference on database theory (ICDT'99). Israel, pp 398–416
15. Burdick D, Calimlim M, Gehrke J (2001) MAFIA: a maximal frequent itemset algorithm for transactional databases. In: Proceeding of the 2001 international conference on data engineering (ICDE'01). Heidelberg, Germany, pp 443–452
16. Zhu F, Yan X, Han J, Yu P, Cheng H (2007) Mining colossal frequent patterns by core pattern fusion. In: Proceeding of the 2007 Pacific-Asia conference on knowledge discovery and data mining
17. Fournier-Viger P, Lin JC-W, Vo B, Chi TT, Zhang J, Le HB (2017) A survey of itemset mining. Wiley Interdisc Rev: Data Mining Knowl Disc 4(7):e1207
18. Zhang L, Fu G, Cheng F, Qiu J, Su Y (2018) A multi-objective evolutionary approach for mining frequent and high utility itemsets. Appl Soft Comput 62:974–986
19. Djenouri Y, Comuzzi M (2017) Combiningapriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem. InfSci 420:1–15
20. Xun Y, Zhang J, Qin X, Zhao X (2017) FiDoop-DP: data partitioning in frequent itemset mining on hadoop clusters. IEEE Trans Parallel Distrib Syst 28(1):101–114
21. Djenouri Y, Lin JC-W, Nørvåg K, Ramampiaro H (2019) Highly efficient pattern mining based on transaction decomposition. In: IEEE international conference on data engineering, pp 1646–1649
22. Nguyen T, Vo B, Snásel V (2017) Efficient algorithms for mining colossal patterns in high dimensional databases. Knowl-Based Syst 122:75–89
23. McIntosh T, Chawla S (2007) High confidence rule mining for microarray analysis. IEEE/ ACM Trans ComputBiolBioinform 4:611–623
24. G. Cong, K.-L. Tan, A. Tung, and F. Pan, "Mining Frequent Closed Patterns in Microarray Data," Proc. Fourth IEEE Int"l Conf. Data Mining (ICDM), vol. 4, pp. 363–366, 2004.
25. Le T, Nguyen TL, Huynh B, Nguyen H, Hong TP, Snasel V (2021) Mining colossal patterns with length constraints. Appl Intell 51(12):8629–8640

26. Alves R, Rodriguez-Baena DS, Aguilar-Ruiz JS (2009) Gene association analysis: a survey of frequent pattern mining from gene expression data. Brief Bioinform 11(2):210–224
27. Madhavi D, Mogalla S (2010) An efficient approach to colossal pattern mining. IJCSNS Int J Comput Sci Netw Secur 10(1)
28. Sohrabi MK, Barforoush AA (2012) Efficient colossal pattern mining in high dimensional datasets. Knowl-Based Syst 33:41–52
29. Zulkurnain NG (2012) DisClose: discovering colossal closed itemsets from high dimensional datasets via a compact row-tree
30. Nguyen TL, Vo B, Snasel V (2017) Efficient algorithms for mining colossal patterns in high dimensional databases. Know-Based Syst 122(C):75–89
31. Sohrabi MK, Barforoush AA (2012) Efficient colossal pattern mining in high dimensional datasets. Knowl-Based Syst 33:41–52
32. Yoon Y, Lee GG (2012) Subcellular localization prediction through boostingassociation rules. IEEE/ACM Trans Comput Biol Bioinform 9(2):609–618
33. Naulaerts S, Meysman P, Bittremieux W, Vu TN, Berghe WV, Goethals B, Laukens K (2015) A primer to frequent itemset mining for bioinformatics. Brief Bioinform 16(2):216–231
34. Alves R, Rodriguez-Baena DS, Aguilar-Ruiz JS (2009) Gene association analysis: a survey of frequent pattern mining from gene expression data. Brief Bioinform bbp042
35. Zaki MJ, Hsiao C-J (2005) Efficient algorithms for mining closed itemsets andtheir lattice structure. IEEE Trans Knowl Data Eng 17(4):462–478
36. Zulkurnain NF, Haglin DJ, Keane JA (2012) Disclose: discovering colossal closeditemsets via a memory efficient compact row-tree. In: Emerging trends in knowledge discovery and data mining. Springer, pp 141–156

A Review Paper on Progressive Approach to Reduce Context Switching in Round Robin Scheduling Algorithm

Kuldeep Vayandade, Ritesh Pokarne, Mahalakshmi Phaldesai, Tanushri Bhuruk, Prachi Kumar, and Tanmay Patil

Abstract Processes/tasks are scheduled in order to finish the task on time. CPU Scheduling is a technique that permits one process to utilize the CPU while other is delayed (on standby) due to a lack of resources such as I/O allowing the CPU to be fully utilized. The goal of CPU scheduling is to improve the system's efficiency, speed, and fairness. When the CPU is not being used, the operating system chooses one of the processes in the queue to start. A temporary CPU scheduler performs the selecting process. The scheduler chooses one of the memory processes that are ready to run and assigns CPU to it. Every system software must have scheduling, and practically, all virtual machines are scheduled over before use. To enhance CPU efficiency, CPU utilization, delay, and CPU cycles is the primary objective of all presently available CPU scheduling techniques. There are various ways to tackle this, for example, algorithms like FCFS, SJN, priority scheduling, and many more, but in this paper, we chose to work with Round Robin (RR) Scheduling algorithm. RR algorithm solves the stated challenges as it's method's reasoning is significantly influenced by the length of the timeslot. In comparison, time slices should be huge than of the context switch time, as it enhances the performance by lowering the load on CPU. In this study, we review an existing technique to reduce context switching and break the fixed. With optimization, the range of the time quantum is utilized through the RR scheduling algorithm. This paper is researching mainly on the employment of context switching, and RR scheduling. Scheduling is discussing about how time slice of one process is completed, processor is allocated to next process, and saving of the state of process is needed because for the next time, it can run the process from the place it was halted. The review of this paper signifies the comparison of context switching based on different scheduling algorithm and the past work significance of study.

Keywords CPU scheduling · Context switching · Round Robin · Scheduling algorithms · Time quantum

K. Vayandade · R. Pokarne (✉) · M. Phaldesai · T. Bhuruk · P. Kumar · T. Patil
Vishwakarma Institute of Technology, Pune, India
e-mail: ritesh.pokarne20@vit.edu

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_17

1 Introduction

Various processes run constantly in a multi-programmed operating system to boost CPU usage [2]. Many of the processes are kept in memory at the same point in time. For the duration of single process, the operating system shifts the CPU from the CPU to an alternative process in its place. This [6] logic persists and repeats. Another process has the option to take over CPU usage whenever one process is forced to wait. Some operating systems use first come first serve (FCFS), shortest job first (SJF), priority, and Round Robin (RR) algorithms to complete this task more successfully [3]. These formulas make decisions and choose when and how long each process takes, taking into account sensitivity, response time, processing time, and latency.

The Round Robin scheduling method, which itself is regarded as the very commonly prevalent microprocessor scheduling algorithm, has significant problems with adding a supplementary matter size. The procedure features are constrained if the quantum's period is too great [5]. Context flipping will be increasingly prevalent because quantum is too tiny due to extra CPU overhead as a result. In order to increase CPU performance, the suggested technique would violate the current constant quant dimension of duration in this way.

Some sessions in an OS with several programs operate continually to use the CPU intensively [9]. Multiple processes are accumulated at the moment within the memory. As a few of the processes have to wait, operating system is getting away that CPU's process and delivers this one to an additional process [8]. The pattern gets repetitive. Whenever one process must wait, other processes can wait excessive CPU usage. Several virtual machines use the first come first serve (FCFS), shortest job first (SJF), priority, and Round Robin (RR) kind of scheduling processes to do this more quickly [1]. In terms of responsiveness, computation, congestion, reaction period, and other process parameters, these techniques decide when and how long each process operates. The RR scheduling method, which is thought to be the most frequently used CPU scheduling mechanism, has significant quantum size issues [4]. The subatomic element will reduce the application's nature if the duration is too long. Context switching is equal to enhancing the CPU workload when the time fraction is less [10].

Essentially, starvation-free execution of processes is not possible using conventional approaches, but the Round Robin algorithm allows you to predict the worst-case turnaround time for a given process provided you know how many processes are currently in the execution queue. Contrary to the Round Robin algorithm, which is considered to be a detriment in most systems, the first come first serve method (FCFS) approach shows to be non-pre-emptive. The shortest job first (SJF) approach also results in starvation or extraordinarily slow turnaround time. These elements highlight how Round Robin is superior to adjusting the intended average waiting time.

2 Related Works

"Comparative Analysis of CPU Scheduling Algorithms and Their Optimal Solutions" [1]. This paper refers to a set of guidelines and techniques for managing the sequence in which a computer system will carry out its tasks. The CPU is surely the highly vital aspect of a computer system. Because computations are increasingly distributed and parallel, recent advancements in software and system architecture have increased processing complexity. In this setting, job scheduling is challenging. The partition queue model, which a virtual machine (VM) may perhaps utilize, is a distinct virtual CPU (VCPU) operational queue for each physical CPU (PQM) [1]. The CPU scheduling mechanism known as sharing queue model (SQM) is an alternative. Using CloudSim, this study examines and assesses the effectiveness of various CPU scheduling methods in a cloud context.

"*Performance Evaluation of Dynamic Round Robin Algorithms for CPU Scheduling*" [2]. This paper includes four diverse RR methodologies: "Dynamic RR, CPU efficient time nanoscale RR, ideal RR using the Manhattan distance technique, as well as the improved Round Robin scheduling algorithm the average latency (AWT), estimated completion time, and turnaround time, which are the four systems of measuring that the authors use to estimate the effectiveness of the above classifiers. The outcomes appear that adaptive RR and optimal RR scheduling with Manhattan distance procedures function further logically when combined since they identified the operational elements with the lowest values.

"Improved Round Robin Scheduling using Dynamic Time Quantum" [3]. By adopting a unique weight modification for tasks which are stalled for I/O and incur some computational power, Tarek Helmy and Abdelkader Dekdouk tried to reconcile the minimal scheduling cost of Round Robin techniques and favor the quickest tasks in an effort to ensure proportionate balance. The outcomes of the study demonstrated that removing the fastest procedures quickly improves timeframe, long waits, and reaction time.

"*Response Time Analysis of Lazy Round Robin*" [4]. This study employs a Round Robin version that is naiver to construct and has subordinate runtime burden than standard RR. The turnaround time of the planner to recently released process occurrences is the main distinction among RR and Lazy RR. The Lazy RR administrator is believed to postpone responding to any task deployments until the existing round is complete, while the RR scheduler analyzes if a recently publicized assignment illustration may run for residual part of the current cycle.

"*Predicting academic success in higher education: Literature review and best practices*" [5]. It determines a figure that is neither overly big nor too little, each operation has a fair reaction time, and the system's efficiency is not reduced by needless context changes. To determine the appropriate temporal quantum, a novel active quantum employing the actual annual Round Robin (AN RR) is developed.

"*Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes*" [6]. This paper has proposed an algorithm called Self Editing Round Robin that can iteratively adjust the time of the quantum

to match the current time batch execution process. Experiments and computations are incorporated to solve fixed quantum time problems.

"*Finding time quantum of round robin CPU scheduling algorithm in general computing systems using integer programming*" [7]. This study suggests a variable time segment that specifies values that are neither absurdly high nor low, each process has a significant reaction time, and system throughput isn't really adversely affected by unnecessary memory management.

"*A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average*" [8]. This research proposes a novel strategy known as the dynamic microscopic point in time. The idea underlying scheduling is to alter the virtual machine time slice proportionate to the collective amount of time that certain operations have been queuing in the priority state.

"*Burst Round Robin as a Proportional-Share Scheduling Algorithm*" [9]. In this paper, the proportional scheduling algorithm is a way to merge a Round Robin scheduling algorithm with the shortest tasks with low overhead, using new weight adjustments for processes with I/O locked and wasting CPU time to ensure proportional fairness. Experimental results quickly demonstrated this by eliminating the shortest processes and achieving the best throughput, latency, and response times.

"A New Approach to CPU Scheduling Algorithm: Genetic Round Robin" [10]. In this paper, self-adjusting Round Robin is a technique used to permanently change time slices based on the timestamp of the currently active process. Fixed-time quantum problems are solved through experimentation and computation**.**

"An Improved Round Robin Scheduling Algorithm for CPU Scheduling" [11]. In this paper, in order to increase CPU usage, they proposed an integrating Round Robin using genetic algorithm. They coordinated the tasks corresponding to their minimum burst times and handed each one an ultimate time quantum. As an alternative of highlighting mean waiting time, this method focuses on standard response time. The maximum peak frequency of a process in the priority state, together with the median of the timestamp, is how the authors expected to determine the duration leap. The time constraint was explicitly determined once for each iteration of the program.

3 Comparison Table

Sr no.	Authors	Year	Title	Algorithms	Conclusion
1	Reddy et al. [1]	2019	Comparative analysis of CPU scheduling algorithms and their optimal solutions [1]	FCFS, SJF, RR	This paper makes reference to a set of guidelines and techniques for managing the sequence in which a computer system will get out its tasks. The CPU is a software system's most crucial component. Computational intricacy has grown due to recent advances in both software and design as calculations become more dispersed and simultaneous
2	Alsulami et al. [2]	2019	Performance evaluation of dynamic round robin algorithms for CPU scheduling [2]	Adaptive RR, Optimal RR, Manhattan	This paper includes four diverse RR methodologies: "Dynamic RR, CPU efficient time nanoscale RR, ideal RR using the Manhattan distance technique, as well as the improved Round Robin scheduling algorithm the average latency (AWT), estimated completion time, and turnaround time, which are the four systems of measuring that the authors use to estimate the effectiveness of the above classifiers
3	Debashree Nayak et al. [3]	2012	Improved round robin scheduling using dynamic time quantum [3]	Improved round robin	By adopting a unique weight modification for tasks which are stalled for I/O and incur some Computational power, Tarek Helmy and Abdelkader Dekdouk tried to reconcile the minimal scheduling cost of Round Robin techniques and favor the quickest tasks in an effort to ensure proportionate balance. The outcomes of the study demonstrated that removing the fastest procedures quickly improves timeframe, long waits, and reaction time

(continued)

(continued)

Sr no.	Authors	Year	Title	Algorithms	Conclusion
4	Tang et al. [4]	2021	Response time analysis of lazy round robin [4]	Response time analysis of lazy round robin	This study employs a Round Robin version that is naiver to construct and has subordinate runtime burden than standard RR. The turnaround time of the planner to recently released process occurrences is the main distinction among RR and Lazy RR. The Lazy RR administrator is believed to postpone responding to any task deployments until the existing round is complete, while the RR scheduler analyzes if a recently publicized assignment illustration may run for residual part of the current cycle
5	Alyhyan and Düştegör [5]	2020	Predicting academic success in higher education: Literature review and best practices [5]	A N Round Robin	This paper suggested a varying time quantum. It determines a figure that is neither overly big nor too little, each operation has a fair reaction time, and the system's efficiency is not reduced by needless context changes. In order to determine the appropriate temporal quantum, a novel active quantum employing the actual annual Round Robin (AN RR) is developed
6	Matarneh [6]	2009	Self-Adjustment time quantum in round robin algorithm depending on burst time of the now running processes [6]	Time quantum, robin round	This paper has proposed an algorithm called Self Editing Round Robin that can iteratively adjust the time of the quantum to match the current time batch execution process. Trials and computations are integrated to work out fixed quantum time challenges

(continued)

(continued)

Sr no.	Authors	Year	Title	Algorithms	Conclusion
7	Mostafa et al. [7]	2010	Finding time quantum of Round Robin CPU scheduling algorithm in general computing systems using integer programming [7]	Improved Round Robin, Integer programming	This study suggests a variable time segment that specifies values that are neither absurdly high nor low, each process has a significant reaction time, and system throughput isn't really adversely affected by unnecessary memory management
8	Abbas Noon et al. [8]	2011	A new round robin based scheduling algorithm for operating systems: dynamic quantum [8]	Dynamic quantum, new round robin	This research proposes a novel strategy referred to as dynamic microscopic point-in-time analysis. In order to adjust the virtual machine clock cycle proportional to the group session of chosen operations awaiting in the priority state
9	Maria Ulfah [9]	2012	A new approach to cpu scheduling algorithm: genetic round robin [9]	Genetic round robin, time quantum	In this paper, self-adjusting Round Robin is a technique used to permanently change time slices based on the timestamp of the currently active process. Fixed-time quantum problems are solved through experimentation and computation

3.1 Graph of Comparison

The comparative study reviews the operations with names X1, X2, X3, X4, and X5 to compare standing and response times after putting the reviewed model into practice.

(a) **Comparison for Waiting Time**

Table 1 shows waiting time before the optimization methods (WTBOM) and waiting time after the optimization methods (WTAOM). Graphical representation is shown in Figure 1. The graph below shows the significant rise and decline in latency for a given procedure after enabling this method.

(b) **Comparison of Turnaround Time**

Table 1 Comparison of waiting time

Process	WTBOM	WTAOM
X1	9	9
X2	1	2
X3	5	3
X4	3	5
X5	9	10

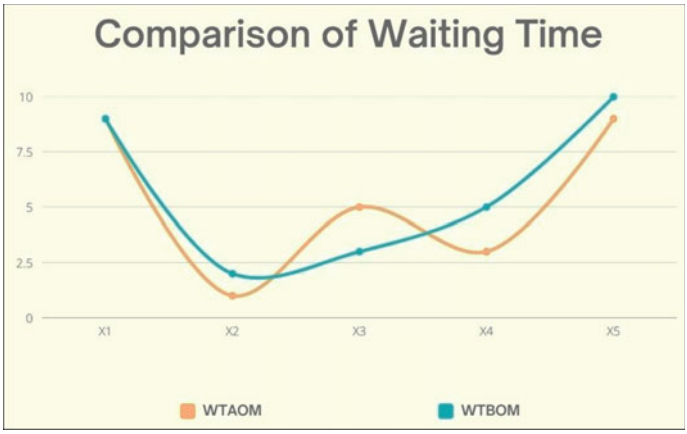


Fig. 1 Comparison of waiting time

The contrast outcome of waiting point in time of five processes is shown in Table 2—turnaround time before optimization methods (TATBOM) and turnaround time after optimization methods (TATAOT). Graphical representation is shown in Fig. 2. The graph below shows the significant increase and decrease in processing time for a given process after enabling this method.

(c) **Comparison of Existing Systems**

S1—Improved Round Robin Scheduling using Dynamic Time Quantum [3]
System using Round Robin with quantum time- 1ms
S2—A New Approach to CPU Scheduling Algorithm: Genetic Round

Table 2 Comparison of turnaround time

Process	TATBOM	TATAOM
X1	19	19
X2	2	3
X3	7	5
X4	4	6
X5	14	15

Fig. 2 Comparison of turnaround time

Robin [10]

System using Round Robin with quantum time- 2 ms

We have compared two existing systems based on their waiting time, turnaround time, and the number of context switches. From this comparison graph, we observe that system S2 which uses Round Robin with 2ms quantum size has the least number of context switches as shown in Figure 2. So from this, we can infer that in Round Robin algorithm, the context switching decreases when the quantum time is increased (Fig. 3).

Burst time is the time criteria for Round Robin algorithm. In the reviewed paper, the burst time for the process was X1=0, X2=1, X3=2, X4=1, X5=5, respectively, for five processes. Figure 4 shows the Gantt chart of System S1 which has Time Quantum 1 ms. Figure 5 shows the Gantt chart of system S2 which has time Quantum 2 ms. From both Gantt charts, we conclude that the context switching between two process is less in system s2.

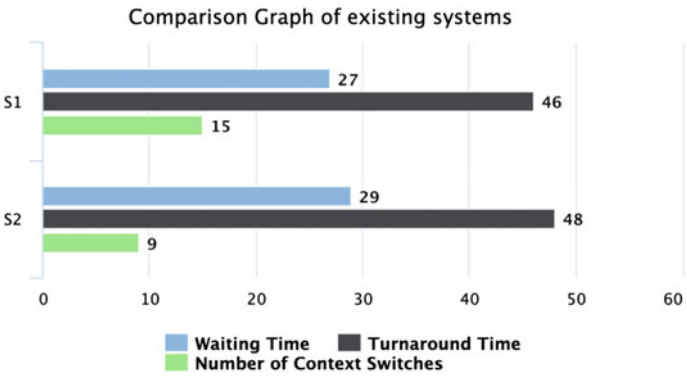


Fig. 3 Comparisons of existing systems

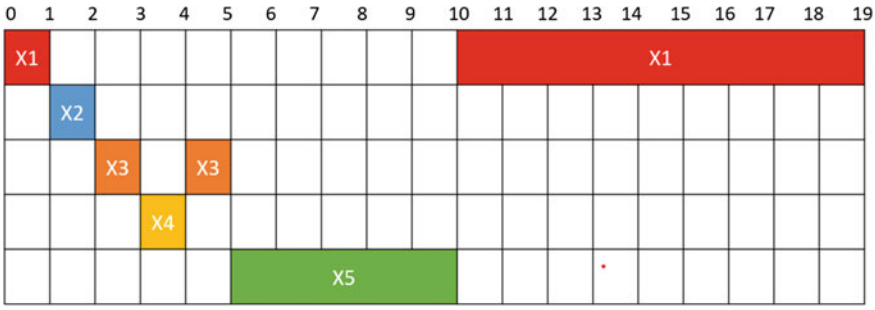


Fig. 4 Gantt chart for S1 System

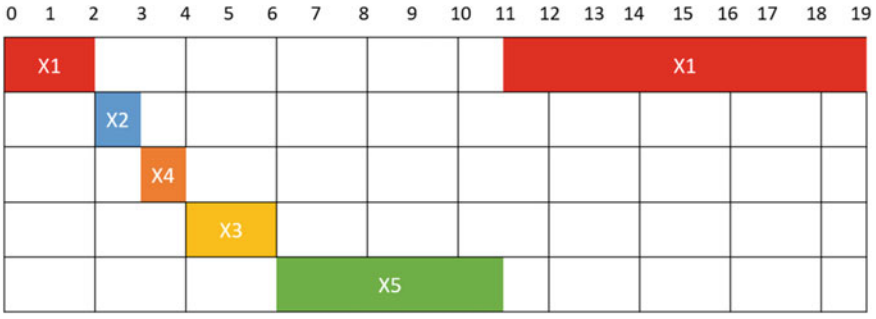


Fig. 5 Gantt chart for S2 System

4 Scope of Implementation

It was incorporated into this review study the context switching methodology and its future impact for the CPU scheduling algorithm Round Robin, which is an essential concept for implementing different scheduling algorithms. This review study shows that the technique not only allows the process to keep maintaining its current state so that it can restart processing later but also allows it to run faster processes and handle multiple process executions together without increasing the time limit. The study explains how Round Robin algorithm's enlargement is affected by the length of the clock cycle and the consequence of context switching among the numerous processes and that in the Round Robin traditional paradigm, the size of both the quantum is constant on all occasions [3]. If the quantum length has become too short, the context switch strengthens as a measure of each application's time consumed and latency has also been illustrated in this review study.

5 Limitations

The constraints of RR include longer waiting times, undesired overhead, and longer turnaround times for operations with varying CPU bursts due to the usage of static time quantum, among other things. All previous works based on Round Robin alter the method of taking time slices. However, each method had their own set of restrictions. When the time slice is too large, the processes in the ready queue are starved. The context switching time is long when it is very little. Time quantum can bypass these limits over time by changing time quantum in a gradual way at various states of the ready queue and minimizes context switching by a limited number. The algorithms can be improved in terms of context switching performance by additional research. Furthermore, by combining all of the scheduling algorithms, more adaptive algorithms that may be employed in any situation can be constructed.

6 Conclusion

This review study explains how context switching is an essential segment of OS. As devoid of context switching, there is no application of distinct scheduling set of rules. We also learn that, if concept of context switch is not implemented, then forcibly we must use FCFS scheduling. It has been discussed in this study how RR and additional scheduling algorithms are not feasible to execute not including switching. This research also stands by the fact that in FCFS, there is no need of context switching as once the process enters it gets executed. A better explanation of how large value of quantum RR will behave like FCFS has been shown in this study. So, we conclude from this research that to show context switching in better way, RR with small quantum value is preferred.

References

1. Reddy MR, Ganesh VVDSS, Lakshmi S, Sireesha Y (2019) Comparative analysis of CPU scheduling algorithms and their optimal solutions. In: 2019 3rd international conference on computing methodologies and communication (ICCMC). https://doi.org/10.1109/iccmc.20
2. Alsulami Q, Al-Haija A, Thanoon MI, Mao Q (2019) Performance evaluation of dynamic round robin algorithms for CPU scheduling. In: 2019 SoutheastCon, 1–5. https://doi.org/10.1109/SoutheastCon42311.2019.9020439
3. Nayak D, Malla SK, Debadarshini D (2012) Improved round robin scheduling using dynamic time quantum. Int J Comput Appl 38(5):0975–8887
4. Tang Y, Guan N, Feng Z, Jiang X Yi W (2021) Response time analysis of lazy round robin. In: 2021 design, automation and test in Europe conference and exhibition (DATE), pp 258–263. https://doi.org/10.23919/DATE51398.2021.9474242
5. Alyahyan E, Düştegör D (2020) Predicting academic success in higher education: Literature review and best practices. Int J Educ Technol Higher Educ 17(1)

6. Matarneh RJ (2009) Self-adjustment time quantum in round robin algorithm depending on burst time of the now running processes. Am J Appl Sci 6(10)
7. Mostafa SM, Rida SZ, Hamad SH (2010) Finding time quantum of round robin CPU scheduling algorithm in general computing systems using integer programming. IJRRAS
8. Noon A, Kalakech A, Kadry S (2011) A new round robin based scheduling algorithm for operating systems: dynamic quantum using the mean average. IJCSI Int J Comput Sci Issues 8(3), No. 1
9. Helmy T, Dekdouk A (2007) Burst round robin as a proportional-share scheduling algorithm. In: IEEE-GCC conference on towards techno- industrial innovations. Bahrain, 424–428
10. Siregar MU (2012) A new approach to cpu scheduling algorithm: genetic round robin. Int J Comput Appl 47(19)
11. Yadav RK, Mishra AK, Prakash N, Sharma H (2010) An Improved round robin scheduling algorithm for cpu scheduling. Int J Comput Sci Eng 02(04):1064–1066

Syn Flood DDoS Attack Detection with Different Multilayer Perceptron Optimization Techniques Using Uncorrelated Feature Subsets Selected by Different Correlation Methods

Nagaraju Devarakonda and Kishorebabu Dasari

Abstract Cyber attackers widely used Distributed Denial of Service (DDoS) attacks to saturate servers with network traffic, preventing authorized clients to access network resources and ensuing massive losses in all aspects of the organizations. With the use of ADAM, SGD, and LBFGS optimization techniques, this paper evaluates a Multilayer Perceptron (MLP) classification algorithm for Syn flood DDoS attack detection using various uncorrelated features chosen with Pearson, Spearman, and Kendall correlation methods. Dataset for a Syn flood DDoS attack was taken from the CIC-DDoS2019 dataset. Experiment results conclude that among optimization techniques, ADAM optimization gives better results and among uncorrelation feature sets and Pearson uncorrelated feature subset produce the best results. Multilayer Perceptron produces the best classification results with ADAM optimization and Pearson uncorrelation subset on Syn flood DDoS attack.

Keywords Syn flood DDoS attack · MLP classification algorithm · Correlation methods

1 Introduction

Confidentiality, Integrity, and Availability are primary principles of information security. Availability principle attacked by Distributed Denial of Service (DDoS) attacks [1] to intercept authorized clients using a system or application resources by shutting down the host. These attacks exploit a number of sources, some of which are

N. Devarakonda
School of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India

K. Dasari (✉)
Department of CSE(AI&ML), Keshav Memorial Institute of Technology, Hyderabad, Telangana 500029, India
e-mail: dasari2kishore@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_18

compromised or under their control, to generate the attack. The target system is overloaded with these requests, which lowers its performance and renders it unavailable to unauthorized users.

DDoS attacks are broadly classified into Volume-based, Protocol-based, and Application layer based attacks. Volume-based attacks are the classical type of DDoS attacks to create a huge amount of traffic to saturate the network bandwidth to deny the services to the legitimate clients. Protocol attacks exhaust the processing capabilities of network resources in order to target the Layer 3 and Layer 4 protocols of OSI ISO models with fake communication requests. Application attacks exploit weaknesses of Application layer or Layer 7 of the OSI ISO model in order to consume network resources.

Syn flood [2, 3] DDoS attack is protocol attack that employ the TCP connection's handshake phase to their advantage in order to completely consume a server's resources and prevent legitimate users from accessing it. TCP connection typically goes through three separate procedures before it is established. For establishing connection, first client sends a SYN request packet. Then server replies with a SYN/ACK packet to the client. Finally, client officially acknowledges ACK packet to the server. After sending and receiving this series of packets, the TCP connection is open and ready to send and receive data. A denial-of-service attack is launched by exploiting these sequences of handshaking activities. The attacker bombards the targeted server with many SYN messages by using spoof IP addresses. The server accepts requests for connections and watches open port to check response. Attacker continues to send SYN packets, while the server waits for ACK of the last packet that never arrives. A server is considered useless if all open ports are in use.

The DDoS attack and Syn flood DDoS attack are discussed in this section. Related work discussed in next section. The methodology of this research, including the pre-processing, correlation methods, Multilayer Perceptron classification algorithm, and its optimization strategies, is described in Sect. 3. In part 4 of this study, the experimental findings are used to explain the results and discussion. Section 5 of this paper contains the findings of this study and a feature enhancement.

2 Related Work

Various DDoS detection methods have been proposed by researchers. Saied et al. [4] proposed known and unknown DDoS attacks DDoS attacks detection method with MLP using specific characteristic features. Chang et al. [5] proposed DDoS attacks detection with MLP using Spark. Perakovic et al. [6] proposed DDoS attacks detection with artificial neural network. Xiao et al. [7] proposed DDoS attacks detection with KNN using correlation analysis. Dasari et al. [8] proposed exploitation-based DDoS attacks detection with machine learning algorithms using uncorrelated feature subsets selected by different correlation methods. Dasari et al. [9] proposed Syn flood DDoS attack detection with Support Vector Machine kernel functions using uncorrelated feature subsets selected by different correlation methods. Dimolianis et al. [10]

proposed Syn flood DDoS attacks detection and mitigation with machine learning classification models using programmable data plane filtering.

Researchers using different feature selection methods to detect DDoS attacks with machine learning models. This research using Pearson, Spearman, and Kendall correlation methods to select uncorrelated features for detect Syn flood DDoS attacks with multilayer perceptron classification algorithm.

3 Methodology

The Syn flood DDoS attack dataset was obtained from the CIC-DDoS2019 datasets and processed. Data preparation is the name given to a group of procedures used to get the data ready for machine learning algorithms. Remove features that differ from network to network first. After that, purge the data of any items with infinite or missing values. Using 0 and 1 to represent the benign and Syn attack target classes, respectively. To improve the efficiency of data classification algorithms, standardize feature values.

In order to speed up model training and data computation, feature selection [11] is a crucial stage in machine learning classification methods. Feature selection [12] is carried out in this study using filter-based feature selection [13] approaches such variance threshold and correlation methods [7]. Features that vary below a predetermined threshold are eliminated using the variance threshold. The variance threshold considers how the feature is related across all data collecting records. The association between the feature and the target label is disregarded. The relationship between features is described by correlation. The correlation coefficient values, which range from -1 to $+1$, show how closely related the traits are to one another. The features are highly uncorrelated when the coefficient value is zero. Uncorrelation features selected this study with Pearson, Spearman, and Kendall correlation approaches [2, 14].

The correlation coefficient calculated by Pearson is

$$r = \frac{\sum(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum(X_i - \overline{X})^2 \sum(Y_i - \overline{Y})^2}} \tag{1}$$

Here r indicates a coefficient of Pearson's correlation

X_i indicates X-feature sample value.
$\overline{X}$ indicates X-feature mean value.
Y_i indiates Y-feature sample value.
$\overline{Y}$ indicates Y-feature mean value.

Correlation coefficient calculated by Spearman is

$$\rho = 1 - \frac{6\sum d_i^2}{n\left(n^2 - 1\right)} \tag{2}$$

Here ρ indicates a coefficient of Spearman's rank correlation

d_i indicates difference value of two ranks of each observation.
n indicates total observations.

Correlation coefficient calculated by Kendall is

$$\tau = \frac{N_c - N_d}{\frac{n(n-1)}{2}} \tag{3}$$

Here τ indicates a coefficient of Kendall rank correlation

N_c indicates concords count.
N_d indicates discords count.

Multilayer perceptron (MLP) is a feed-forward neural network [15, 16]. The neurons are stacked in layers, with a predetermined number of identical neurons in each layer. All neurons of the each layer are connected to all adjacent layer's neurons. The first layer is called input layer, and its neurons gather information about the features of the dataset. The final layer is the output layer contains one neuron for every outcome value. The output layer contains single neuron, in the case of regression and binary classification, while output layer contains N-neurons in the case of N-class classification. All the layers in between input and output layers are called hidden layers because we don't know what these units should calculate ahead of time and have to figure it out while learning.

Optimization [17] is the process of minimizing or maximizing any mathematical expression. By minimizing the function used to change the attributes of the neural networks, optimization methods are used to solve optimization problems. This study evaluates the sgd, lbfgs, and adam optimization methods with a multilayer perceptron classification algorithm for DDoS attack detection.

Adaptive Moment Estimation (ADAM) optimization method is combination of momentum and Root Mean Squared Propagation (RMSP). This method is convergence rapidly and very fast. It rectifies the vanishing learning rate and high variance. It is a default optimization method for MLP classification algorithm. Computationally, it is more expensive.

An effective method of optimization for fitting linear classifiers under convex loss functions is Stochastic Gradient Descent (SGD). SGD is used to apply large scale machine learning problems. The advantages of SGD are efficiency and ease of implementation. SGD needs a certain amount of hyperparameters and iterations. Scaling of features affects it adversely.

Limited Memory Broyden Fletcher Goldfarb Shanno algorithm (LBFGS) is a memory-constrained optimization method of the quasi-Newton family of methods. It is more suitable for solving problems having large number of features. It does not

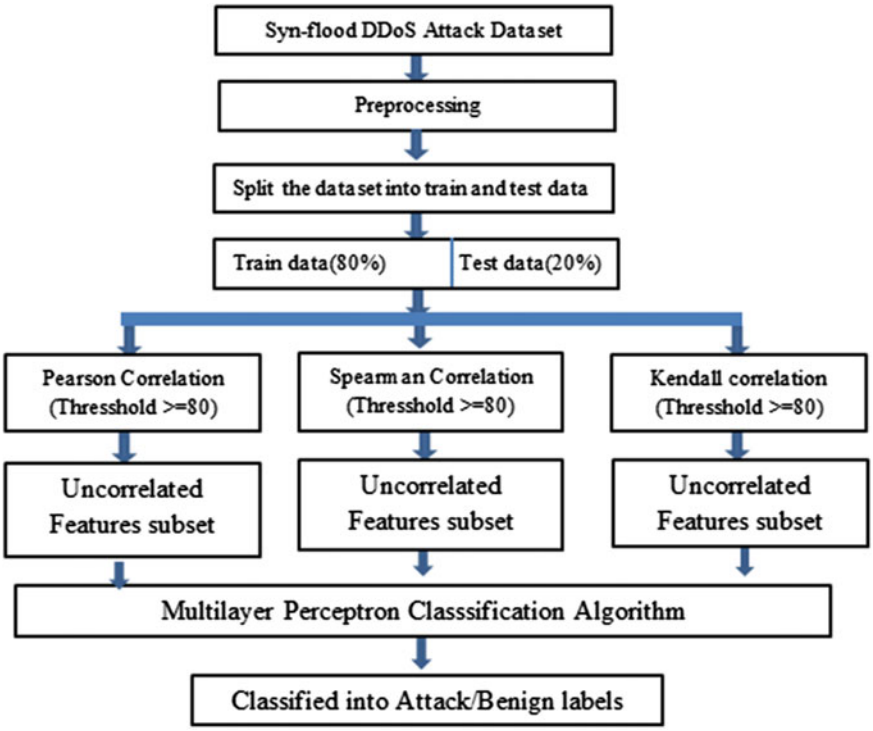


Fig. 1 Proposed framework for Syn flood DDoS attack detection with Pearson, spearman, and Kendall uncorrelated feature subsets with the MLP classification algorithm

require extensive hyperparameter tuning. Compare to SGD, it requires more memory and more iterations.

Carry out the experiments in this study with the Python programming language using sklearn, pandas, and numpy libraries for SVM classification algorithm processing and matplotlib and seaborn libraries for visualization of ROC-AUC curve on a Google Collab with 25 GB of RAM and a TPU environment. CICFlowMeter is network traffic flow generator tool used in this study to generate CSV files from extracted pcap files which are network traffic packet capture files (Fig. 1).

4 Results and Discussion

In this study, experiments are carried out using a dataset of Syn flood DDoS attacks. Apply the pre-processing on dataset to remove missing value records. Features which have variance threshold is 0 is called constant features. Quasi-constant features are those whose variance threshold is 0.01. Syn flood dataset contains a number of

constant and quasi-constant features are 12 and 7, respectively. Number of correlation features detected based on correlation threshold value > = 80 with Pearson, Spearman, and Kendall correlation methods are 38, 44, and 46, respectively. Select uncorrelation feature subsets of correlation methods by removing correlation feature subsets of the feature set of Syn flood dataset. 22, 16, and 19 numbers of uncorrelated features identified by Pearson, Spearman, and Kendall correlation methods, respectively, in Syn flood dataset (Table 1).

This study evaluates Syn flood DDoS attack detection with three proposed uncorrelated feature subsets using accuracy, specificity, log-loss, K-fold cross-validation, and ROC-AUC score valuation metrics.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4}$$

Table 1 List of un-correlated feature subsets of Syn DDoS attack

Pearson uncorrelated feature subset	Spearman uncorrelated feature subset	Kendall uncorrelated feature subset
Protocol	Protocol	Protocol
Active mean	Active mean	Flow duration
Flow duration	Flow duration	Total fwd packets
Active min	Active min	Down/Up ratio
Total backward packets	Total fwd packets	Total backward packets
Flow packets/s	Flow packets/s	Init_Win_bytes_forward
Total length of fwd packets	Total backward packets	Total length of fwd packets
Fwd packet length max	Down/Up ratio	Min_seg_size_forward
Flow IAT mean	Total length of fwd packets	Fwd packet length std
Fwd packet length min	Fwd packet length std	Idle min
Flow IAT min	Bwd packet length std	Bwd packet length std
Bwd packet length max	Flow IAT min	Active min
Fwd IAT min	Bwd IAT std	Flow packets/s
Bwd packet length min	Init_Win_bytes_forward	Bwd IAT min
Down/Up ratio	Idle min	Flow IAT std
Bwd IAT total	Min_seg_size_forward	Bwd IAT std
Init_Win_bytes_forward		Flow IAT max
Bwd IAT min		Flow IAT min
Act_data_pkt_fwd		Active mean
Bwd packets/s		
Init_Win_bytes_backward		
Min_seg_size_forward		

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

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

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

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

Here, TP indicates TRUE POSITIVE, FP indicates FALSE POSITIVE, TN indicates TRUE NEGATIVE, and FN indicates FALSE NEGATIVE.

$$\text{Log} - \text{loss} = -\frac{1}{N}\sum_{i=1}^{N}[Y_i \ln P_i + (1 - Y_i)\ln(1 - P_i)] \tag{9}$$

Here, 'P' indicates prediction probability, 'N' indicates total observations, and 'Y' indicates actual value.

K-fold cross (KFC)-validation dataset is divided into K folds, onefold is chosen as a test, the other folds are used for training, and finally the model is assessed. This procedure is repeated until a test fold has already been chosen from each fold.

To assess the effectiveness of classification models by mapping True Positive Rate with the False Positive Rate at various threshold levels is called Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

Accuracy results of Syn flood DDoS attack Detection utilizing MLP classification algorithm with various optimization methods using various uncorrelation feature subsets are given in Table 2. MLP with ADAM optimization produces best accuracy with Pearson uncorrelation feature subset. Among three MLP three optimization methods, ADM produces better accuracy with all uncorrelated feature subsets. Among uncorrelated feature subsets, Pearson feature subset produces better accuracy values with all MLP optimization techniques.

Table 3 gives the accuracy results of various machine learning models with Pearson, Spearman, and Kendall uncorrelated feature subsets. Gradient Boost and Support Vector Machine algorithms give better accuracy values than other machine learning models. Proposed MLP with ADAM optimization using Pearson uncorrelated feature subset gives better value than Gradient Boost and Support Vector

Table 2 Overall model accuracy of the MLP classification algorithm on Syn flood attack

Optimization techniques	Pearson	Spearman	Kendall
ADAM	99.96	99.92	99.92
SGD	99.94	99.88	99.87
LBFGS	99.94	99.87	99.93

Table 3 Overall model accuracy of other ML classification algorithms on Syn flood attack

Classification algorithms	Pearson	Spearman	Kendall
Logistic regression	99.94	84.28	84.50
Decision tree	99.89	99.86	99.88
Random forest	99.94	99.87	99.83
Naïve Bayes	99.44	99.60	99.61
Gradient boost	99.95	99.95	99.89
Support vector machine	99.94	99.94	99.94

Machine algorithms. MLP with ADAM optimization method using Spearman uncorrelated subset gives nearly equal results compared to Gradient Boost and Support Vector Machine algorithms results using Spearman uncorrelated feature subset. MLP with ADAM optimization method using Kendall uncorrelated subset gives nearly equal results compared Support Vector Machine algorithms results using Spearman uncorrelated feature subset.

Table 4 gives the MLP with LBFGS optimization with a Pearson uncorrelation feature subset is produced the best K-fold cross-validation accuracy. Among optimization methods, ADAM optimization and among uncorrelated feature subsets, Pearson feature subset with MLP produces best K-fold cross-validation accuracy values than others.

Table 5 gives the specificity results of the Syn flood DDoS attack detection with MLP classification algorithm with different optimization techniques using different uncorrelation feature subsets. MLP with ADAM optimization produces best specificity value with Pearson uncorrelation feature subset. Among three MLP three optimization methods, ADM produces better specificity results with all uncorrelated feature subsets. Among uncorrelated feature subsets, Pearson feature subset produces better specificity results with all MLP optimization techniques.

Table 4 MLP classification algorithm's accuracy scores for K-fold cross-validation on a Syn flood attack

Optimization techniques	Pearson	Spearman	Kendall
ADAM	99.9527 (0.0140)	99.9086 (0.0147)	99.9054 (0.0149)
SGD	99.9415 (0.0097)	99.8982 (0.0164)	99.8966 (0.0167)
LBFGS	99.9615 (0.0155)	99.9030 (0.0147)	99.9078 (0.0105)

Table 5 Specificity values of the MLP classification algorithm on Syn flood attack

Optimization techniques	Pearson	Spearman	Kendall
ADAM	0.92	0.82	0.86
SGD	0.85	0.59	0.62
LBFGS	0.87	0.80	0.87

Table 6 gives the log-loss values of the Syn flood DDoS attack detection with MLP classification algorithm with different optimization techniques using different uncorrelation feature subsets. MLP with ADAM optimization produces best log-loss value with Pearson uncorrelation feature subset. With all uncorrelated feature subsets, ADM generates superior log-loss values than the other three MLP optimization techniques. With all MLP optimizations, the Pearson feature subset outperforms the other uncorrelated feature subsets in terms of log-loss values.

Table 6 Log-loss value of the MLP classification algorithm on Syn flood attack

Optimization techniques	Pearson	Spearman	Kendall
ADAM	0.01550976	0.02880388	0.02880380
SGD	0.02215686	0.04099041	0.04320601
LBFGS	0.01994115	0.04542134	0.02548029

Table 7 gives the ROC-AUC scores of the Syn flood DDoS attack detection with MLP classification algorithm with different optimization techniques using different uncorrelation feature subsets. MLP with SGD optimization produces best ROC-AUC score value with Kendall uncorrelation feature subset. The MLP ADAM optimization method produces better ROC-AUC scores with Pearson and Spearman uncorrelation feature subsets, while SGD and LBFGS optimization methods produce better ROC-AUC score values with Kendall uncorrelated feature subset. The ROC-AUC curves of MLP utilizing three optimization techniques—Pearson, Spearman, and Kendall correlation—are shown in Figs. 2, 3 and 4, which are used to detect Syn flood DDoS attacks.

Table 7 ROC-AUC values of the MLP classification algorithm on Syn flood attack

Optimization techniques	Pearson	Spearman	Kendall
ADAM	0.97169062	0.98552292	0.94947445
SGD	0.98452295	0.97891490	0.99409244
LBFGS	0.97787982	0.95259236	0.97765161

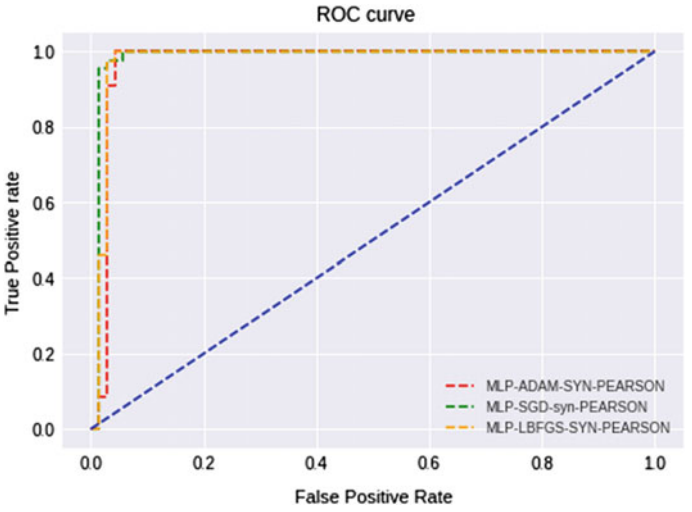


Fig. 2 ROC curves of the MLP algorithm on the Syn attack using the Pearson uncorrelated feature subset

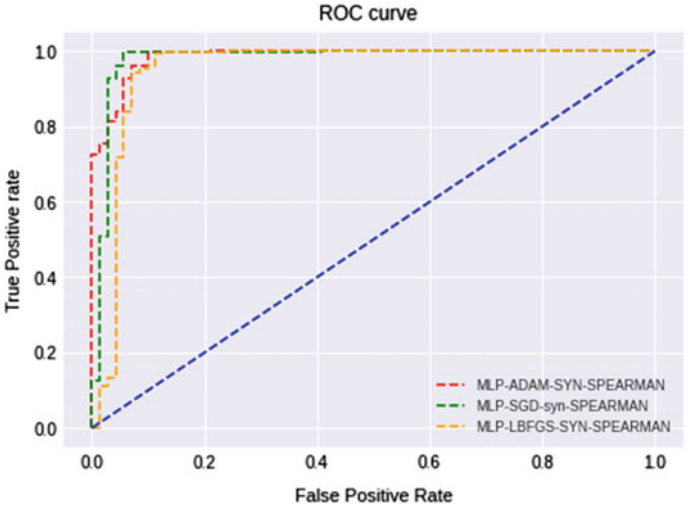


Fig. 3 ROC curves of the MLP algorithm on the Syn attack using the Spearman uncorrelated feature subset

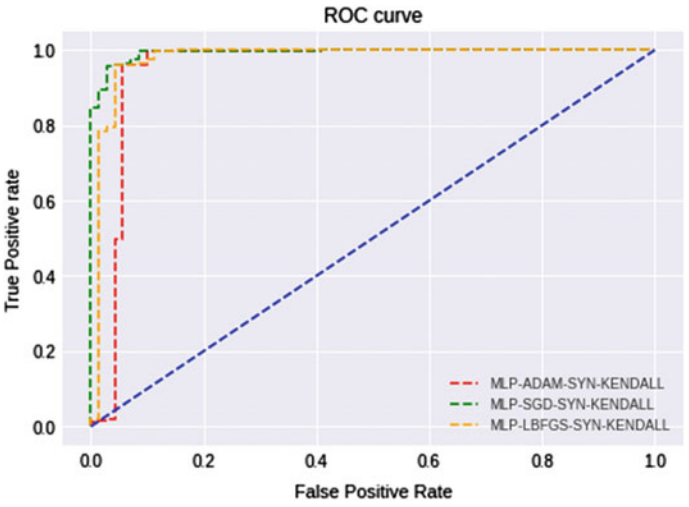


Fig. 4 ROC curves of the MLP algorithm on the Syn attack using the Kendall uncorrelated feature subset

5 Conclusion

This study evaluated classification of Syn flood DDoS attack and benign label classes with Multilayer Perceptron classifier with ADAM, SGD, and LBFGS optimization methods. This study proposed using a feature selection method for classifying that select a common uncorrelated feature subset by Pearson, Spearman, and Kendall correlation methods. MLP classifier with ADAM optimization method gives better results on the Syn DDoS attack dataset. MLP classifier with Pearson uncorrelation feature subset gives better results. MLP classifier gives best results with the ADAM optimization method using Pearson uncorrelation feature subset of the Syn DDoS attack dataset. These proposed experimental results are compared with other machine learning models. MLP classifier with ADAM optimization method using Pearson uncorrelated feature subset gives the best results than other machine learning models. The extent of research on DDoS attack detection using MLP classification algorithm with features selected by KPCA [8] dimensionality reduction.

References

1. Dasari KB, Dr Devarakonda N (2018) Distributed denial of service attacks, tools and defense mechanisms. Int J Pure Appl Math 120(6):3423–3437. https://acadpubl.eu/hub/2018-120-6/3/247.pdf
2. Dasari KB, Devarakonda N (2021) Detection of different DDoS attacks using machine learning classification algorithms. Ingénierie des Systèmes d'Information 26(5):461–468. https://doi.org/10.18280/isi.260505

3. Tuan TA, Long HV, Son LH, Kumar R, Priyadarshini I, Son NTK (2020) Performance evaluation of botnet DDoS attack detection using machine learning. Evol Intell 13:283–294
4. Saied A, Overill RE, Radzik T (2016) Detection of known and unknown DDoS attacks using artificial neural networks. Neurocomputing 172:385–393. ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2015.04.101
5. Chang-Jung H, Ting-Yuan C (2016) Detection DDoS attacks based on neural-network using Apache spark2016 international conference on applied system innovation (ICASI), pp 1–4. https://doi.org/10.1109/ICASI.2016.7539833
6. Peraković D, Periša M, Cvitić I, Husnjak S (2016) Artificial neuron network implementation in detection and classification of DDoS traffic. In: 2016 24th telecommunications forum (TELFOR), pp 1–4. https://doi.org/10.1109/TELFOR.2016.7818791
7. Xiao P et al (2015) Detecting DDoS attacks against data center with correlation analysis. Comput Commun 67:66–74
8. Dasari KB, Devarakonda N (2022) TCP/UDP-based exploitation DDoS attack detection using AI classification algorithms with common uncorrelated feature subset selected by Pearson, Spearman and Kendall correlation methods. Revue d'Intelligence Artificielle 36(1):61–71. https://doi.org/10.18280/ria.360107
9. Dasari KB, Devarakonda N (2022) SynFlood DDoS attack detection with SVM kernels using uncorrelated feature subsets selected by Pearson, spearman and Kendall correlation methods. In: 2022 second international conference on computer science, engineering and applications (ICCSEA), pp 1–6. https://doi.org/10.1109/ICCSEA54677.2022.9936114
10. Dimolianis M, Pavlidis A, Maglaris V (2021) SYN flood attack detection and mitigation using machine learning traffic classification and programmable data plane filtering. In: 2021 24th conference on innovation in clouds, internet and networks and workshops (ICIN), pp 126–133. https://doi.org/10.1109/ICIN51074.2021.9385540
11. Al-Naymat G, Al-Kasassbeh M, Al-Harwari E (2018) Using machine learning methods for detecting network anomalies within SNMP-MIB dataset. Int J Wireless Mobile Comput 15(1):67–76
12. Mekala S, Rani BP (2020) Article ID: IJARET_11_11_121 Kernel PCA based dimensionality reduction techniques for preprocessing of Telugu text documents for cluster analysis. Int J Adv Res Eng Technol 11(11):1337–1352. https://doi.org/10.34218/IJARET.11.11.2020.121
13. Chen L et al (2018) Detection of DNS DDoS attacks with random forest algorithm on spark. Procedia Comput Sci 134:310–315
14. Dasari KB, Devarakonda N (2022). Detection of TCP-based DDoS attacks with SVM classification with different kernel functions using common uncorrelated feature subsets. Int J Saf Secur Eng 12(2):239–249. https://doi.org/10.18280/ijsse.120213
15. Wang M, Lu Y, Qin J (2020) A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput Secur 88:101645, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2019.101645
16. Ramkumar BN, Subbulakshmi T (2021) Tcp syn flood attack detection and prevention system using adaptive thresholding method. ITM Web Conf 37:01016. https://doi.org/10.1051/itmconf/20213701016
17. Kushwah GS, Ali ST (2017) Detecting DDoS attacks in cloud computing using ANN and black hole optimization. In: 2017 2nd international conference on telecommunication and networks (TEL-NET), pp 1–5. https://doi.org/10.1109/TEL-NET.2017.8343555

Ensemble Model Detection of COVID-19 from Chest X-Ray Images

Lavanya Bagadi, B. Srinivas, D. Raja Ramesh, and P. Suryaprasad

Abstract The rapid advancement of deep learning techniques usage in treating several medical issues and search for innovative methods in predicting COVID-19 is the leading cause for this finding. Feature-level ensemble model is proposed to distinguish COVID-19 cases from other similar lung infections. Three different pre-trained CNN models, namely VGG16, DenseNet201, and EfficientNetB7 are tested and finally combined to form the proposed ensemble model. Ensemble approach synergizes the features extracted by deep CNN models to deliver accurate predictions and further improve classification. This approach not only enhances the model performance but also reduces generalization error as compared to a single model. To show the efficacy of this proposed model, it has been compared with the existing pre-trained models and tested for 3-class, 4-class, and 5-class on public available datasets. The proposed models' performance is estimated in terms of accuracy, precision, recall, and f1-score parameters and achieved better results for detection purpose. Hence, the proposed model is a promising diagnostic tool for accurate screening of COVID-19 disease.

Keywords COVID-19 · Chest X-ray images · Ensemble model · Multiclass

L. Bagadi (✉) · B. Srinivas · D. R. Ramesh · P. Suryaprasad
Department of ECE, MVGR College of Engineering (A), Vizianagaram, India
e-mail: lavanyabagadi@gmail.com

B. Srinivas
e-mail: srinivas.b@mvgrce.edu.in

D. R. Ramesh
e-mail: rajaramesh@mvgrce.edu.in

P. Suryaprasad
e-mail: suryaprasad@mvgrce.edu.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_19

1 Introduction

COVID-19 has infected millions of individuals and claimed thousands of lives all over the world with an exponential growth in both infections and deaths. So, there is a need to distinguish COVID-19 infected patients from other similar lung infections for saving human lives at an early stage. The common diagnosing techniques available are chest X-ray scan, computed tomography (CT) scan or reverse transcription-polymerase chain reaction (RT-PCR) test [1]. RT-PCR test is the most effective one but cannot be handled with the increasing number of infected people due to its time-consuming or delay process and includes intense lab work to get the result, after collecting nose or throat samples. CT scan is conducted faster but costly and, in most cases, CT scan dataset is not publicly available. To overcome these issues, the early detection method used which is low cost and time effective is chest X-ray images. Hence, analysis of chest X-ray images is considered in this paper.

Majority of radiologists face real difficulty in distinguishing COVID-19 from other similar cases on radiographic images. Because, several viral pneumonia images seem to be like other infectious and inflammatory lung diseases and overlap with them. Therefore, to provide a precise solution to support the clinical diagnosis of COVID-19 in a quick and inexpensive way is involving artificial intelligence with deep learning [2].

Deep learning has evolved to be one of the emerging techniques and found to produce considerable results in various fields like medicine, agriculture and remote sensing applications. It is used in medical field for the detection and classification of different diseases like skin and lung cancer, pneumonia, and Alzheimer's [3], etc. These deep learning techniques improve the diagnosing efficiency and help doctors to significantly save patient's life by early diagnosis. The use of deep learning techniques can avoid human errors in detecting different diseases and avoid manual selection of parameters. It uses convolutional neural network (CNN) as one of the popular methods among which AlexNet, ResNet, Inception, DenseNet, and VGG are few of them.

Worldwide experience is gained by researchers in the classification of X-ray or/and CT scan images. The author Shankar et al. [4] gathered a dataset containing 219 COVID-19 positive X-ray images, 220 healthy lung images, and 220 pneumonia images. The models used in this case are VGG19, ResNet50, MobileNet, and user-defined model. The user-defined model is made of max pooling, same padding and batch normalization layers. ReLu and softmax activation functions have been applied to model after each layer. The classifier classifies the images into COVID-19 and non-COVID-19 in binary classification task, whereas for multi-class, it classifies images into 3 classes, namely COVID-19, pneumonia or healthy. VGG19 showed an accuracy of 100% and user-defined model with 95.33% accuracy for binary class. For triple class, VGG19 provided an accuracy of 98.67% and user-defined model offered 93.33% accuracy [4].

Rashid et al. [5] have worked on the same issue based on transfer learning method. The model has trained X-ray images of 408 COVID-19, 1590 normal lung images,

and 4273 pneumonia images. Model architecture used InceptionV3 followed by data augmentation, and along with classification layers like GAP, ReLu, dropout, and softmax layers. For training 40 epochs, a batch size of 32 is used. The model was trained for each of the 5 folds of data and tested on 3-class test set of each fold. It is observed that binary class accuracy of 99.39% is achieved by the model for 2 folds and 3-class accuracy of 98.37% for 5 folds.

Kanakaprabha et al. [6] analyzed detection of COVID-19 and pneumonia using deep learning on chest X-ray images. The dataset contains 219 COVID-19 chest X-ray images, 1341 normal, and pneumonia images of 1345. Image is initially classified as infected and normal. Further, the infected images are classified into COVID-19 and pneumonia. Finally, the pneumonia cases are classified into bacterial and viral pneumonia. The model attains 95% accuracy for COVID-19 detection with subtype as viral or bacterial pneumonia identification accuracies of 91.46% and 80%.

Chaudhary et al. [7] developed a model that detects COVID-19 and CAP using chest CT scan images. The dataset used is SPGC dataset which contains 171 COVID-19 chest CT scans, 60 CAP, and 76 normal images. Each CT scan has 25 slices. In stage 1 of the model, slice label prediction is done using DenseNet-121 architecture in which the binary classification [COVID-19 and CAP] is done. EfficientnetB6 architecture is used in stage-2 classification for diagnosis. In this stage, model is further classified into 3 classes (COVID-19, CAP, and Normal). Stage-1 has 94% accuracy and stage-2 has 89.3% accuracy.

Hilmizen et al. [8] used chest CT scan and X-ray images for diagnosing COVID-19 pneumonia from deep learning multimodal. The combined models used are ResNet50 with VGG16, DenseNet-121 with MobileNet and Xception with Inception. DenseNet-121 with MobileNet showed maximum accuracy of 99.87% and faster compared to all models.

The contributions of this paper include:

- Three different pre-trained CNN models, namely VGG16, DenseNet201, and EfficientNetB7 models are implemented and evaluated on lung X-ray image dataset.
- To improve the classification accuracy further and achieve optimum performance, ensemble model is proposed and considered.
- The comparison of accuracy for pre-trained models with proposed ensemble model for 3-class, 4-class and 5-class datasets is performed.

The organization of rest of the paper into different sections is as follows. Section 2 provides with the data collection of images from various datasets used along with the pre-processing of images. The details of the proposed ensemble model are also discussed in this Sect. 2. Result analysis part is presented in Sect. 3. Lastly, conclusion is given in Sect. 4.

Fig. 1 Sample images of 5-class labeled dataset

2 Methodology

2.1 Data Collection

Three publicly available datasets of chest X-ray images with different classes are considered for testing the pre-trained models and evaluating the proposed model.

- 3-Class dataset (**COVID-19**, **Normal**, **Pneumonia**)—Total 519 chest X-ray images of COVID-19 infected patients are taken, of which 403 images for training and 116 for testing are used. Out of 1425 healthy patient images, 1108images for training, and 317 for testing purposes are used. Total 3845 pneumonia patient images, 2991 are used for training, and 855 for testing [9].
- 4-Class dataset (**COVID-19**, **Lung Opacity**, **Normal**, **Viral**) —Chest X-ray images of 1600 are collected for each class of COVID-19, Normal, and Lung Opacity. Out of which, each class used 1000 training images and 400 testing images. For viral pneumonia class, 1000 training images, and 345 testing images are used[10].
- 5-Class dataset (**COVID-19**, **Lung Opacity**, **Normal**, **Viral** and **Bacterial**)—A balanced dataset of 611 images of chest X-ray for each class is collected from COVID-19 infected subjects. Such as, Healthy, Lung Opacity, Viral, and Bacterial Pneumonia cases. From each class, 404 images for training, and 207 for testing are used [11].

  Few sample images for the above mentioned dataset is shown below in Fig. 1.

2.2 Pre-processing

The three chest X-ray datasets with different classes considered for experimentation contains images of different size and resolutions. Pre-processing step involves resizing and normalization. So, to ensure consistency across the datasets, all input images are resized to 224×224 pixels that speeds up the training model convergence [12] and normalized to the interval [0, 1].

2.3 *Proposed Model*

The proposed work employs ensemble of three pre-trained models like VGG16, DenseNet201, and EfficientNetB7. Ensembling is termed as method of combining different models to make better predictions resulting in consistent model. The pre-trained models are trained images on high resolution ImageNet database to classify into 1000 different classes. So, while using these models for a new different dataset, the initial layers are freezed to extract the common features and the last few layers are modified to extract specific features in the network.

The pre-trained models used for this work are, namely VGG16, DenseNet201, and EfficientNetB7 are initially trained on the collected datasets individually as mentioned in Sec. 2.1 and their weights are saved. The models are then combined using feature-level ensemble model to produce optimal prediction algorithm. The ensemble model is implemented using without and with dropout, as ensemble learning increases the performance of prediction. Each model is trained and combined to improvise stability and better prediction of the model. The final prediction is made based on the combined results of all the base models used for training purpose. The workflow of the proposed ensemble model is shown in Fig. 2.

The concatenation block produces the extracted features of the neural network from three different transfer learning models [13] and then connected to final output layer for classification. Before classification, the concatenation block combines the initial features collected from the three deep CNN models after they are trained with pre-trained weights. This improves the identification of different classes.

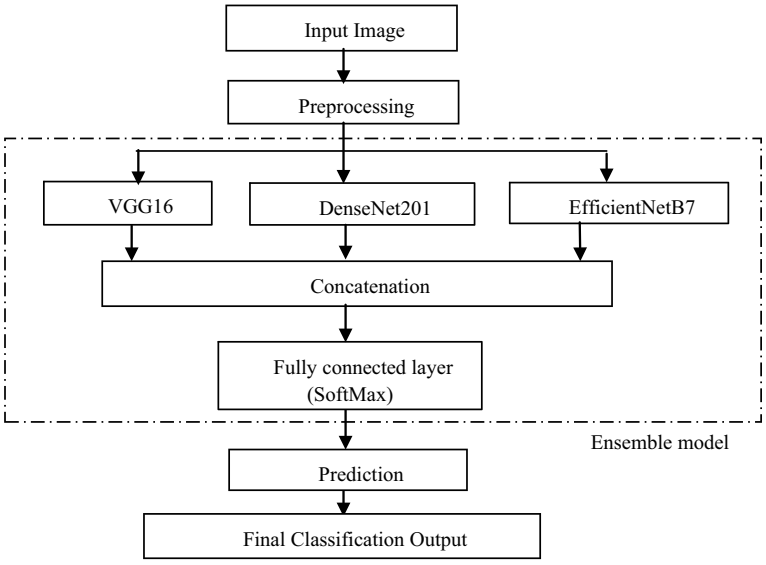


Fig. 2 Proposed ensemble model

The convolutional feature maps generated from different models after global average pooling layer are concatenated before passing them through the fully connected layers for combined optimization. The fully connected layer uses activation function as ReLu and output layer is softmax with variable classes. After concatenation, to ensure further training, a softmax activated fully connected layer is added to adapt to the required number of classes.

3 Result Analysis

The images resized to 224 × 224 × 3 which is given as input to the considered deep learning models. The train: test split of datasets is 80: 20 ratio for different class classification. The training images are fed to the pre-trained model layers to extract their features. The three different datasets discussed are implemented on pre-trained and evaluated on ensemble model. Various metrics like accuracy, recall, precision, and f1-score are calculated from the confusion matrix.

The confusion matrix of proposed model for 3-class is given in Table 1. This table gives COVID-19 images correctly classified as COVID-19 for 111 images out of 116 test images and normal images as normal for 294 images out of 317 total images of normal, and pneumonia patients are correctly classified as pneumonia for 820 images out of 855 images in total.

The comparison plot for 3-class models accuracy for the proposed model along with pre-trained models is shown in Fig. 3. For 3-class dataset, the pre-trained models DenseNet201, EfficientNetB7, and VGG16 show an accuracy of 94.64%, 60.7%, and 89.6%, respectively. The accuracy for ensemble learning without dropout is 84.94% and with dropout is 95.11%.

The confusion matrix of 4-class dataset for the proposed model is given in Table 2. This table classified COVID-19 images correctly as COVID-19 for 361 images, lung opacity images correctly as lung opacity for 360 images, and normal images as normal for 351 images out of 400 total test images of each class mentioned above, and viral pneumonia patients are correctly classified as viral for 322 images out of 345 images in total.

For 4-class, DenseNet201, EfficientNetB7, VGG16 show an accuracy of 90%, 78.5%, and 87.63%, respectively. The accuracy for ensemble learning without dropout is 25.12% and with dropout is 90.81%. The comparison plot for 4-class models accuracy for the proposed model along with pre-trained models is shown in Fig. 4.

Table 1 Confusion matrix of proposed model for 3-class dataset

True/Predicted	COVID-19	Normal	Pneumonia
COVID-19	111	1	4
Normal	7	294	16
Pneumonia	3	32	820

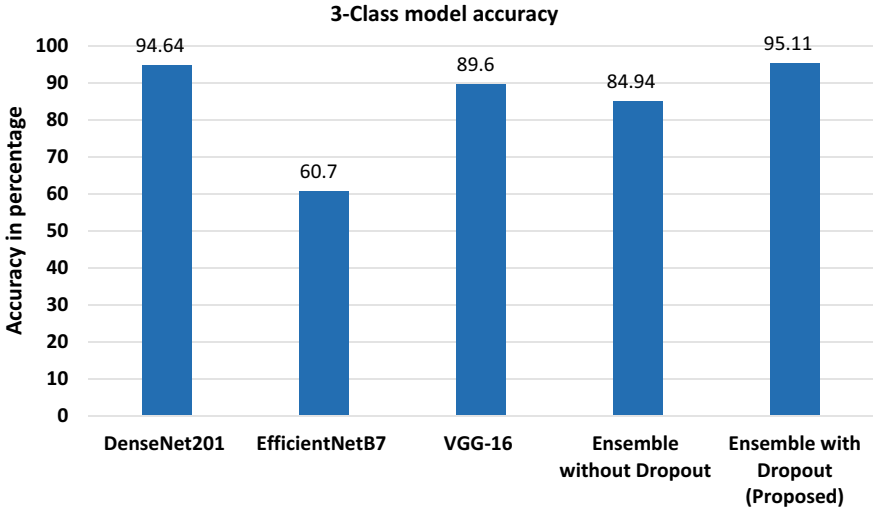


Fig. 3 Comparison of 3-class models accuracy for proposed model

Table 2 Confusion matrix of proposed model for 4-class dataset

True/predicted	COVID-19	Lung opacity	Normal	Viral
COVID-19	361	36	1	2
Lung opacity	26	369	3	2
Normal	11	6	351	32
Viral	12	3	8	322

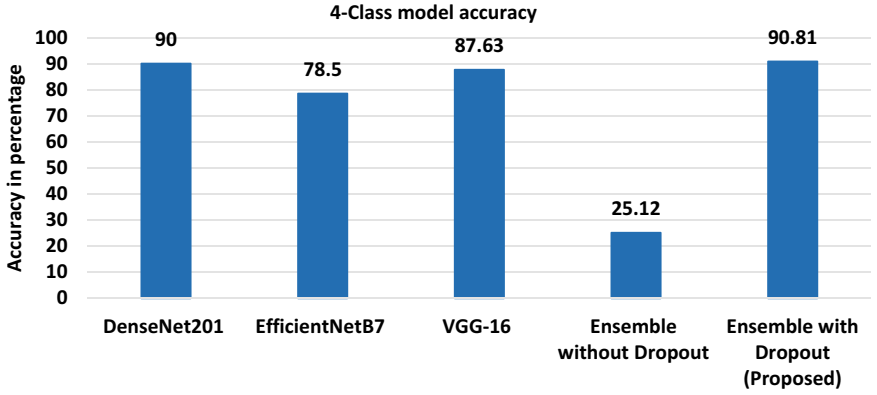


Fig. 4 Comparison of 4-class models accuracy for proposed model

The confusion matrix of 5-class dataset for the proposed model is given in Table 3. This table classified COVID-19 images correctly as COVID-19 for 199 images, lung opacity images correctly as lung opacity for 200 images, and normal images as normal for 64 images, viral and bacterial pneumonia patient images are correctly classified as viral and bacterial for 167 and 86 images, respectively, in total of 207 test images of each class mentioned below. Table 3 gives the confusion matrix with 5-class dataset for the proposed model.

For 5-class dataset, DenseNet201, EfficientNetB7, VGG16 show an accuracy of 58.74%, 56.43%, 62.66%, respectively. The accuracy for ensemble learning without dropout is 47.03% and with dropout is 69.17%.The comparison plot for 5-class models accuracy for the proposed model along with pre-trained models is shown in Fig. 5.

Table 4 gives the ensemble learning model comparison with pre-trained models like VGG16, DenseNet201, and EfficientNetB7. The proposed model results in better values as highlighted in bold when compared to other models presented in the table. In all the cases, ensemble with dropout has shown better results and high accuracy.

Table 3 Confusion matrix of proposed model for 5-class dataset

True/predicted	Bacterial	COVID-19	Lung opacity	Normal	Viral
Bacterial	86	4	4	14	99
COVID-19	0	199	5	3	0
Lung opacity	1	2	200	3	1
Normal	2	18	121	64	2
Viral	17	8	2	13	167

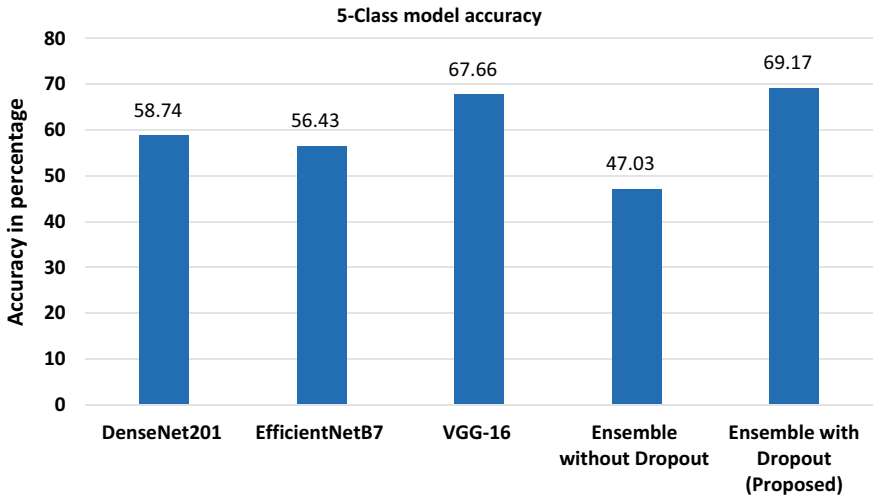


Fig. 5 Comparison of 5-cass models accuracy for proposed model

For 3-xlass, DenseNet201, EfficientNetB7, VGG16 show an accuracy of 94.64%, 60.7%, 89.60%, respectively. The accuracy for ensemble learning without dropout is 84.94% and with dropout is 95.11%.

Further, in Table 5, the performance parameters like recall, precision, and f1-score are measured along with accuracy values for 3-class, 4-class and 5-class for ensemble learning model with dropout (proposed and better) case only. A dropout of 0.2 is used. For 3-class, precision, recall, and f1-score are 95.78%, 95.08%, and 95.92%, respectively. For 4-class, precision is 90.89%, recall is 90.95%, and f1-score is 90.91%. For 5-class, precision, recall, and f1-score are 69.87, 71.11%, and 70.12%, respectively.

Figure 6 shows the comparison of performance parameters of accuracy, precision, recall, and f1-score for the proposed model with three different class datasets like 3-class, 4-class, and 5-class. The proposed model showed better accuracy for all the parameters taken.

For 3 and 4-class classification, ensemble learning with dropout has shown better accuracy and loss when compared to pre-trained models. The reason for decrease

Table 4 Comparison of proposed model with pre-trained models

No. of classes	Models	Accuracy
3-class	DenseNet201	0.9464
	EfficientNetB7	0.6070
	VGG16	0.8960
	Ensemble layer without dropout	0.8494
	Ensemble layer with dropout (**Proposed**)	**0.9511**
4-class	DenseNet201	0.9000
	EfficientNetB7	0.7850
	VGG16	0.8763
	Ensemble layer without dropout	0.2512
	Ensemble layer with dropout (**Proposed**)	**0.9081**
5-class	DenseNet201	0.5874
	EfficientNetB7	0.5643
	VGG16	0.6766
	Ensemble layer without dropout	0.4703
	Ensemble layer with dropout (**Proposed**)	**0.6917**

Table 5 Performance parameters of proposed model

True/Predicted	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
3-class	95.11	95.78	95.08	95.92
4-class	90.81	90.89	90.95	90.91
5-class	69.17	69.87	71.11	70.12

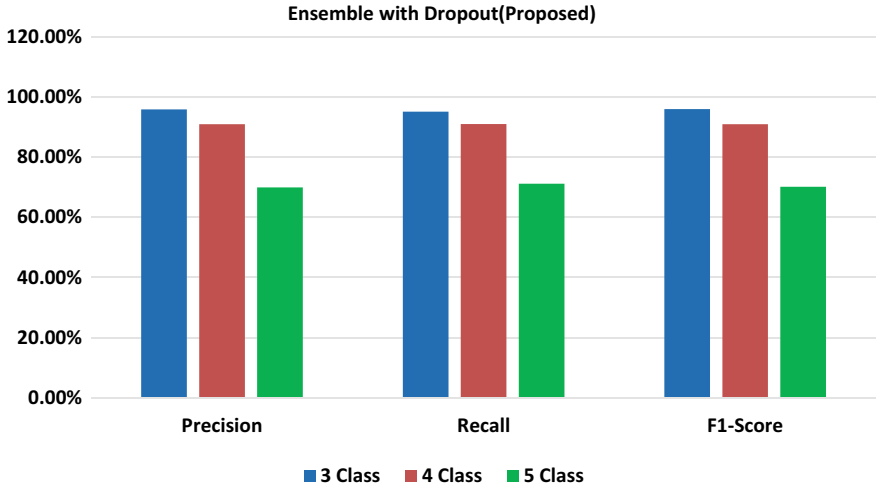


Fig. 6 Comparison of performance parameters for proposed ensemble model

in accuracy for 5-class is, as seen from confusion matrix (Table 3) the model is confusing for viral and bacterial cases. So, when the number of classes increases, the performance of ensemble learning is decreasing but can be improved if the dataset collected is properly able to discriminate within the various classes incorporated in it. The challenging task faced with 5-class is to clearly differentiate between viral and bacterial pneumonia cases as they look almost similar and is the cause for fall in accuracy that needs to be improved further. This can be achieved with proper collection of different classes for a particular dataset.

4 Conclusion

This paper implemented three models like VGG16, DenseNet201, and EfficientNetB7 for the detection of COVID-19. The ensemble learning model proposed by combining these models showed better performance compared to individual pretrained models. The proposed ensemble model with dropout has shown better accuracies of 95.11% for 3-class, 90.81% for 4-class and 69.17% for 5-class. The model achieved consistent accuracy for less than 10 epochs only means the model converges faster on all different class classification problems. Hence, the proposed model helps in assisting clinicians for detecting COVID-19 disease and providing treatment at an early stage to save human lives from this deadly disease.

References

1. Turkoglu M (2020) COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble. Appl Intell 51:1213–1226. https://doi.org/10.1007/s10489-020-01888-w
2. Hashmi HAS, Asif HM (2020) Early detection and assessment of Covid-19. Front Med 7:311
3. Biswas S et al (2021) Prediction of COVID-19 from chest CT images using an ensemble of deep learning models. Appl Sci (11):7004. https://doi.org/10.3390/app11157004
4. Shankar A et al (2020) Detection of COVID-19 using chest X-ray scans. In: 2020 IEEE Bangalore humanitarian technology conference (BHTC)
5. Rashid N et al (2020) Transfer learning based method for COVID-19 detection from chest X-ray images. In: IEEE region 10 conference (TENCON)
6. Kanakaprabha S, Radha D (2021) Analysis of COVID-19 and pneumonia detection in chest X-Ray images using deep learning. In: 2021 international conference on communication, control and information sciences (ICCISc)
7. Chaudhary S et al (2021) Detecting Covid-19 and community acquired pneumonia using chest CT scan images with deep learning. In: IEEE international conference on acoustics, speech and signal processing (ICASSP)
8. Hilmizen N et al (2020) The multimodal deep learning for diagnosing COVID-19 pneumonia from chest CT-scan and X-ray images. In: 3rd international seminar on research of information technology and intelligent systems (ISRITI)
9. https://www.kaggle.com/datasets/ibrahimsobh/chest-x-ray-covid19
10. https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database
11. https://www.kaggle.com/datasets/edoardovantaggiato/covid19
12. Rehman ZU et al (2021) Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: an application for smart agriculture. IET Image Proc 15:2157–2168
13. Muhammad U et al (2022) SAM: self-augmentation mechanism for COVID-19 detection using chest X-ray images. In: Institute of electrical and electronics engineers (IEEE)

Transfer Learning-Based Effective Facial Emotion Recognition Using Contrast Limited Adaptive Histogram Equalization (CLAHE)

D. Anjani Suputri Devi, D. Sasi Rekha, Mudugu Kishore Kumar, P. Rama Mohana Rao, and G. Naga Vallika

Abstract Recognition of facial emotions are the recent research area computer vision's field and human computer interactions. Profound models of deep learning are being widely employed to study the rate of acknowledging facial feelings. For images with noise and poor visibility, deep learning models may perform poorly. The field of emotion recognition faces challenges because to things like facial decorations, unlevel lighting, different stances, etc. Feature extraction and classification are the key drawbacks of emotion detection using conventional methods. A new method called Transfer Learning-based Effective Facial Emotion Recognition Using Contrast limited adaptive histogram equalization has been developed to address this issue (CLAHE). The obtained dataset is initially sent through a combined trilateral filter to remove noise. To improve image visibility, the filtered images are next treated to CLAHE. Jobs requiring classification require the use of techniques of deep learning. Transfer learning techniques are employed in this study to address emotion recognition. This research is related to one of the important networks pre-trained Resnet50, and Inception V3 networks are used. It removes the complete associated layer's from the pre-trained ConvNet's and replaces them with fully associated layers that are appropriate for the guidance count of the proposed assignment. Here, technique was carried out using CK+ database, and it recognizes emotions.

Keywords CK+ · Transfer learning of deep CNN · Pre-trained Resnet50 · Pre-trained InceptionV3

D. A. S. Devi (✉) · D. Sasi Rekha · P. R. M. Rao · G. N. Vallika
Sri Vasavi Engineering College, Tadepalligudem, Andhra Pradesh, India
e-mail: anjanihasini@gmail.com

M. K. Kumar
Aditya College of Engineering, Surampalem, Andhra Pradesh, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_20

1 Introduction

As artificial intelligence advances, it is predicted that robot's eventually be able to understand human emotions [1]. The efficient recognition of emotions, however, is the definition of an ill-posed problem [2]. There are many methods that can be used to recognize the human emotions [3]. Numerous emotion identification systems have been developed and used in the literature to identify human emotions [4]. Many more networks are available for to train the networks but still have performance degradation problem recognization system [5]. FER uses two techniques like traditional and deep learning models [6]. Because of this, it first obtains face data before describing an emotion [7]. Major networks are developed for CNN and RNN [8].

Traditional methods are also available for recognizing face expressions but they have least classification accuracy so we better to go and use deep learning methods to organize the FER system [9, 10]. The max size of the dataset's are important for deep learning's architectures; larger datasets produce higher performance. To increase the amount of data available, researchers are using techniques such data augmentation [11], translation, standardization, cropping, noise addition, and scaling methodologies [12]. CNN is the most effective methods for segmentation and classification applications. The automatic feature extraction offered by this convolution neural network is one of its main advantages. Transfer learning is the most well-known network in the area of deep learning. It is one of the reusable networks. First one model is developed for one task and the same one is used for another different kind of model to train the network [13]. By using transfer learning, it achieves very high speed and dynamic in concatenate and coordinate the upper bound of the networks [14]. Shamoilet al. [15] discussed a method for transferring learning using a SVM classifier. CNNs are used successfully by the authors of [16] to identify facial emotions. The experiment was conducted using a variety of models, including VGG 19, VGG 16, and ResNet50, utilizing the FER'2013 dataset. With a result of 72.18%, VGG16 have the good accuracy of the three models. Despite the fact that these algorithms yield good results, the overfitting issue is a concern. Additionally, these models struggle with photographs that are noisy and have poor visibility. The overfitting issue was resolved in the proposed work by utilizing the transfer learning technique.

The following are this paper's main contributions:

1. A new technology for recognizing facial emotions based on deep learning is suggested.
2. On the collected dataset, the joint trilateral filter is used to eliminate noise, and to make images more visible, CLAHE is added to the filtered photos.
3. The Nadam optimizer is used to optimize the cost function.
4. Finally, facial emotion recognition employs a transfer learning strategy.
5. The dataset and competing FER's are used to consider experiments.
6. The following are the remaining arrangements of the paper: The related work is discussed in Sect. 2. The proposed model is mathematically described in Sect. 3. Comparative outcomes are talked about in Sect. 4. In Sect. 5, concluding remarks are offered.

2 Related Work

The process of creating machines that can recognize and mimic human emotions is known as affective computing [17]. In order to interact with people, affective computing tries to offer computers more intelligence. Only a few applications of emotional computing can be found in the fields of distance learning, banking of Internet, assistants, medical, and security [18]. Identification of human emotions through voice signals, body language, or facial expressions is the first stage of affective computing [19]. An artificial neural network (ANN) model for recognizing facial emotions was put out by Li and Deng [20]. The neural network of spatial–temporal recurrent was created by Zhang et al. [21] to identify facial emotions. In order to obtain better outcomes, Kims et al. [22] applied hiera rchical's deep learning's networks to identify adaptive facial features (HDL). Human emotion recognition is frequently performed using Support Vector Machine [23, 24], Random Forest [25, 26], and ANNs [20]. Jain's et al. [27] developed deep neural networks using deep residual blocks. Wang discussed [28] RNN combined CNN and to categorize human emotions (CCNNRNN). A ResNet and attention block (CRAB)-based human emotion recognition model was created by Arpita et al. in 2020 [20]. The features were extracted by Lakshmi's [29] using a histograms of oriented gradient's and LBPs or HOGLBP. An emotion's recognition system based on the LeNet architecture was proposed by Ozdemir et al. [30]. Some datasets from JAFFE, KDEF, and the author's own bespoke data. The ResNet50 and VGG16 architectures were presented by Dhankhar et al. [31] for the aim of recognizing facial expressions. The majority of the algorithms now in use, according to linked studies, perform well overall but have trouble with the overfitting issue. Additionally, images with noise and poor visibility don't work well with the existing models.

3 Proposed Work

Here, deep CNN is built to identify emotions of humans taken from facial photographs. The recommended network might dramatically focus on the key elements of the photographs occurring in the training phase. To lessen noise, the obtained dataset is initially treated using a combined trilateral filter.

3.1 Modified-Joint-Trilateralfilter

To lessen image noisyness, an novel modified-joint-trilateral-filter is recommended. Edge's preservation refers to the fact that no additional artifacts of any type are added to the filtered images when employing this filter. The first consideration is the guided picture Gd, or the images one kit self. Assume that Ival's and Gval's, respectively,

represent the illumination levels at pixel q and the guided picture. Operating at k [32, 33] is the bilateral filter-dependent kernel window Wr. The modified joint trilateral filter is described as follows:

$$J_{tf}(I_\kappa) = \frac{1}{\sum_{q\in k_r} M_\theta^{pq}(G_d)}\left(\sum_{q\in k_r} M_\theta^{pq}(G_d) \times I_q \times \sigma^2\left(I_q, G_d\right)\right) \tag{1}$$

Next the kernel weight function is rewrited as.

$$M_\theta^{pq}(G_d) = \frac{1}{n^2}\sum_{n;pq\in kr}\left(1 + \frac{\left(G_{dp} - \mu_n\right)\left(G_{dq} - \mu_n\right)}{\frac{\sigma^2}{n} + \in}\right) \tag{2}$$

Here, pixels numbers are n presented window and σ_n^2 displays variance's of G_d in the localized window k_r.

3.2 CLA Histogram Equalization

The accuracy of emotion identification is often poor if the face's visibility changes. The traditional histogram equalization-based solutions, however, are useless for uneven lighting conditions because they result in overly improved pictures [34]. CLAHE can combat the noise overamplification by lowering the contrast. This is how it can be put into practice:

$$\begin{aligned} f(D) = {} & (1-\omega y)((1-\omega x)\mathrm{ful}(D) + \omega x f\mathrm{bl}(D)) \\ & + \omega y((1-\omega x)\mathrm{fur}(D) + \omega x f br(D)) \end{aligned} \tag{3}$$

To identify the best results, each pixel should be assigned four consecutive cumulative-distribution-function (CDF) of the histogram.

3.3 The Proposed Models' Training Process

Today, the extraction of human emotions plays a significant role of computing affective. Emotion process of recognition utilizing trained Convnets is shown in Fig. 1. CK+ dataset of sample images are shown in Fig. 2. The initial process of step is image scaling. The inputs must be scaled to every image is therefore resized to fit the input dimensions of previously trained Convnets. The layers of the pre-trained Convnets are then all frozen, with the exception of the layers that are fully coupled. To summarize, only weight updates can be trained on the fully connected layers. Count classes in a totally connected layer are the basis for classifying emotions. ResNet 50

and Inception V3 are used in this work. Many more pre-trained networks are utilized to train the TL techniques. All of these pre-trained networks are optimized using the Adam optimizer.

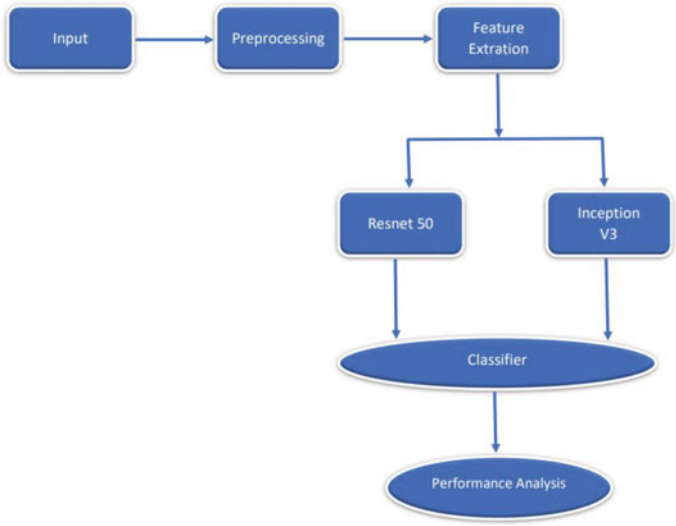


Fig. 1 Process of emotion detection

Fig. 2 Sample images from CK+ dataset

3.3.1 Proposed Transfer Learning Technique

The term "transfer learning" refers to a method of adapting a model created for one industry to another. Transfer learning's central idea is to use a model that have been developed upon complex dataset's and have been applied it to a smallest dataset. The creation of a convolutional brain network without any prior training, which takes more information and is more expensive, is computationally inefficient and requires more information than transfer learning. All models go through a similar training approach to that of the emotion dataset. The weights chosen for the ImageNet database before the models are trained. Pre-trained classification output layer is subsequently swapped out with a new layer formed from the transfer learning benefit.

Adam Optimizer

Adam is an improvement calculation that can be utilized to iteratively update network loads in light of preparing information rather than the ordinary stochastic slope plummet strategy. By applying three proposed methodologies, Adam optimizer is used for optimization purpose. In below algorithm β is moment vector and m is moment estimate.

Algorithm 1

Step1: Initialize the step size value.
Step2: Calculate exponential decay rates for the moment estimates.
Step3: Calculate object function of stochastic parameters with respect to θ.
Step4: Initialize parameter vector θ_0
Step5: Initialize the first moment vector.
Step6: Initialize timestep function.
Step7: Estimate biased first moment.
Step8: Compute bias corrected second raw moment.
Step9: Update the parameters.
Step10: Resulting the parameters.

3.3.2 Resnet50

50 layers are used in resnet50. Here the number 50 indicates the number of layers. It is also one of the CNN technique. The residual block very notable building blocks are the Resnet's architecture, with the residual's block's objective being to create connections between real inputs and projections represented in Fig. 3.

Prediction x and residual $F(x)$ in the diagram above. When x equals the actual input, $F(x)$ has a value of zero. The same x value is then copied by the identity connection. The basic components of the ResNet50's architecture are '5' phases with convolution of identity blocks. The ResNet50 has three channels and a 224*224 input size. It starts out with a max-pooling layer with a kernel size of 3*3 and a convolution layer with a kernel size of 7*7. In this architectural design, each identity block and for each convolution block each have '3' convolution's.

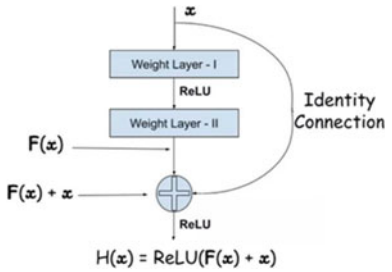


Fig. 3 Residual block

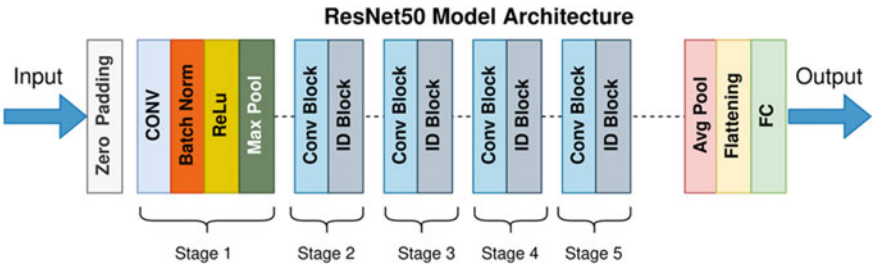


Fig. 4 ResNet50's architecture

layers. The average pooling layer comes after the first five stages, and the fully linked layer with 1000 neurons comes last. Figure 4 depicts the resnet50's architecture. According to our research, we used the ResNet50 model as the foundation and then built it.

3.3.3 Inception V3

Convolution neural network models include Inception V3. The 48 layer deep network used by Inception V3 has an input size of 299*299 pixels. The foundational Inception V3 module is seen in Fig. 5. To minimize dimensionality, the prior to the larger convolutions, and the same is done after the pooling layer as well. The two 3*3 layers are formed by splitting the 5*5 convolutions in half to improve the architecture's performance. N*N convolutions may also be factorized into N*1 and N*N convolutions. The classification of our task, rearranging the last layers of the InceptionV3 architecture'.

3.3.4 Implementation

A piece of each of the three models' execution bounds that were employed in this work. The information shapes for the two organizations and ResNet 50 are similar, however Inception V3 is exceptional. Adam is the optimizer, Softmax is the classifier,

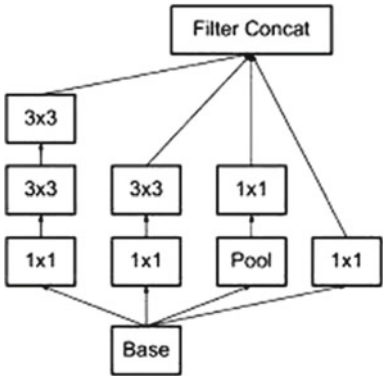


Fig. 5 Block diagram of the Inception V3 module

and categorical cross entropy is the loss function used in the models. Loads are implemented from ImageNet for all organizations. The coordinate technique used in every models is batch specification. Additionally, a few parameters—including Batch size, Epoch size, Dropout—are the same for all three models. The execution settings of Resnet 50 and Inception V3, Softmax classifier, and Adam optimizer are employed.

4 Results and Discussions

The experiment outcomes for the variety models utilized in this process are shown below.

4.1 The ResNet 50 Test Data Results

Table 1 shows that the model performs best at predicting the emotions of melancholy and less well at predicting the emotions of happiness. The suggested model's performance metrics are shown in the table below utilizing Resnet50 as a feature extractor. The proposed ResNet 50 achieves highest specificity, sensitivity, and accuracy than existing prevailing methods.

Table 1 ResNet50 model performance measures

	TP's value	TN's value	FP's value	FN's value	Sensitivity's value	Specificity's value	Precision's value	F1score's value	Accuracy's value
Angry	20	130	2	2	0.90	0.98	0.90	0.9	0.97
Contemptness	7	129	1	2	0.77	0.98	0.77	0.76	0.97
Disgustness	32	105	2	1	0.96	0.98	0.94	0.94	0.98
Fearness	9	139	2	1	0.9	0.98	0.81	0.84	0.98
Happy	23	125	6	3	0.88	0.95	0.79	0.83	0.93
Sadness	9	137	1	2	0.81	0.99	0.9	0.84	0.97
Surprise	49	107	1	3	0.94	0.99	0.98	0.95	0.97
Results					**0.88**	**0.97**	**0.87**	**0.86**	**0.96**

4.2 The Inception V3 Test Data Results

Table 2 gives that the model is quite good at predicting surprise emotions but less good at predicting pleasant emotions. Using Inception's V3 as a feature extractor, the table below gives the suggested model's performance metrics. According to the calculations above, the model's accuracy when, and the F1 score is 0.75. Using Inception V3, the following Figs. 6 and 7 demonstrate the suggested model's accuracy and loss.

Proposed Methods Comparison

In order to recognize emotions the proposed two methods attained highest accuracy than existing prevailing methods is displayed Tables 3 and 4.

Discussion

The existing works Saravanan et al. [35], Gan et al. [36], Orozco [37], Liliana [16], Zadeh et al. [34] average accuracies are 60%, 64.2%, 90%, 92.81%, 97.1%. But the proposed ResNet 50 and Inception V3 achieves highest accuracy 96% and 97%, respectively.

5 Conclusion

The proposed work presented transfer learning-based effective facial emotion recognition using CLAHE, and a method for recognizing facial expressions of emotion is provided. In this study, facial emotion recognition is performed using pre-trained that were trained on the ImageNet database. The CK+ database was used to validate the studies. ResNet50 accuracy is 9%, Inception V3 accuracy is 97%.

Table 2 Inception V3 model performance measures

	TP's value	TN's value	FP's value	FN's value	Sensitivity's value	Specificity's value	Precision's value	F1-score's value	Accuracy's value
Angry	21	129	2	1	0.91	0.97	0.91	0.9	0.99
Contemptness	8	140	1	3	0.78	0.97	0.78	0.80	0.98
Disgustness	33	116	1	2	0.97	0.99	0.95	0.95	0.99
Fearness	10	140	3	2	0.97	0.99	0.89	0.85	0.99
Happy	24	126	5	2	0.89	0.96	0.78	0.85	0.97
Sadness	10	138	2	1	0.82	0.99	0.9	0.89	0.95
Surprise	50	108	2	2	0.95	0.99	0.99	0.98	0.98
Average results					**0.89**	**0.98**	**0.88**	**0.88**	**0.97**

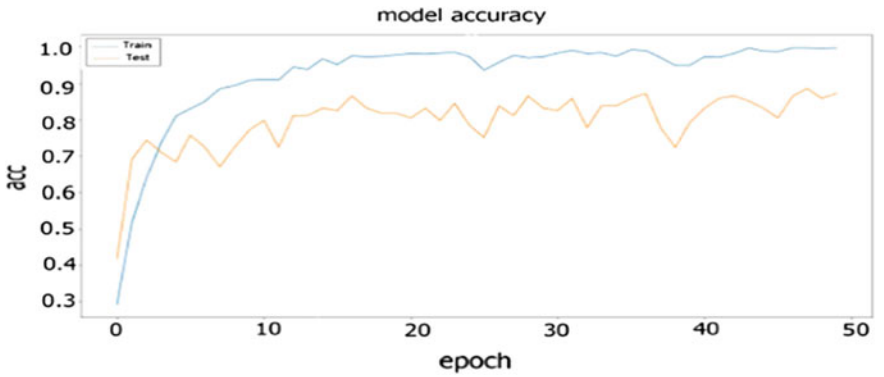


Fig. 6 Performance of Inception V3's accuracy

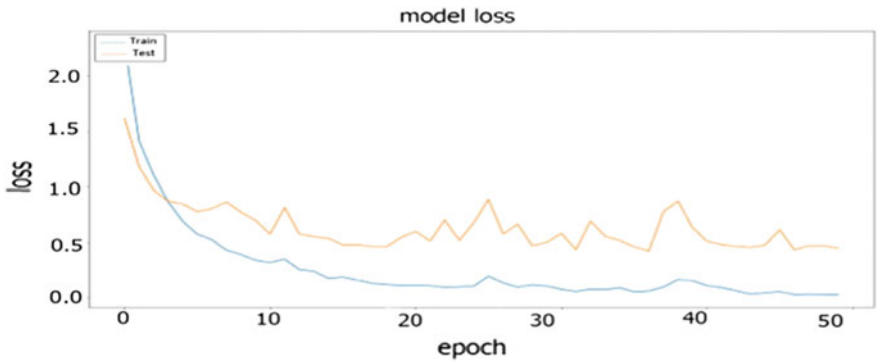


Fig. 7 Performance of InceptionV3's loss

Table 3 Comparison of proposed ResNet 50 and inception V3 networks

Performance metrics	Resnet50	InceptionV3
Sensitivity	0.88	0.89
Specificity	0.97	0.98
Precision value	0.87	0.88
F1-score value	0.86	0.88
Value of accuracy	**0.96**	**0.97**

Table 4 Existing works comparision

Work	Accuracy (%)
Saravanan et al. [35]	60
Gan et al. [36]	64.2
Orozco [37]	90
Liliana [16]	92.81
Zadeh et al. [34]	97.1
Proposed Resnet50, inception V3	**96, 97**

References

1. Zhang H, Jolfaei A, Alazab M (2019) A face emotion recognition method using convolutional neural network and image edge computing. IEEE Access 7:159081–159089. https://doi.org/10.1109/ACCESS.2019.2949741
2. Gao L, Zhang R, Qi L, Chen E, Guan L (2019) The labeled multiple canonical correlation analysis for information fusion. IEEE Trans Multimedia 21(2):375–387. https://doi.org/10.1109/TMM.2018.2859590
3. Zhang T, Zheng W, Cui Z, Zong Y, Yan J, Yan K (2016) A deep neural network-driven feature learning method for multi-view facial expression recognition. IEEE Trans Multimedia 18(12):2528–2536. https://doi.org/10.1109/TMM.2016.259809
4. Hua W, Dai F, Huang L, Xioang J, Gui G (2019) Hero: human emotions recognition for realizing intelligent internet of things. IEEE Access 7:24321–24332. https://doi.org/10.1109/ACCESS.2019.2900231
5. Zhang S, Pan X, Cui Y, Zhao X, Liu L (2019) Learning affective video features for facial expression recognition via hybrid deep learning. IEEE Access 7:32297–32304. https://doi.org/10.1109/ACCESS.2019.2901521
6. Ferreira PM, Marques F, Cardoso JS, Rebelo A (2018) Physiological inspired deep neural networks for emotion recognition. IEEE Access 6:53930–53943. https://doi.org/10.1109/ACCESS.2018.2870063
7. Alam M, Vidyaratne LS, Iftekharuddin KM (2018) Sparse simultaneous recurrent deep learning for robust facial expression recognition. IEEE Trans Neural Netw Learn Syst 29(10):4905–4916. https://doi.org/10.1109/TNNLS.2017.2776248
8. Li TS, Kuo P, Tsai T, Luan P (2019) CNN and LSTM based facial expression analysis model for a humanoid robot. IEEE Access 7:93998–94011. https://doi.org/10.1109/ACCESS.2019.2928364
9. Jain N, Nguyen TN, Gupta V, Hemanth DJ (2021) Dental X-ray image classification using deep neural network models. Ann Telecommun
10. Vu D, Nguyen T, Nguyen TV, Nguyen TN, Massacci F, Phung PH (2019) A convolutional transformation network for malware classification. In: 2019 6th NAFOSTED conference on informa- tion and computer science (NICS), pp 234–239
11. Li S, Deng W (2018) Deep facial expression recognition: a survey. arXiv preprint arXiv:1804.08348
12. Pitaloka DA, Wulandari A, Basaruddin T, Liliana DY (2017) Enhancing CNN with preprocessing stage in automatic emotion recognition. Proc Comput Sci 116:523–529
13. Ng HW, Nguyen VD, Vonikakis V, Winkler S (2015) Deep learning for emotion recognition on small datasets using transfer learning. In: Proceedings of the 2015 ACM on international conference on multimodal interaction, pp 443–449
14. Xu M, Cheng W, Zhao Q, Ma L, Xu F (2015) Facial expression recognition based on transfer learning from deep convolutional networks. In: Proceedings of 11th international conference on natural computation. Zhangjiajie, China, pp 702–708
15. Shaees S, Naeem H, Arslan M, Naeem MR, Ali SH, Aldabbas H (2020) Facial emotion recognition using transfer learning. In: 2020 International conference on computing and information technology (ICCIT-1441). IEEE, pp 1–5
16. Gulati N, Arun Kumar D (2020) Facial expression recognition with convolutional neural networks. Int J Future Gener Commun Netw 13(3):1923–1931
17. Picard RW (1999) Affective computing for HCI. HCI (1):829–833
18. Daily SB, James MT, Cherry D, Porter JJ III, Darnell SS, Isaac J, Roy T (2017) Affective computing: historical foundations, current applications, and future trends. Emotions Affect Hum Factors Hum-Comput Interact 1:213–231
19. Tao J, Tan T (2005) Affective computing: a review. International conference on affective computing and intelligent interaction. Springer, Berlin, Heidelberg, pp 981–995
20. Li S, Deng W (2019) Reliable crowd sourcing and deep locality-preserving learning for unconstrained facial expression recognition. IEEE Trans Image Process 28(1):356–370

21. Zhang T, Zheng W, Cui Z, Zong Y, Li Y (2019) Spatial–temporal recurrent neural network for emotion recognition. IEEE Trans Cybern 49(3):839–847. https://doi.org/10.1109/TCYB.2017.2788081
22. Kim J, Kim B, Roy PP, Jeong D (2019) Efficient facial expression recognition algorithm based on hierarchical deep neural network structure. IEEE Access 7:41273–41285
23. Varma S, Shinde M, Chavan SS (2020) Analysis of pca and lda features for facial expression recognition using svm and hmm classifiers. In: Techno-societal 2018. Springer, pp 109–119
24. Kar NB, Babu KS, Sangaiah AK, Bakshi S (2019) Face expression recognition system based on ripplet transform type II and least square SVM. Multimedia Tools Appl 78(4):4789–4812
25. Valstar M, Gratch J, Schuller B, Ringeval F, Lalanne D, Torres Torres M, Scherer S, Stratou G
26. Cowie R, Pantic M (2016) Avec 2016: depression, mood, and emotion recognition workshop and challenge. In: Proceedings of the 6th international workshop on audio/visual emotion challenge, pp 3–10
27. Jain DK, Shamsolmoali P, Sehdev P (2019) Extended deep neural network for facial emotion recognition. Pattern Recogn Lett 120:69–74
28. Wang X, Chen X, Cao C (2020) Human emotion recognition by optimally fusing facial expression and speech feature. Sign Process: Image Commun 84:115831. https://doi.org/10.1016/j.image.2020.115831. https://www.sciencedirect.com/science/article/pii/S0923596520300540
29. Lakshmi D, Ponnusamy R (2021) Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders. Microprocess Microsyst 82:103834. https://doi.org/10.1016/j.micpro.2021.103834. https://www.sciencedirect.com/science/article/pii/S0141933121000144
30. Ozdemir MA, Elagoz B, Alaybeyoglu A, Sadighzadeh R, Akan A (2019) Real time emotion recognition from facial expressions using CNN architecture. In: Proceedings of international conference on medical technologies national congress. Kusadasi, Turkey, pp 1–4
31. Dhankhar P (2019) ResNet-50 and VGG-16 for recognizing facial emotions. Int J Innov Eng Technol 13(4):126–130
32. Singh D, Kumar V (2017) Dehazing of remote sensing images using improved restoration model based dark channel prior. Imag Sci J 65(5):282–292
33. Kaur M, Singh D, Kumar V, Sun K (2020) Color image dehazing using gradient channel prior and guided 10 filter. Inf Sci 521:326–342
34. Zadeh MMT, Imani M, Majidi B (2019) Fast facial emotion recognition using convolutional neural networks and Gabor filters. In: 5th conference on knowledge based engineering and innovation (KBEI). IEEE, pp 577–581
35. Saravanan A, Perichetla G, Gayathri DK (2019) Facial emotion recognition using convolutional neural networks. arXiv preprint arXiv:1910.05602
36. Gan Y (2018) Facial expression recognition using convolutional neural network. In: Proceedings of the 2nd international conference on vision, image and signal processing, pp 1–5
37. Liliana DY (2019) Emotion recognition from facial expression using deep convolutional neural network. J Phys Conf Ser 1193(1):012004

Transformer Model for Human Activity Recognition Using IoT Wearables

S. Sowmiya and D. Menaka

Abstract Human activity recognition plays a vital role in the modern life of its immense applications in medical care, sports, detection and prediction of unpredicted events and in biometrics field. HAR is the field of study for recognizing human activities from time-series data collected from video, images or from multimodal sensors equipped with smartphone, IoT wearables. A wide area of research in the field of HAR has been presented till now from machine learning and deep learning. Deep learning methods utilize the temporal view of the data, while shallow methods utilize the handcrafted features, statistical view. Most of the deep learning methods utilized CNN, LSTM architecture. Now its time to use the power of transformers in this area. For sequence analysis tasks, transformers were shown to outperform the deep learning methods. Transformers are proved to be best in finding relevant time steps and has the potential to model long term dependencies. In this work proposed, a novel method that employs a transformer-based model to classify human activities from data collected using Intertial measurement Units (IMU). This approach works by leveraging self-attention mechanisms in transformers to explore complex patterns from time-series data. In this paper, a transformer-based approach is experimented for HAR and its evaluation parameters are presented for two data sets USCHAD and WISDM. From the results, we can analyse the improvement in performance matrix and efficiency of the transformer-based model.

Keywords Transformer · Human activity recognition · Time series · IoT sensors · IMU

S. Sowmiya (✉) · D. Menaka
Noorul Islam Centre for Higher Education, Kanyakumari, Tamil Nadu, India
e-mail: sowmiyabashir@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_21

1 Introduction

Because of the advancement in deep learning, computational power and progress in sensor development and its availability in the daily using gadgets like smartphones and smartwatches, HAR using IoT wearables is of prime focus in the research field. Also IoT wearable sensors are small in size, high in sensitivity and have robust anti-interference ability, so the sensor-based identification method is more appropriate for practical circumstances. And when considering the privacy of the subjects, sensors have a significant weightage over other video monitoring methods. Moreover, sensor-based behaviour recognition is not limited by time or scene, and can well portray the nature of human activities. As the percentage of elderly population has a significant increase, HAR has a prominent role in ambient assistant living. The research for HAR started with machine learning techniques with manual feature engineering. The next era is the deep learning techniques with a wide variety and combination of deep models. With the introduction of transformers, its more easy to build an efficient model exploiting its memory capacity of self-attention.

The advanced research area for human activity recognition is especially concentrated on IMU-based recognition. The principal reasons for this inclination are (i) Recent advances in sensor technology which is powered by high speed and low cost electronic circuits (ii) Reduction in the power consumption of the electronic devices. (iii) The sensors are wearable and they are portable as well as more user friendly. (iv) A smaller amount invasion in privacy (iv) Modern sensors give more accurate and reliable output. (v) Sensors like gyroscope, accelerometer, ambient light sensor, proximity sensor, motion sensor and magnetometer are used in modern smartphones. These elements are vital for human activity recognition. An efficacious and fast classification using sensor data is easily possible due to the advances in data science for collecting, analysing and interpreting huge amount of data using deep neural networks. The progress in IoT technology, based on copious sensors and sensor networks, has also resulted in an escalation in the research field of HAR. HAR plays an important role and is widely applied in fields such as ambient assistive living (AAL), context-aware computing, surveillance-based security and assembly tasks analysis in industries. Also, personal biometrics and behavioural interpretation are other essential applications of human activity recognition. While considering the medical field, HAR is used to increase the pace of recovery by monitoring the patients, and thereby aiding the process of medical diagnosis. It is also useful in assisting the elderly and chronic patients along with their helpers. HAR has its significance in applications like game console, sports monitoring and fitness workouts. Most of the earlier attempts for the activity classification employed machine learning (ML) algorithms and have achieved pretty good accuracy rate of 80% or higher. ML and its signal-processing techniques are based on shallow architecture, which contain a single layer utilizing nonlinear functions to learn complex feature and relationships from the data. Even though these techniques are non-adaptive, ML is empowered with sophisticated learning algorithms and efficient pre-processing techniques. ML algorithm gives effective results in controlled or lab environments, where there are

minimal labelled data or where less domain knowledge is required. The drawback of ML algorithms is, they heavily depends on domain experts since they rely on heuristic handcrafted feature extraction [1]. From some statistical values, shallow features can only be learned by machine learning techniques [2] which results in an undermined performance. The disadvantage of this approach for HAR is that it capable to recognize only low level or basic human activities.

Deep learning (DL) refers to a subset of ML techniques, implemented by stacking layers of neural networks used hierarchically for the refinement of accuracy and optimized prediction through better pattern classification and top level feature learning. Four key reasons for the popularity of deep learning methods are (i) advances in GPU processing proficiencies and the diverse service offerings by cloud computing, (ii) automatical feature extraction and classification from huge data, (iii) recent advances in DL models (hybrid) and signal/information processing research, and the model using transfer learning (iv) exploit attention mechanism inherent in transformers for time series classification. The DL techniques have two key properties: the generative or unsupervised nature of the model, which entails including a top layer to accomplish discriminative tasks, which can be used as pre-processing steps for supervised algorithm, and an unsupervised deep learning process that uses a large amount of unlabelled data to uncover hidden features and representations in the data at a deeper level. Another remarkable advantage of this approach is, the procedure of feature extraction and model building techniques is performed simultaneously which speed ups the learning process. Human arbitration is not at all needed and the features can be learned automatically via the deep network. By extracting high level features using deep layers, complex and minute activities can be recognized.

Several research works are investigated using CNN, RNN-LSTM, stacked autoencoders, DBN and various hybrid networks [3–7]. Another research work in deep learning method proposed to use a hybrid method combining gated recurrent units (GRU) [8], and recurrent neural networks (RNN) with a gating system, for the recognition of human activities. The purpose of this study is to examine whether the proposed hybrid architecture can be used to effectually infer pairwise similar activities using sensor data.

Now, most of the research works in activity recognition tasks use the transformer architecture which uses the mechanism of attention to derive dependencies between input and output. The attention mechanism is very effective in finding the modal dependencies for a sequential data. The model in this study uses a transformer-based architecture which utilizes self-attention mechanism, input empowered with sensor-based attention and CNN layer.

The key contributions and intuitions of this paper are synopsized as follows:

- This study proposed a transformer model based on attention mechanism for processing sparse and irregularly sampled multivariate clinical time-series data.
- Based on the multi-head attention mechanism inherent in transformers [9], a self-attention model unit is designed for time-series data from different sensor readings.

- The study proposed to comprise temporal order into the sequence representation of time-series data using the mechanism of positional encoding to inject input position information and a sensor attention module is placed at the input.
- The proposed model is assessed using two publically available ADL data sets; USCHAD and WISDM, and achieved state-of-the-art prediction performance.

The remainder of this paper is structured as follows. In Sect. 2, the process of HAR is presented. In Sect. 3, a description about transformers and its variants are described. In Sect. 4, we present a review of prevailing work in this area of research. In Sect. 5, we present our proposed approach followed by providing the relevant evaluation parameters in Sect. 6, Results and Discussions in Sect. 7, and Conclusion in Sect. 8.

2 Related Work

In 2019, Sun et al. [10] proposed an attention-based model with LSTM for activity recognition from wearables. In the [11] study, the transformer model was proposed for a time series analysis of signals from IMU. The transformer architecture features a self-attention mechanism that allows the model to process the input sequence contextually by encoding the dependencies between the elements in the sequence, can match the performance of state-of-the-art CNN with LSTM model and obtained an outstanding average accuracy of 99.2% with a large data set. Conformer is a hybrid network structure which exploit the power of both CNN and transformers. It takes advantage of convolutional operations and self-attention mechanisms for superior representational learning. The architecture consists of a concurrent structure, in which local representations and global features are hold on to the maximum level. It is proved from the experiments that Conformer, under the comparable parameter complexity, outpaces the visual transformer -DeiT-B by 2.3% on ImageNet. Studies are made by Yeon-Wook et al. [12] on HAR based on IMU using conformer. Using two data sets, they proved that the model outperforms transformer-based model and CNN. Pan et al. [13] used a method of GRU with temporal and channel attention for sensor-based HAR showed that the model outperforms the current methods in terms of classification accuracy on data sets without Independent Identical Distribution. The method has reduced number of model parameters and resources consumption, which has applications in low-resource embedded devices. A similar study was done by Jian et al. for real-time applications [14]. Attention-based CNN is used for the modelling of HAR with weak labels [15]. The research study shows the utilization of weakly labelled data without manual annotation. Attention mechanism is used to utilize compatibility score and convert it into density to determine the specific location of a labelled activity. Sharma et al. [16] studied the effectiveness of CNN-based feature embedding with transformers. 2D-CNN is used for short-term Fourier transformation of the data obtained from accelerometer. 1D-CNN is used for the heart rate, and post fusion feature is given to the input of transformer. The experiments showed that

the discriminative capability of the feature-fusion on transformer combined with the power of CNN outperforms the existing methods by 3.7%.

Utilizing the self-attention mechanism which contextually encodes individual dependencies between signal values in time-series data, another recent research work is proposed based on transformers [17]. This model boosts up performance state-of-the-art CNN in terms of computational speed and memory requirement. The model is evaluated using 3 publically available data set and achieved a F1-Score 19.04% higher than that of existing methods.

Liu et al. [18] combined representation learning with transformers and obtained good time series classification and they obtained an increase in accuracy rate of about 5%.

3 HAR Method

The algorithm for human activity recognition comprises the following steps. Step 1: Acquire the raw input signals from various sensors. Step II: Data pre-processing: In this step, the noise is removed and segmentation of the data is done with overlapping/non-overlapping window. Step III: Data set is categorized into train, test and validation sets. Step IV: Features are manually or automatically extracted through the dense stacked layers of neural network or attention base mechanism. Step V: Classify and make inference in human activity recognition tasks. Figure 1 represents the process of HAR. The data obtained from sensors are to be pre-processes, cleaned before extracting the features. Also the data set has to be split into train, validation and test data sets. Using DL, the features can be extracted automatically. What activities are aimed to be recognized, Simple or Complex? Is it hand-oriented activities or ambulation related activities ? The answer to this question depends on the sensor selection, the number of sensors used, sensor placements and the setting of data acquisition environment. The more number of sensors provide more information about the activity, and thus more complex activities can be identified. The sensors must be placed optimally in waist, pockets, head or wrist depending on the kind of activity to be recognized.

4 Transformers

Transformers provide an easy way to pre-train models with its API. The pre-trained model has several advantages like reduce computing cost, carbon footprint and time saving. The transformers provide a platform for training which is of very helpful for the community including researchers. More than 100 different models of transformers have been released for different applications like NLP, image recognition, etc. In transformer, the self-attention mechanism is used to state individual dependencies

Fig. 1 HAR process

amid signal values within a time-series sensor data, commonly from accelerometer and gyroscope.

4.1 Architecture

Transformer consists of encoders and decoders, the number varies according to the model. Encoder comprises two parts, self-attention layer followed by feed forward neural network, whereas decoder is build using three main components, decoder attention layer, self-attention layer and the feed forward layer. Figure 2 represents the architecture of transformer. Architecture of transformers is based on attention-driven mechanism for sequential data analysis. Transformers are meant to aggregate information from the sequential data. It will also generate a position and context-aware representations. The performance of transformers is proved to be better than the deep learning models RNN and LSTM for various sequence-based problems like NLG, CV, etc. In this paper, an approach based on transformer is used for the recognition of inertial-based activity recognition tasks. In transformers, positional encoders are used to maintain the order of the sequence in data.

4.2 Multi-headed Attention

Attention is a costly mechanism since it need to compute a value for each combination of output and input. In the transformer, encoder-decoder architecture with attention mechanism idea is outspread to learn intra output and intra input dependencies also. Figure 3 shows the structure of multi-headed attention. The attention mechanism used is multi-head attention, where the weights are calculated in parallel which accelerate the training. It has multiple layers of self-attention, in which the values, keys and queries are generated from the input sequence. In multi-head attention mechanism, the output is the result of combining different representations of the values by taking a weighted sum. The weighting is based on the similarity between the query and

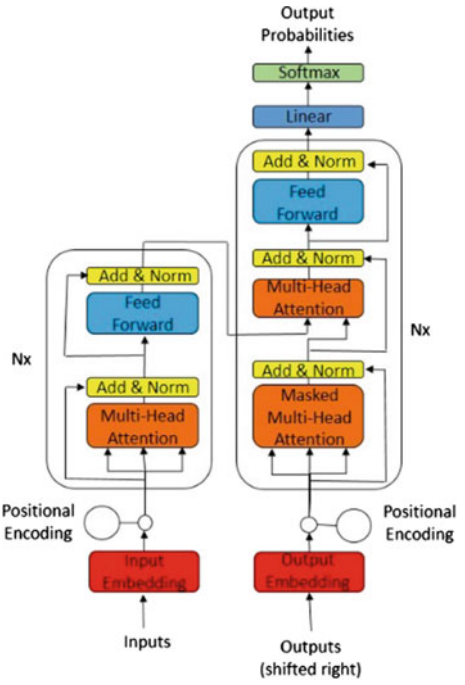


Fig. 2 Transformer model architecture

each of the keys, which is determined by the dot product between them. The output obtained from multiplicative attention weighed sums is a linear conversion of the input representation.

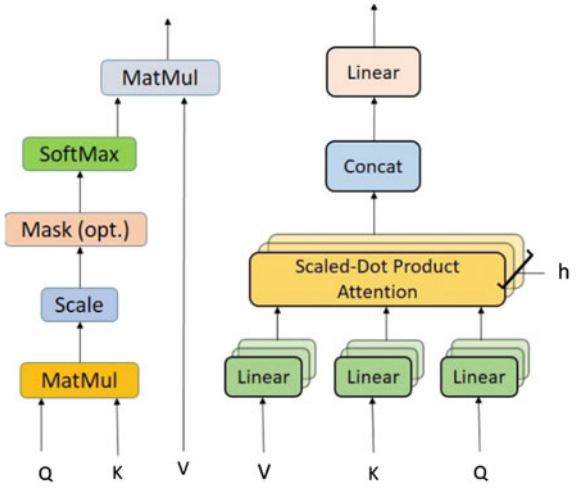


Fig. 3 Multi-head attention

The role of the decoder is to identify the association between sequences by utilizing the attention mechanism. Tokenizer is used for pre-processing of the data prior to training and model building and can be called easily via single string.

4.3 Why Transformers?

The popularity of transformers are increasing due to the reasons (i) A standard API can be used for pre-trained model. (ii) Can reuse trained models and hence the computation cost can be reduced. (iii) Can easily choose the right framework for training, evaluation and production at any part of product's lifetime.

5 Proposed Model

In this paper, a transformer model is proposed for the recognition of inertial-based HAR. Two data sets USCHAD and WISDM are used for evaluation of this model. The number of heads employed in multi-head attention block of this model is 6. A total of 3 encoder blocks are implemented in the entire model and Adam is the optimizer used during training. Figure 4 shows the model architecture for the proposed method. In the input section, a sensor attention layer and a 1D convolutional layer is provided followed by encoder blocks. The output of encoder is driven by attention with context layers which is then passed on to the final shaping and drop out layers (Table 1).

For the proposed model, 3 blocks are used for self-attention. For self-attention, fixed size input is required and so, the dimension was fixed to 128. The number of components in position-wise feed forward layer was set to 4 times the dimension, which is 512.The model used Adam optimizer and a learning rate of 0.001 for training.

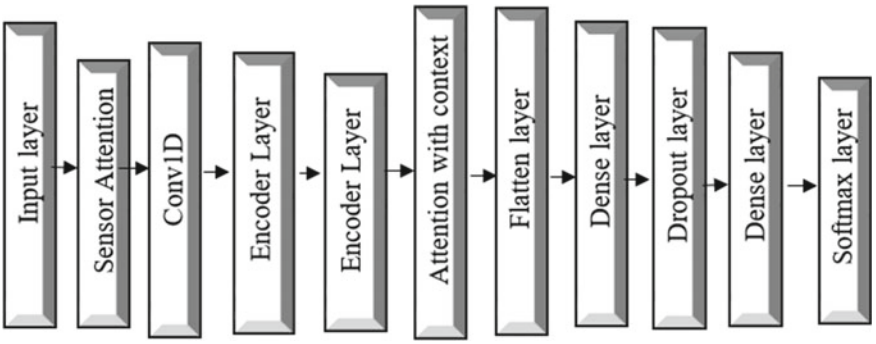


Fig. 4 Architecture of the proposed model

Table 1 Evaluation matrix USCHAD

Activity	Precision	Recall	F1-score	Support
Walking forward	0.90	0.89	0.90	189,300
Walking left	0.90	0.89	0.90	168,000
Walking right	0.86	0.94	0.90	185,600
Walking up-stairs	0.90	0.88	0.89	160,800
Walking down-stairs	0.92	0.83	0.88	151,200
Running forward	0.97	0.94	0.95	105,300
Jumping up	0.90	0.89	0.90	64,700
Sitting	0.88	0.95	0.91	159,500
Standing	0.80	0.88	0.84	148,031
Sleeping	1.00	1.00	1.00	249,969
Micro-avg	0.91	0.92	0.91	1,582,400
Macro-avg	0.90	0.91	0.91	1,582,400
Weighted-avg	0.91	0.92	0.91	1,582,400

5.1 Experimental Set-Up

The proposed model consists of a convolutional base and a classifier head. The convolutional encoder includes two 1D-CNN with RELU nonlinearity function followed by Dropout layer and maxpooling layer. The classifier head consists of two fully connected layers with RELU for nonlinearity. Softmax is applied at the output of the final fully connected layer. Data set is split into 3 groups train, validation and test set in the ratio of 8:1:1,while ensuring all the classes are represented in the three sections. The value of weight decay chosen is 0.0001 and a batch size of 256 is employed in the experiment. A learning rate of value 0.0001 is selected at the beginning of training process and further reduced using ReduceLROnPlateau from Keras. The model is trained for 50 epochs.

5.2 Data Set

Two publically available data set using inertial measurement units are used for this study. USCHAD [19] is USC human activity data set consists of low level activities based on wearable sensors in daily life which is meant to recognize activities especially in healthcare. The data set includes information such as the individual's acceleration, angular velocity and magnetic field measurements, as well as labels indicating the type of activity being performed. The data is collected from a variety of individuals and in various environments, making it a diverse and rich data set for research on human activity recognition. Data set used three axis accelerometer and three axis gyroscope as sensing hardware and is attached to subject's front right

hip. WISDM data set [20] includes data set for observing ADL recorded using a smartphone and a smartwatch. The data set was created using 51 subjects and each of them wore an LGG smartwatch with Android Wear 1.5 on their left wrist hand and had a smart phone Samsung Galaxy S5 or Google Nexus mobile phone in their right pocket. The phone was placed right-side up with screen side facing outwards from the body. 18 distinct activities were performed by each of the contributors with smartphone a and smartwatch for a time duration, 3 successive minutes each. In the proposed work, a sampling rate of 20 per second with a sliding window approach is used. The activities include stationary activities, dynamic activities, ambulation related activities, hand-oriented activities including eating and non-eating activities. The sensor data for activity recognition is taken from readings of gyroscope and accelerometer from both the devices. Activity code is the target variable used and 6 predictor variables used are the sensor readings in x, y and z dimension of both the gyroscope and accelerometer. To conduct the analysis of the activities effectively, the 18 different activities in the WISDM data set is categorized into three groups (1) Ambulation-related activities that are the basic activities performed independently (2) Hand-oriented activities-non-eating, that comprises non-eating activities using hand (3) Hand-oriented activities-eating, that is formed of the activities that include eating.

6 Evaluation Parameters

Evaluating a model is an essential step in the model development process as it helps to determine the model's performance and identify any potential issues. The most frequently used evaluation metric for classification is accuracy. But along with accuracy, other key classification metrics like Recall, Precision and F1-Score are equally important. Confusion matrix is used for a multiclass classification problem to calculate the classification performance. From confusion matrix, Recall (Sensitivity), Accuracy, Specificity, Precision and AUC-ROC curve can be calculated. To get an optimal blend of Precision and Recall, F1-score is used, which is efficient for uneven class distribution. In this research work, the confusion matrix is plotted. Also the evaluation parameters Precision, Recall and F1-Score are presented.

7 Results and Discussions

The parameters Recall, Precision and F1-Score are given in the results. It shows that the transformer model shows outstanding performance with other DL counterparts or with hybrid models like CNN-GRU, CNN-BiLSTM. In Figs. 5 and 6, the confusion matrix is given and from the picture it is clear that the diagonal elements are with good proposition than the other one which highlights a better model for human activity recognition.

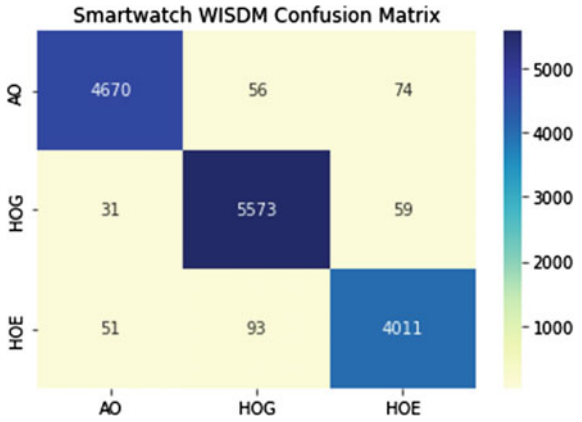


Fig. 5 WISDM confusion matrix

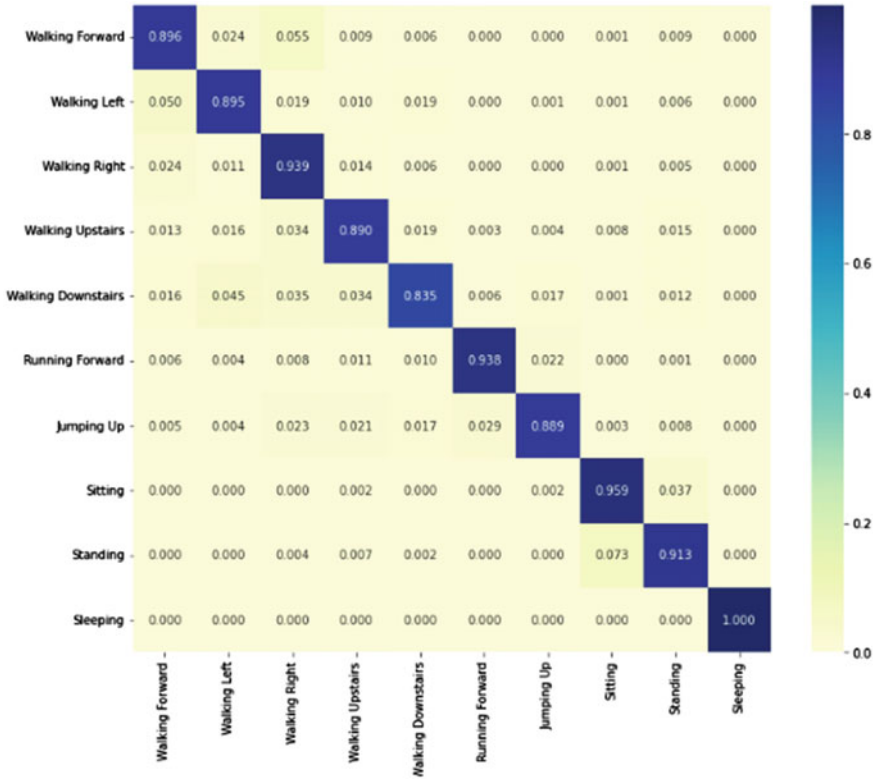


Fig. 6 USCHAD confusion matrix

Table 2 Comparison among various models for USCHAD and WISDM data set

USCHAD		Smartphone WISDM	
Method	Accuracy	Method	Accuracy
CNN-LSTM	90.6	CNN-LSTM	88.7
CNN-BiLSTM	91.8	CNN-BiLSTM	89.2
CNN-GRU	89.1	CNN-GRU	89.6
CNN-BiGRU	90.2	CNN-BiGRU	90.2
Proposed method	91.2	Proposed method	90.8

Table 2 gives the evaluation of the model and its comparison with various hybrid models [21] for two data sets, USCHAD and Smartphone WISDM. It can be shown that the proposed model shows an improvement in accuracy of about 91.2% and 90.8%, respectively. The various hybrid neural network models are experimented and the results are compared with the transformer model which utilizes the attention mechanism for time-series data classification.

8 Conclusion and Future Scope

The modified transformer model proposed in this manuscript is studied to exhibit its relevance as a revolutionary alternative to various deep learning models including RNN or CNN. This model showed a better representation than existing DL models. The transformer can scale reasonably to more than one million parameters depends on the number of classes and can also be used on low power real-time applications. There is no need for pre-transformation of the time-series data and can fed it to the neural network after normalization. Moreover, the transformer model is fit parallelized to execute on GPU [13]. Confusion matrix is presented in Figs. 5 and 6. It is clear from the matrix that the maximum counts lie up in the diagonal elements indicating that match between predicted labels and the true labels are more. Table 1. gives the statistical evaluation of the USCHAD data set. Values of Precision, Recall, F1-Score and support are given in Table 1 for each of the activities. The support is the number of samples of the true response that lie in the activity. Its value range lies in the range 10×10^4 to 15×10^5, which are practically enormous support sets.

The results of this study showed the effectiveness of the transformer model in classifying human activities. The data set selected for the evaluation of this model is the publicly available data set for smartphone motion sensor data, WISDM and USCHAD comprising an extensive range of activities in daily life. The adapted transformer model proposed in the present paper demonstrated an appreciate level of accuracy and precision, indicating that it is a promising method and has a potential to be considered among other cutting-edge technologies for HAR.

In future, it would be beneficial to test the adapted transformer model on a larger data set, incorporating a wider range of sensor data. This will allow for a more

thorough evaluation of the model's performance and provide insights on how well it generalizes to new data. Also various models of transformers from hugging face should be evaluated using different data sets. The pre-training efficiency should be utilized for real-time applications using IoT wearables, and it could lead to new and innovative applications, such as providing direct support for humans.

References

1. Bengio Y (2013) Deep learning of representations: looking forward. In: International conference on statistical language and speech processing. Springer, pp 1–37
2. Yang JB, Nguyen MN, San PP, Li XL, Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. In: IJCAI. Buenos Aires, Argentina, pp 25–31
3. Bianchi V, Bassoli M, Lombardo G, Fornacciari P, Mordonini M, De Munari I (2019) IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment. IEEE Int Things J 6(5):8553–8562
4. Chen Y, Zhong K, Zhang J, Sun Q, Zhao X (2016) LSTM networks for mobile human activity recognition. In: Proceedings of the 2016 international conference on artificial intelligence: technologies and applications, advances in intelligent systems research, ISBN-978-94-6252-162-9
5. Almaslukh B (2017) An effective deep autoencoder approach for online smartphone-based HAR. Int J Comput Sci Network Secur 17
6. Abu Alsheikh M, Selim A, Niyato D, Doyle L, Lin S, Tan H (2016) In AAAI workshops. Retrieved https://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12627/12337
7. Wang H, Zhao J, Li J, Tian L, Tu P, Cao T, An Y (2020) Wang: wearable sensor-based human activity recognition using hybrid deep learning techniques. Secur Communun Networks 2020:2132138
8. Siraj MS, Ahad MAR (2020) A hybrid deep learning framework using CNN and GRU-based RNN for recognition of pairwise similar activities. https://doi.org/10.1109/ICIEVicIVPR48672.2020.9306630
9. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. arXiv preprint arXiv:1706.03762
10. Sun B, Liu M, Zheng R, Zhang S (2019) Attention-based LSTM network for wearable human activity recognition, pp 8677–8682. https://doi.org/10.23919/ChiCC.2019.8865360
11. Luptáková I, Kubovčík M, Pospíchal J (2022) Wearable sensor-based human activity recognition with transformer model. Sensors 22:1911. https://doi.org/10.3390/s22051911
12. Kim Y-W, Cho W-H, Kim K-S, Lee S (2022) Inertial-measurement-unit-based novel human activity recognition algorithm using conformer. Sensors 22:3932. https://doi.org/10.3390/s22103932
13. Pan J, Hu Z, Yin S, Li M (2022) Sensor-based human activity recognition. Electronics 11:1797
14. Jian Q, Guo S, Chen P, Wu P, Cui G (2021) A robust real-time human activity recognition method based on attention-augmented GRU, pp 1–5. https://doi.org/10.1109/RadarConf2147009.2021.9455322
15. Wang K, He J, Zhang L (2019) Attention-based convolutional neural network for weakly labeled human activities recognition with wearable sensors
16. Dhall A, Subramanian R, Sharma G (2022) A transformer based approach for activity detection
17. Liu YY, Liu JW (2022) The time-sequence prediction via temporal and contextual contrastive representation learning. In: Khanna S, Cao J, Bai Q, Xu G (eds) PRICAI 2022: trends in artificial intelligence. PRICAI 2022. Lecture notes in computer science, vol 13629. Springer

18. Le T-H, Tran T-H, Pham C (2022) Human action recognition from inertial sensors with transformer. In: 2022 international conference on multimedia analysis and pattern recognition (MAPR), pp 1–6. https://doi.org/10.1109/MAPR56351.2022.9924794
19. USC-HAD Dataset. http://sipi.usc.edu/HAD/
20. Weiss GM, Yoneda K, Hayajneh T (2019) Smartphone and smartwatch-based biometrics using activities of daily living. IEEE Access 7:133190–133202
21. Sowmiya S, Menaka D (2022) A hybrid approach using bidirectional neural networks for human activity recognition, third international conference on intelligent computing instrumentation and control technologies (ICICICT), pp 166–171. https://doi.org/10.1109/ICICICT54557.2022.9917906

Design and Development of a Chatbot for Personalized Learning in Higher Education

Hayder Kareem Algabri, Rajanish K. Kamat, Kabir G. Kharade, and Naresh Babu Muppalaneni

Abstract Technology now plays a more significant role in our everyday lives, impacting how students learn and gain knowledge. The use of artificial intelligence allows educators to provide students with a personalized learning environment. Systematically detecting whether or not students are grasping the material has been created. Based on the feedback, we can also determine the student's starting point. Conversational agents or chatbots could be used to supplement teacher instruction and reduce the costs of informal education. The development and training of an intelligent conversational agent have long been a goal of humanity and a significant challenge for scientists and programmers. It was still just a dream until recently. On the other hand, language teachers have a new hope thanks to cutting-edge technologies like deep learning and neural networks, which provide intelligent chatbots that can learn through communication just like humans. His research aims to create an emotionally realistic chatbot system using artificial intelligence to improve the chatbot's believability. This study aims to create a chatbot that can be integrated into a personalized teaching–learning process. The goal of the study is to determine the efficacy of the developed chatbot in personalized learning. Its developed system enables it to provide a suitable personalized learning solution.

Keywords AI · Chatbot · Conversational agent · Personalized learning

H. K. Algabri (✉) · K. G. Kharade
Department of Computer Science, Shivaji University, Kolhapur, Maharashtra, India
e-mail: 7ayder.kareem@gmail.com

K. G. Kharade
e-mail: kgk_csd@unishivaji.ac.in

R. K. Kamat
Dr. Homi Bhabha State University, Mumbai, Maharashtra, India
e-mail: rkk_eln@unishivaji.ac.in

N. B. Muppalaneni
Department of CSE, Indian Institute of Information Technology Design and Manufacturing, Kurnool, Andhra Pradesh 518008, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_22

1 Introduction

Personalized learning is an approach in which specific elements, such as a student's abilities, strengths, needs, interests, and other factors, are considered when designing their learning experience. The main goal of this strategy is to help students do better in school. AI (artificial intelligence) is advancing rapidly and improving how various industries, such as healthcare, banking, energy, and retail, carry out their business. One sector in particular where AI technologies could have a significant impact is education. Culture shifts are among the fundamental challenges that colleges and universities are facing today, including disengaged students, high dropout rates, and the inability to apply a "one-size-fits-all" model to education. Every student would enjoy a unique educational approach tailored to their abilities and needs, which is only possible through personalized learning [2]. Increasing motivation could directly affect student persistence and reduce the likelihood of them dropping out. As a result, professors might be able to teach better students based on their unique learning methods. As new research becomes available, educators all over the country adopt new teaching methods, techniques, and practices. The education system has evolved into a learning experience in which both students and teachers can have a hands-on approach. Every student has different learning styles, so teachers must be willing to adapt their teaching methods to meet the needs of each of their students. Another positive aspect of NLP is how well it fits into modern learning methods, and this is why so many teachers use it in their classrooms. We examine NLP, particularly as it relates to learning and education. Students must realize that over time, increasingly routine and repetitive tasks will be automated, performed by robots, or automated. Cognitive, emotional, and creative roles will exist as long as positions require each of these qualities. Many universities across the globe are failing to adequately prepare students for the future jobs that they will have. The best results will combine AI and human abilities as artificial intelligence is applied to education. We will never be able to perform these tasks without the use of humans. While we cannot underestimate the value of human interaction and critical thinking in education, teachers play a crucial role in our society. We must never underestimate the importance of such interaction and reflection in education [3]. AI and chatbots have the potential to transform certain practices in teaching and learning. The application of artificial intelligence and chatbots has the potential to revolutionize particular teaching and learning practices. As AI-related technologies, for example, can support hybrid courses that mix face-to-face classroom experiences with self-directed student learning in the online environment, they can aid in the education of the hybridized course. Personalized learning made possible by artificial intelligence uses data analysis to produce individualized learning experiences that best suit each learner's unique needs and learning style. Programs that utilize artificial intelligence can help guide an undergraduate student with difficulty with a concept or lesson and help her locate additional educational resources [11]. As AI frees up professors to focus on more essential tasks, this is one of several ways it can happen. Faculty can use class time to shore up knowledge, connect the lessons to bigger concepts, or focus on collaborative peer learning by

having students learn outside of the classroom using artificial intelligence tools. AI applications in class can also carry out other administrative functions. Several organizations have started to utilize AI to assist with tests and even grade assignments for students [12].

2 Related Work

When it comes to evaluating the effectiveness of a chatbot based on its interaction with a user, the results are often highly subjective. Depending on the interaction's context, domain, and nature, a different measure may be used in different situations. As a result, benchmarks or specific standards must be developed to evaluate chatbots consistently [21]. These technologies must work together seamlessly behind the facade of the chatbot to provide such a natural flow of messaging conversation that the user truly does not realize they are talking to a machine [22]. Naive Bayes is a reasonable classifier in this sense because it requires little storage and is quick to train. It is often used in time-sensitive applications such as automatically classifying web pages into types and spam filtering, which need minimal storage and training time [6].

2.1 Chatbot in Education and Their Importance

Using AI-based chatbots in educational institutions offers many apparent benefits. Some of them are as follows:

(a) **Tutoring made easy**: Today's most remarkable thing in the education industry is Intelligent Tutoring Systems' content. Personalized learning is here today. They are assessing student progress and responses and looking at their learning process to help to personalize learning. A good example is when teachers use chatbots to build conversation by breaking up long lectures into short chains of messages. Chatbots can assess the learning level of students, as well as the rate of advancement. In the history of teaching, technology has never made it this simple [6].

(b) **Students can get instant help**: We live in a world where the pace never stops. Students are now able to obtain anything instantly because of technology. Any of these can be done in a few simple clicks: sending an email, posting a picture, searching a place, or even finding assignment help. To increase the likelihood of drawing this generation's attention, educational institutions must ensure they can talk to students quickly [5].

(c) **Engaging students effectively**: In academic assignments, students prefer social media channels such as Facebook and Twitter to communicate, research topics, and search for relevant solutions. As a result, learning does not have to be limited

to the classroom. Finding out about lectures, assignments, and other important events is a great way to save time. Alumni groups and activity clubs can even be established for sharing information.

(d) **Engagement of students**: Today's students are already used to messaging platforms like WhatsApp and Facebook Messenger. They use platforms like these to research topics, contact each other, or ask for assistance in getting assignments done. It could be utilized to enhance student engagement and overall understanding of a subject [7].

(e) **Simple Access to Information**: Educational bots provide simple mechanisms fochers. It helps strengthen your online presence. Students can search your website, ask any question, and receive an immediate answer. In that case, chatbots help learners obtain instant access to information. In the education industry, access to the correct data is critical. Students must acquire proper learning and knowledge, both of which are necessary. Chatbots help targets their customers and provides accurate information at the most suitable time. Even people contacting administrators for solutions cannot reach that quick [2].

(f) **Teaching Assistants who are efficient**: Students often put "do my assignment requests" over the internet to find someone to assist them in completing the assignments and clearing their doubts. Similarly, our modern-day teachers also require assistance to simplify the hectic schedules. Nowadays, bots are being used as virtual teaching assistants to do the repetitive tasks of the teachers.

(g) **Learning through chatbots**: An intelligent tutoring system provides a dedicated learning environment. Students dedicate more time to studying and form close connections with their peers in an environment like this. For teaching, chatbots use a library of training messages and in-depth conversations. To analyze information, assess a student's understanding level, and provide lectures, chatbots analyze every piece of information, consider the student's personal needs, and go over concepts and concepts in detail [8].

(h) **Smart and secure feedback**: It is crucial to provide feedback to students to improve their learning. By analyzing feedback from students, teachers can identify areas where they need to improve. Giving feedback on teachers' areas of instruction helps students identify areas where they need to put in more effort. Assignments, assessments, and tests are quickly supplied to the students and excellent feedback from the teachers. The majority of educational institutions have online to printed forms to collect student feedback [20].

(i) **Learning with spaced intervals**: With the help of different assumptions, you have developed an online course for a specific group of people. Here are the men and women who look after their personal and professional lives. Instead of working hard to pass the course, they attempt to complete it however they can, which is good enough to earn the credits hours and pass the course. Concentration is fostered when learning breaks lengthy periods into smaller chunks. It is used to assist learners with recalling what is taught in the system. Learners can memorize classes for a longer time when they use spaced interval learning. As a result, more engaged learners [10].

(j) **Proctoring**: It comes with its benefits and pitfalls when it comes to online education. It's an interactive and fun way of learning for some students. Some people, however, do not even think twice about cheating the system. It tends to happen when you are learning on your own. Frauds in the examinations are detected and stopped by the use of AI-based online proctoring facilities. Students are no longer allowed to attempt any form of cheating when administering examinations. To earn points, they have to make them properly [19].

(k) **Test Scoring**: Tutors can use chatbots to practice tests and check answers. Results are turned into points as soon as they are turned into results. Teachers do not have to worry about plagiarism and grammar as well because of modern technology. This bot can also do that. It corrects grammar, typos, and image duplication, as well as a lack of proper referencing.

(l) **It allows the teacher to reduce the time**: The time required to organize and execute tasks has been reduced since chatbots provide immediate answers to students' frequently asked questions. By saving time, students can devote it to research or projects pending for the course and the supervision and motivation of their peers [17].

(m) **Store and analyze data effectively**: When conducting a review of students' evaluation and progress. Due to the utilization of AI, students can allocate their time effectively and apply it to the things they need to accomplish in an accessible manner.

(n) **Improves access to education**: It is designed to help students develop and interact without considering various resources, like textbooks, equipment, etc., the language, or the student's location. The movement is very much in line with the idea of democratizing learning [18].

(o) **Personalization in education**: Victor Garcia Hoz popularized the concept of personalization in 1970. Since then, the word has acquired a wide range of definitions. Some teachers try to tailor their lessons to the individual needs of their students. In other words, others emphasize the students' most significant potential for development. According to Sir Ken Robinson, a guru in the education world, personalized learning is the process of tailoring learning to individuals based on their unique strengths, weaknesses, interests, and learning styles. Individualization and differentiation are not the same things as personalization. It may be the case that the current lack of a consistent definition for personalization is behind the fact that today no widely accepted theory of personalization exists [9].

3 Future of Chatbot

Essentially, AI promises to help humanity more deeply think and reason and increase the pace of discovery by leveraging technology. According to some experts, AI will play a similar role in the information age as the industrial revolution did in the

manufacturing industry. AI analysis will enhance the search for student success and the growth of more advanced business intelligence and operational processes in the near term. AI may eventually do more. The use of artificial assistants in the design of textbooks, delivery of course content, development of test questions, evaluation of answers to them, and monitoring of online discussions are all conceivable ways of employing this kind of technology. Although universities will design educational programs to ensure their graduates can compete in a rapidly changing workplace, AI is also likely to drive the design of new courses and even majors in the future [18].

3.1 Types of Chatbots

(a) **Scripted chatbots**: Chatbots that behave according to rules are considered to be bots. There are predefined paths that conversations with this kind of chatbot must follow. When using a chatbot, the user must choose from explicit options to find the next step at each conversion step. It will depend on the features of the chat platform that the user is using and the design of the bot whether the user will be required to choose between text, voice, or touch to communicate [4].
(b) **Intelligent chatbots**: A chatbot is built using artificial intelligence techniques. Artificial intelligence is more flexible in terms of what it can take from the user. You can enter free-form text or voice input into the form (but of course, they are not limited to other forms of information if that makes sense). With AI, they can constantly get better as more times they are used. For simple, one-time tasks, artificial intelligence works very well. On the other hand, though, the actual intelligence of the bot is minimal. Bots cannot "understand" context or ambiguity, and they cannot remember over time.
(c) **Application chatbots**: Depending on the nature of the chatbot, users may be exposed to graphical user interfaces (GUIs) in scripted and intelligent chatbots. The two main categories of chatbots are intellectual and scripted. There is no such thing as application bots as a separate category of bots. An essential concept for chatbot developers is that bots can be interacted with using a GUI. The bot should display a GUI to complete the task efficiently via a graphical interface [16].

4 Proposed System

Conversational interfaces have evolved out of the "chat robot" idea, known as chatbots. They allow for highly engaging, personal interactions, whether in voice and text conversations on smartphones, browsers, or popular chat platforms such as Facebook Messenger or Slack. The usage of voice recognition and natural language processing for chatbots like personal assistants and assistants in call centers are made possible by advances in deep learning technologies such as text-to-speech, automatic speech

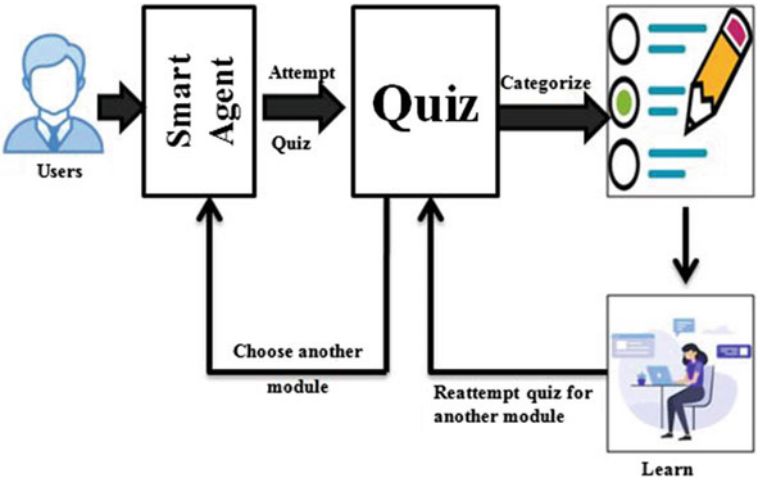


Fig. 1 Workflow of the developed system

recognition, and natural language processing. Programs of this type are called "bots." A chatbot is programmed with artificial intelligence to simulate conversation with a user through speech or text. Virtual assistants like Amazon's Alexa and Apple's Siri are notable examples. There are two other hypothetical examples, and one is an automated chat that directs customers to the products on a website. Conversational bots have improved at answering user questions by programmatically preparing predefined responses and enhancing responsiveness [13].

AI holds the potential to improve and streamline many back-office processes, including human resources, IT, business operations, research, and records management. Customer service can improve when AI chatbots field routine questions around the clock, freeing employees for other tasks. AI is being used in educational facilities to improve operational effectiveness and operational efficiency. To process volumes of data, discover operational efficiencies, maximize the use of space, maintain security, and optimize facility maintenance, HVAC and lighting systems linked through the Internet of Things have a high capacity for data processing. At UT Austin, AI controls the university's sprinkler system, saving both water and money. Santa Clara University employs artificial intelligence to aid in the management of campus parking. On Michigan's campus, numerous universities are conducting tests on driverless cars (Fig. 1) [14].

4.1 AI Techniques to Train the Chatbot

(a) When training a chatbot, it is usual practice to start with a wish list of things that you would like your bot to be able to do. However, it is vital to start with

a specific business problem that your bot will be created to tackle. This should be the starting point.

(b) Create very specific intents that only serve one specific goal in order to prevent the user experience from becoming annoying and to correctly understand how to teach a chatbot.

(c) An ability to make your AI chatbot usable is directly proportional to how accurately the sample utterances resemble how language is actually used in the real world.

(d) Use a wide variety of expressions to activate each intent while you are developing and testing the application.

(e) It is more likely that a diverse team will ask questions in a variety of different ways. This is an essential part in training chatbots.

(f) After you have composed a number of utterances, you should make a note of the words or phrases that represent key information about the variable. These are going to turn into the entities.

4.2 Features of a Chatbot

(a) **Automation leads to efficiency**: Chatbots can assist in simplifying processes by automating steps within complex procedures through a few simple voices or text requests, which cuts the time it takes to complete processes and boosts business efficiency.

(b) **It is flexible**: Both voice and text chatbots can be built to respond to the user's language of choice. Chatbots can be embedded in workflows to communicate with employees and consumers.

(c) **Broader Customer Engagement**: A superior customer experience can distinguish a business from the competition. The great thing about chatbots is that they can be installed in channels where your customers and prospects are already engaged, like Facebook Messenger, where you can reach them more quickly and fulfill their needs [18].

(d) **Well-trained FAQs**: Chatbots can become more powerful through consistent training, which also enables them to more easily handle questions and interactions with users.

(e) **Saves Time**: Chatbots enable businesses to become more efficient and save time by providing answers to fundamental questions. This is one of the many benefits of using chatbots. The members of the support team who are executives are only notified of the most difficult questions that require human intervention.

5 Result and Discussion

See Figs. 2, 3 and 4.

We have designed a chatbot so that the user has to log in to the system first. Users can select the subject based on their preference. After that, a short quiz will be given to discover the level of knowledge of the issue. Depending on the user's story, the content will be delivered. The user will be categorized into three groups. Beginners, intermediates, and experts make up these levels. Each student must choose the module of the selected subject based on their level. Students have to complete the module after choosing it. In the end, they can decide whether to go on to the next module or to repeat the module. Ultimately, students will be able to learn at their own pace and according to their interests.

5.1 Limitations and Future Enhancements

The developed chatbot needs a proper training. If it is trained in wrong way, it is difficult to train. The training requires sufficient time and efforts. The present chatbot

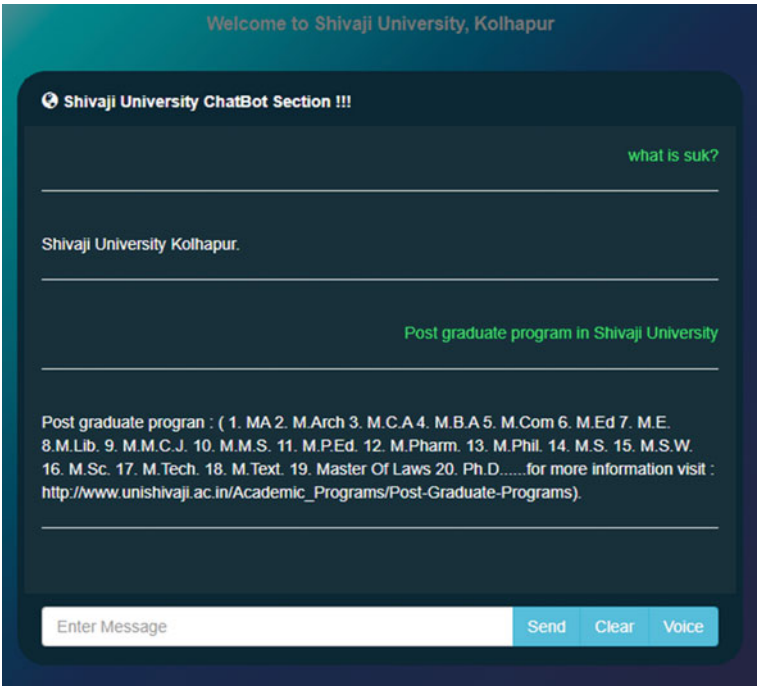


Fig. 2 Developed chatbot for a customized learning Initial screen

Fig. 3 Developed chatbot for customized learning with only keywords

deals with queries related to FAQs of Shivaji University, Kolhapur, as it is designed based on their problems. But, the same can be modified to provide solutions to all the problems of any university in India. Also it can be used to provide solutions in school education system as well.

Fig. 4 Developed chatbot for customized learning with sentences

6 Conclusion

While personalization of learning has been embraced as a significant effort and strategy across the modern education system, its widespread use has been limited. It seems that governments and policymakers favor personalized learning models over educators and researchers. To encourage teachers and researchers to share their concerns and seek solutions, we must consider the lack of interest. The present paper focuses on advanced chatbot helps in personalized learning. Using such a chatbot, the student can get the data based on their previous knowledge. The ultimate aim of this chatbot is to provide content as per their interest and prior knowledge. A personalized learning environment tailors instruction according to each student's readiness, strengths, needs, and interests. Individualized learning environments contain a range of teaching methods such as competency-based progression, performance-based assessments, and individualized learning plans.

References

1. Algabri HK (2020) Curriculum technology integration for higher education. J Adv Res Dyn Control Syst 12(1):295–300. https://doi.org/10.5373/JARDCS/V12I1/20201043
2. Algebri HK, Husin Z, Abdulhussin AM, Yaakob N (2017) Why move toward the smart government. In: 2017 international symposium on computer science and intelligent controls (ISCSIC), pp 167–171. https://doi.org/10.1109/ISCSIC.2017.34
3. Atkinson S (2006) Factors influencing successful achievement in contrasting design and technology activities in higher education. Int J Technol Des Educ 16(2):193–213
4. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A (2020) A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing 408:189–215. https://doi.org/10.1016/j.neucom.2019.10.118
5. Dokukina I, Gumanova J (2020) The rise of chatbots—new personal assistants in foreign language learning. Proc Comput Sci 169:542–546
6. Kaviani P, Dhotre S (2017) Short survey on Naive Bayes algorithm. Int J Adv Eng Res Develop 4(11):607–611
7. Kharade KG, Kamat RK, Kharade SK (2019) Online library package to boost the functionality and usability of the existing libraries. Int J Future Revolut Comput Sci Commun Eng 5(8):5–7
8. Kharade KG, Kharade SK, Katkar SV (2019) Cyber security-a method of generic authentication of data with IP security. Int J Inf Syst 9(2):63–65
9. Kharade SK, Kharade KG, Kamat RK, Kumbhar VS (2020) Setting barrier to removable drive through password protection for data security. Our Heritage 68(27):19–23
10. Patel H, Prajapati P (2018) Study and analysis of decision tree based classification algorithms. Int J Comput Sci Eng
11. Patil BP, Kharade KG, Kamat RK (2020) Investigation on data security threats and solutions. Int J Innovative Sci Res Technol 5(1):79–83
12. Pradubthong N, Petsangsri S, Pimdee P (2018) The effects of the SPACE learning model on learning achievement and innovation and learning skills in higher education. Mediterr J Soc Sci 9(4):187–199. https://doi.org/10.2478/mjss-2018-0128
13. Rouhiainen L (n.d.) How AI and data could personalize higher education. Retrieved July 27, 2021, from https://hbr.org/2019/10/how-ai-and-data-could-personalize-higher-education
14. Shemshack A, Spector JM (2020) A systematic literature review of personalized learning terms. Smart Learn Environ 7(1):33. https://doi.org/10.1186/s40561-020-00140-9
15. Singh R (2018) AI and chatbots in education: what does the futurehold? https://chatbotsmagazine.com/ai-and-chatbots-in-education-what-does-the-futurehold-9772f5c13960
16. Smutny P, Schreiberova P (2020) Chatbots for learning: a review of educational chatbots for the facebook messenger
17. Vanichvasin P (2021) Chatbot development as a digital learning tool to increase students' research knowledge. Int Educ Stud 14(2):44. https://doi.org/10.5539/ies.v14n2p44
18. Vijayaraghavan V, Cooper JB, J RL (2020) Algorithm inspection for chatbot performance evaluation. Proc Comput Sci 171:2267–2274. https://doi.org/10.1016/j.procs.2020.04.245
19. Zhang L, Basham JD, Yang S (2020) Understanding the implementation of personalized learning: a research synthesis. Educ Res Rev 31:100339. https://doi.org/10.1016/j.edurev.2020.100339
20. Zhang Y (2012) Support vector machine classification algorithm and its application. In: Liu C, Wang L, Yang A (eds) Information computing and applications, vol 308. Springer, Berlin, Heidelberg, 179–186. https://doi.org/10.1007/978-3-642-34041-3_27
21. V, V, Cooper JB, J RL (2020) Algorithm Inspection for chatbot performance evaluation. Proc Comput Sci 171:2267–2274
22. Chougale NP, Kharade KG, Kharade SK, Ghatage SR, Mendagudli MG, Yuvaraj S, Vengatesan K (2021) Deployment of Computer assisted instruction in higher educational organization. In: Recent trends in intensive computing. IOS Press, pp 461–465

Performance Evaluation of Concentric Hexagonal Array for Smart Antenna Applications

Sridevi Kadiyam and A. Jhansi Rani

Abstract Many of the antenna array geometries considered so far are linear and circular arrays. In this paper, a different geometrical array, Concentric Hexagonal Array (CHA) is reported for the beamforming of Smart Antenna applications. The advantage of this array over other geometrical configurations is it gives less side lobe level (SLL) and also it achieves pattern symmetry for the non-uniform amplitude excitation. Since the radiation pattern becomes symmetric, and the element's amplitude carry even symmetry about the array's center, the computing time and the number of attenuators turn to be halved for the amplitude—only synthesis which in turn reduces the cost of the system. Using the 12-element hexagonal array, a 24-element and 36-element concentric hexagonal array is constructed and their performances are evaluated. The 24-element CHA allows less side lobe level as with 36-element. The performance of 24-element is also analyzed with circular and hexagonal arrays with same elements. Firefly algorithm is used for beamforming and also to obtain the excitation coefficients.

Keywords Side lobe level · Concentric hexagonal array · Firefly algorithm

1 Introduction

In antenna arrays to determine the unique radiation pattern, there are five important factors to consider. They are as follows:

- The array geometry
- Element distance

S. Kadiyam (✉)
GITAM University, Visakhapatnam, India
e-mail: skadiyam@gitam.edu

A. J. Rani
Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India
e-mail: jhansirani@vrsiddhartha.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_23

- The individual element's amplitude excitation
- The individual element's phase excitation
- The individual element's relative pattern.

The array geometries which were considered previously are uniform linear arrays (ULA) [1–8], uniform rectangular array (URA) [4, 9], and uniform circular array (UCA) [5, 6, 10–13]. As the circular array has high side lobe level, a uniform hexagonal array (UHA) [10, 11, 14, 17] were designed. The concept of hexagonal antenna array is introduced in [15]. Concentric arrays are also used to overcome high side lobe level [16]. Concentric arrays such as planar uniform circular arrays (PUCA) [10, 12, 16] and concentric hexagonal array (CHA) [11, 12] are implemented for SA applications.

Using the UHA configuration, CHA is constructed with 24 isotropic elements. The investigation reveals that CHA gives less SLL and also achieves symmetry in the radiation pattern as compared to the circular and hexagonal array.

2 Methodology

2.1 Concentric Hexagonal Array (CHA)

CHA is depicted in Fig. 1. It is developed in [18, 21, 22].

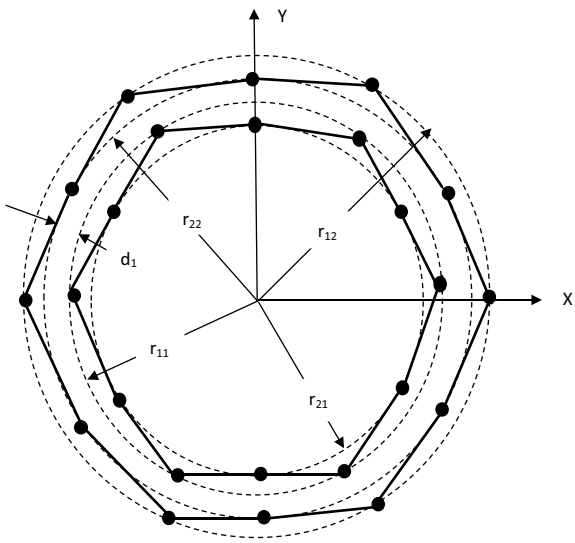


Fig. 1 CHA with 12 elements on each hexagon arrangement

2.2 *Firefly Algorithm*

Firefly algorithm (FFA) [19, 20] is a novel nature-encouraged algorithm. From the bioluminescence process, flashing light of fireflies is produced upon which FFA is based. The basic stepladders of FFA are the following:

Step I. Initialization of populace is the first step which initializes 'm' number of fireflies in N-dimensional space within the boundary [19, 20].
Step II. The brightness of each firefly can be calculated by the cost function at their current location.
Step III. Update firefly's locations based on Eqs. (6) and (7) [19, 20]

$$x_m = x_m + \beta_0 e^{-\Upsilon r_{m,n}^2 (x_m - x_n)} + \alpha\left(\text{rand} - \frac{1}{2}\right) \tag{1}$$

$$r_{\text{mn}} = x_m - x_n = \sqrt{\sum_{k=1}^{d} \left(x_{m,k} - x_{n,k}\right)^2} \tag{2}$$

Step IV. Rank all fireflies and estimate the current global best.
Step V. Terminate the condition when the desired cost function is reached.

2.3 *Objective Function*

It is given by the Eq. (3) [6, 10, 11, 21, 22].

$$\text{Objective function} = \sum_{i=1}^{N} a_i G(\theta_i) - \sum_{j=1}^{M} b_j G(\theta_j) \tag{3}$$

Notice that the objective function of Eq. (8) does not make a constraint on side lobe level as well as null depth.

3 Results and Discussions

The populace = 20 and light absorption coefficient, (Υ) = 1, randomization parameter ($\propto$) = 0.2, and attractiveness (β) = 0.2 for maximum number of 100 iterations. For 24-element CHA, the SOI is at 0° and two SNOIs at 50° and 55°. It is compared with the UCA and hexagonal array of the same number of elements.

Figures 2 and 3, respectively, describe the radiation patterns of CHA, UHA, and UCA of 24 elements at the interferences 50° and 55°. It is concluded from the figures that the radiation pattern of CHA becomes symmetric, because of the element amplitudes carry even regularity about the array's center; the number of attenuators and the

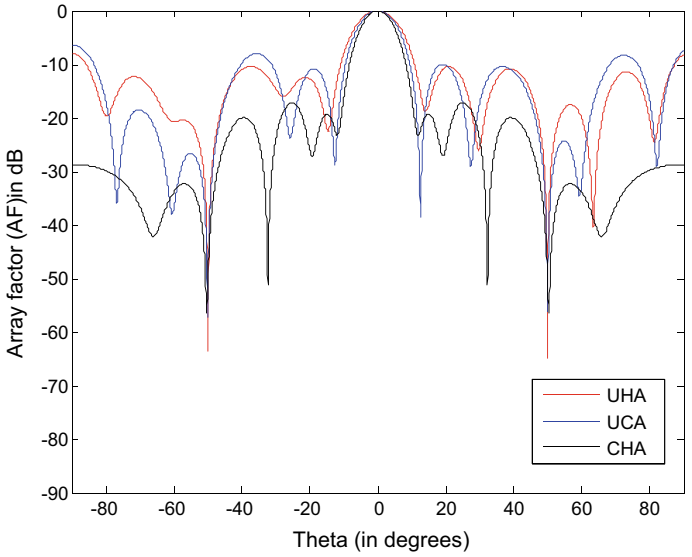


Fig. 2 The three configuration's radiation pattern for intended user is 0° and interference is 50°

computing time became halved. The excitation amplitudes are shown in Tables 1 and 2 for the interferences of 50° and 55°, respectively. From the Table.3, it is concluded that CHA gives less SLL of -19.36 dB at 50° and -13.61 dB at 55° interferences, respectively, which is less as compared to UCA and UHA.

In Eq. (2), if $M = 3$, then CHA becomes 36-element array as shown in Fig. 4. The Fig. 5 shows the radiation pattern of 36-element CHA at interference of 40° in which SLL obtained is -11.56dB and HPBW is 4.6°. Figure 6 illustrates the radiation pattern of 36-element CHA at an interference of 65° in which SLL obtained is -11.56dB and HPBW is 5°. Table 4 shows the excitation coefficients at 65° and 40° of interferences. It is concluded that among the 24-element and 36- element CHA, 24-element CHA gives better SLL.

4 Conclusions

UCA, UHA, and CHA are examined, and the prime issue related to the SA for beamforming is explored with the equal number of elements using the FFA. CHA gives a better side lobe level range of -13.61 dB and -19.36 dB at 55° and 50° interferences, respectively, as compared to UCA and UHA, with satisfactory null depths and good directional pattern has been obtained. For CHA, radiation pattern becomes symmetric, and the element amplitudes carry even regularity at array's center, and the number of attenuators and the computing time becomes halved. 24-element CHA gives better SLL as compared to 36-element CHA.

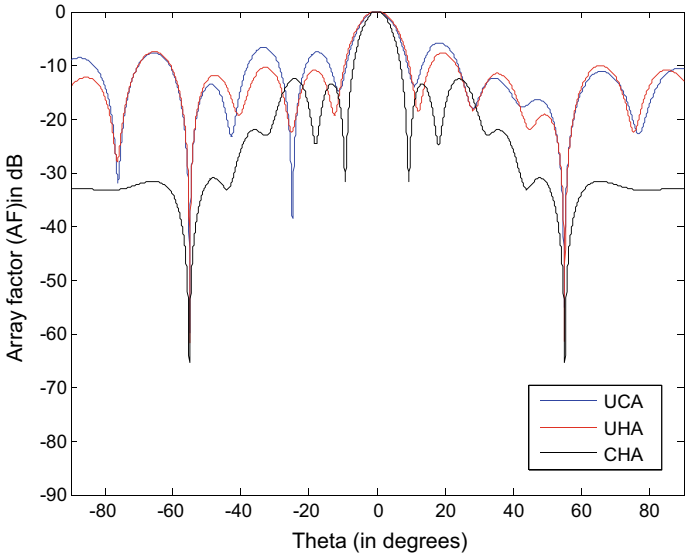


Fig. 3 The three configuration's radiation pattern for intended user is 0° and interference is 55°

Table 1 24 element's amplitude excitation coefficients of different array types at an interference of 50°

Element No.	UCA	UHA	PUHA
#1	0.4399	0.2259	±0.8841
#2	1.000	0.9094	±0.8589
#3	0.3457	0.6428	±0.663
#4	0.0743	0.6148	±0.1015
#5	0.0963	0.7792	±0.0572
#6	0.444	0.5826	±1.000
#7	0.057	0.1802	±0.4357
#8	0.1661	0.5149	±0.9148
#9	0.0022	0.398	±0.2132
#10	0.1528	0.1425	±0.2056
#11	0.7241	0.9371	±0.998
#12	0.6806	0.9949	±0.1631
#13	0.2924	0.2662	Because of the symmetry, excitation coefficients are existed for only 12 elements
#14	0.7916	0.8454	
#15	0.4723	0.5011	
#16	0.0023	0.074	
#17	0.7896	0.4069	
#18	0.6202	0.0409	
#19	1.000	0.0118	
#20	0.1416	0.0667	
#21	1.000	0.423	
#22	0.9594	0.0072	
#23	0.9713	0.9946	
#24	0.8713	0.0247	

Table 2 24 element's amplitude excitation coefficients of different array types at an interference of 55°

Element no.	UCA	UHA	PUHA
#1	0.9986	0.4601	± 0.7362
#2	0.788	0.7249	± 0.1302
#3	0.6101	0.1865	± 0.4844
#4	0.9478	0.2451	± 0.2119
#5	0.191	0.1486	± 0.8366
#6	0.5299	0.1686	± 0.4898
#7	0.121	0.2247	± 0.8921
#8	0.0187	0.1435	± 0.3909
#9	0.856	0.0076	± 0.9311
#10	0.3083	0.4819	± 0.4546
#11	0.4261	0.3716	± 0.2962
#12	0.2605	0.6446	± 0.2269
#13	0.7032	0.5741	Because of the symmetry, excitation coefficients are existed for only 12 elements
#14	0.5764	0.3372	
#15	0.3942	0.4475	
#16	0.1898	0.7786	
#17	0.9938	0.0472	
#18	0.0165	0.3965	
#19	0.0527	0.1764	
#20	0.2132	0.0583	
#21	0.6259	0.0687	
#22	0.2703	0.2893	
#23	0.9941	0.2073	
#24	0.8713	0.0247	

Table 3 SLL, Null depth and HPBW comparison of three configurations

Parameter	UCA		UHA		CHA	
	SNOI at 55°	SNOI at 50°	SNOI at 55°	SNOI at 50°	SNOI at 55°	SNOI at 50°
SLL in dB	−7.6	−10.87	−11	−12.29	−13.61	−19.36
Null depth in dB	−59.74	−57.12	−61.74	−63.47	−65.35	−56
HPBW in degree	10.7	11.4	11.5	12.9	8.5	9.7

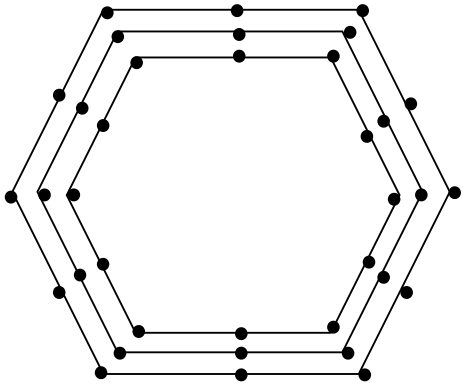


Fig. 4 PUHA of 36 element

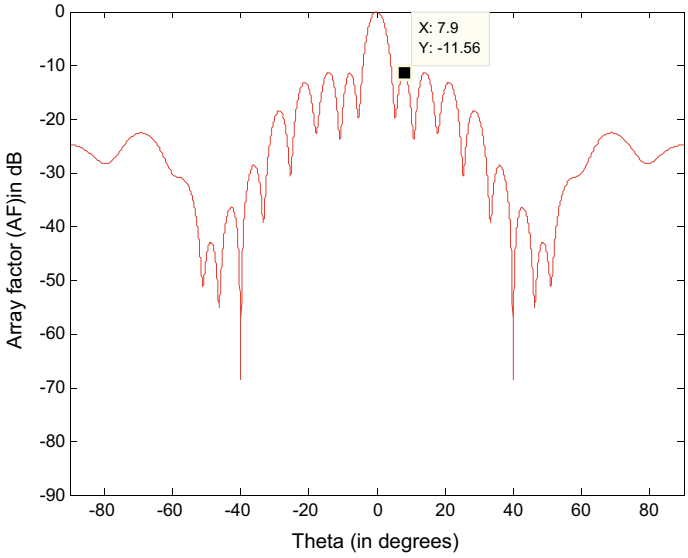


Fig. 5 36-element radiation pattern for intended user is 0° and interference is 40°

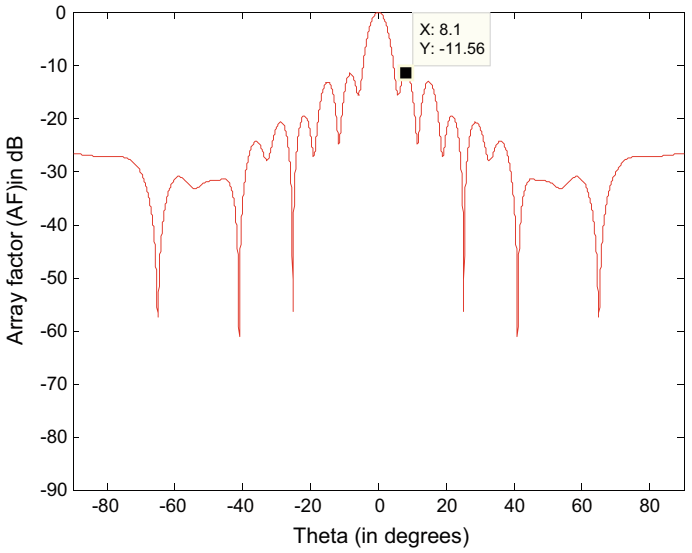


Fig. 6 36-element radiation pattern for intended user is 0° and SNOI is 65°

Table 4 Excitation coefficients of 36-element CHA at an interference at 65° and 40°

Element No.	At 65° interference	At 40° of interference
1	±0.5968	±0.502
2	±0.434	±0.8871
3	±0.0753	±0.6796
4	±0.7145	±0.373
5	±0.348	±0.918
6	±0.4713	±0.8072
7	±0.725	±0.7861
8	±0.9093	±0.9703
9	±0.6508	±0.66
10	0.7971	0.000
11	±0.558	±0.45
12	±0.6914	±0.6644
13	±1.000	±0.221
14	±0.533	±0.873
15	±0.297	±0.2644
16	±0.3807	±0.6635
17	±0.6635	±0.1791
18	±0.1791	±0.8677

References

1. Mouhamadou M, Vaudon P (2006) Smart antenna array pattern synthesis: null steering and multi user beamforming by phase control. Progress Electromagn Res PIER 60:95–106
2. Zuniga V, Erdogan AT, Arslan T (2010) Adaptive radiation pattern optimization for antenna arrays by phase perturbations using particle swarm optimization. In: NASA/ESA IEEE conference on adaptive hardware and systems, 978–1–4244–5889–9/10
3. Wang Y, Gao S, Hang Y, Tang Z (2011) Synthesis of antenna array by complex valued genetic algorithm. Int J Comput Sci Netw Secur 11(1):91–96
4. Raoa AP, Sarma NVSN (2014) Performance analysis of differential evolution algorithm based beamforming for smart antenna systems. I.J. Wireless Microwave Technol 4(1)
5. Singh U, Rattan M (2014) Design of linear and circular antenna arrays using cuckoo optimization algorithm. Progress Electromagn Res C 46:1–11
6. Magdy A, El-Ghandour OM, Hamed HFA (2015) Adaptive beamforming optimization based Hybrid PSOGSA algorithm for smart antenna system. In: Progress in electromagnetics research symposium proceedings, 973–97
7. Banerjee S, Dwivedi VV (2015) Linear array synthesis using schelkunoff polynomial method and particle swarm optimization. In: IEEE international conference on advances in computer engineering and applications (ICACEA)
8. Lakshmi MLM, Kamal KR (2016) Amplitude only linear array synthesis with desired nulls using evolutionary computing technique. ACES J 31(11)
9. Ioannides P, Balanis CA (2005) Uniform circular arrays for smart antennas. IEEE Antennas Propag Mag 47(4):192–206
10. Mahmoud KR, Eladawy MI, Bansal R, Zainud-Deen SH, Ibrahem SMM (2007) Analysis of uniform circular arrays for adaptive beamforming applications using Particle swarm optimization algorithm. https://doi.org/10.1002/mmce.20265
11. Mahmoud KR, El-Adawy M, Ibrahem SMM (2007) A comparison between circular and hexagonal array geometries for smart antenna systems using particle swarm optimization algorithm. Progress Electromagn Res PIER 72:75–90
12. Noordin NH, Zuniga V, El-Rayis AO, Haridas N, Erdogan AT, Arslan T (2011) Uniform circular arrays for phased array antenna. In: IEEE loughborough antennas and propagation conference, 978–1–4577–1016–2/11
13. Montaser AM, Mahmoud KR, Abdel-Rahman AB, Elmikati HA (2012) Circular, hexagonal and octagonal array geometries for smart antenna systems using hybrid cfo-hc algorithm. J Eng Sci Assiut Univ 40(6):1715–1732
14. Gozasht F, Dadashzadeh GR, Nikmehr S (2007) A comprehensive performance study of circular and hexagonal array geometries in the LMS algorithm for smart antenna applications. Progress Electromagn Res PIER 68:281–296
15. Kretly LC, Cerqueira Jr AS, Tavora AAS (2002) A hexagonal adaptive antenna array concept for wireless communication applications. In: The 13th IEEE International symposium on personal, indoor and mobile radio communications, vol 1, pp 247–249
16. Dessouky M, Sharshar H, Albagory Y (2006) Efficient sidelobe reduction technique for small—sized concentric circular arrays. Progress Electromagn Res PIER 65:187–200
17. Bera R, Lanjewar R, Mandal D, Kar R, Ghoshal SP (2015) Comparative study of circular and hexagonal antenna array synthesis using particle swarm optimization. Proc Comput Sci 45:651–660
18. Sridevi K, Rani AJ (2018) Investigation of hybrid hexagonal antenna arrays for smart antenna applications. In: Proceedings of the international conference on communication and electronics systems, IEEE Xplore Part Number: CFP18AWO-ART; ISBN:978–1–5386–4765–3
19. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T (eds) Stochastic algorithms: foundations and applications, vol 5792. SAGA 2009. Lecture Notes in Computer Science, Springer, Berlin, Germany, pp 169–178
20. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio-Inspired Comput 2:78–84

21. Sridevi K, Rani AJ (2018) Performance analysis of different array configurations for smart antenna applications using firefly algorithm. ARPN J Eng Appl Sci 13(22)
22. Sridevi K, Rani AJ (2022) Synthesis of the varients of adaptive hexagonal antenna array configuration. Lecture Notes Data Eng Commun Technol 126

A Comprehensive Study on Bridge Detection and Extraction Techniques

P. Rishitha, U. Venkata Sai, S. Dyutik Chaudhary, and G. Anuradha

Abstract Identification of bridges had a major role in providing the status of constructions. Generally, satellite images include information about geographical capabilities including bridges and those capabilities are very beneficial for military and civilian people. The identity of bridges in main infrastructure works is critical to offer data approximately the fame of those structures and guide feasible decision-making processes. Typically, this identity is achieved with the aid of using human marketers that need to hit upon the bridges into large-scale datasets, reading pictures with the aid of using pictures, a time-eating task. Bridge scrutinizing is vital to reducing the safety concerns caused by aging and deterioration of the bridges. Usually, there are traditional methods for the inspection and identification of bridges by using IOT sensors and lasers, but these can be identified only if the object is within the medium range of distance and at a minimum time, this can only detect them in succession. This identification can be done by using convolution neural networks and deep learning techniques. Also, the Geographic Information System helps to analyze, gather, capture, and manage geographical features. GIS is used to control and combine disparate assets of spatial and characteristic records for tracking bridge health.

Keywords Convolution neural network · Deep learning · Scrutinizing · Satellite images · Computer vision · Bridges · Geographic information system

P. Rishitha (✉) · U. V. Sai · S. D. Chaudhary · G. Anuradha
Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India
e-mail: ponnururishitha@gmail.com

G. Anuradha
e-mail: ganuradha@vrsiddhartha.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_24

1 Introduction

Bridges are one of the best ways of transportation. Bridges are divided into water bridges and avenue bridges. Bridges spanning rivers are typical man-made objects, and their automated recognition is critical for both military and civilian applications. Automatically recognizing bridges in high-resolution satellite images are used for storing geographical databases up to date, automatically identify the imaging satellite at a low cost, easily and quickly assessing the volume of harm with inside the occasion of herbal screw ups consisting of flooding or earthquakes. Most of the bridges are being detected using the segmentation techniques and they are broadly divided into several types, such as gray change-based [1, 2], differential image recognition-based [3, 4], and river centerline-based [5]. Normally, the progress of a building project is observed in person. But growing access to the latest technologies for aerial imaging has made large-scale remote data analysis possible.

Human agents are used to do bridge identification, which is a time-consuming activity that requires them to visually identify bridges in large collections by examining each image. Automatically extracting bridges is helpful in several crucial applications, including controlling an Unmanned Aerial Vehicle (UAV), retaining the geographic information, attacking a sure object, etc. Many resources have been used recently for the automatic extraction of artificial objects from aerial pictures, satellite pictures, and SAR pictures. To identify and assess the regular operation of roads and bridges in real time, an intelligent system for real-time bridge inspection must be established. The bridge's primary structure's bearing capacity, readiness for use, and durability are all important factors throughout the operation.

Yu and Nishio [6] proposed the multilevel structural components using the types of bridges and components of bridges and parts of bridges using CV Technology and CNN. This proposed method uses techniques like CNN model, ResNet-50, YOLOv3 model, and Mask R-CNN model. Yang et al. [7] proposes a Bridge on land Extraction Algorithm Using Deep Learning Techniques. The key to efficiently detecting road bridge targets from remote sensing images is to extract bridge features from remote sensing images [8]. The extraction of the road bridges is done by comparing the methods like Sobel, Roberts [9], LoG, and Canny edge detection.

The organization of the article is as follows: Sect. 2 briefs about an overview of bridge detection techniques over water, Sect. 3 briefs about an overview of bridge detection techniques on land, Sect. 4 briefs about discussion, and Sect. 5 concludes the article.

2 An Overview of Bridge Detection Techniques Water

Tang and Dong [10] proposed a methodology used for the detection of bridges on water using the modulated deformable convolution and attention mechanisms. As the bridges have different aspect ratios in remote sensing images, it is difficult to find small bridges accurately in remote sensing images. Usually, to find the small bridges, they proposed a methodology of selective attention usage strategy that improved detection performance. Here spatial decision rules are applied to find bridges. Correlation, homogeneity, and entropy features are used for distinguishing rivers. Hough transformation is utilized for the extraction of bridges. Deep network stack convolution layers are used to obtain the different features of each level such as ResNet [11] and DarkNet [12]. The last four steps are then processed by other constructs of the DL-based detector. At the end of the last four stages, the attention mechanism is employed. The rotations and OBBs of the bridges were calculated using BBCM containing filtering, clustering, and spatial domain operations in the frequency domain. In the verification part, mostly DarkNet-53 in YOLOv3 and ResNet-50 in R-CNN with FPN are used [13]. And selective channel-spatial attention usage is further implemented to erase redundancies and evade inferences.

Guo et al. [14] proposed a methodology useful for accurately detecting bridges in aerial images even if there is a waterbody constraint. Here, the deep learning model is used to detect the bridges in optical aerial images, where both the waterbody segmentation and the bridge detection can be performed simultaneously. Two-stage detectors, such as Faster R-CNN [15], Cascade R-CNN [16], and DetectoRS [17], create predictions based on region recommendations. First, a model for oriented bridge recognition with waterbody segmentation as an additional task serves as a direction for bridge localization. Second, a waterbody segmentation map was built as the waterbody constraint, which adds the prior knowledge of bridge distribution to sharpen the network predictions by using the semantic properties of the waterbody as spatial attention to separate bridges from crowded backdrops.

Chen et al. [18] proposed a methodology, i.e., deep learning is used for the auto bridge detection of SAR images that are based on adaptively effective feature vision. The background of the SAR images is complex [19]. For target detection, SSD uses multi-scale feature maps [20]. Since SSD produces a huge number of candidate boxes while training, if it is well classified, then a small gradient will produce. The input SAR dataset contains all the SAR images of size 500*500, and their corresponding labels. TerraSAR images (3-m resolution) are used as the experiment dataset and contain 1560 samples. SSD-AEFF can efficiently extract river bridge features and reduce background interference, and other features such as false alarms and missing detection have been reduced. The bridges can also be classified based on their length as large, medium, or small. SSD-AEFF module is used for multi-scale feature map enhancement with an effective squeeze excitation module to reduce background information noise.

Malini and Moni [21] proposed detailed information about the earth's geographical characteristics, including highways, rivers, bridges, and buildings can be found

in satellite images. Both military and civilian applications need these features. These features must be automatically detected for maps of the landscape and Geographic Information Systems (GIS). ADC is used for digitally storing the image. Unevenly spaced Fourier transform (USFFT) or principally based on the wrapping of spatially selected Fourier data are 2 techniques for performing the fast discrete curvelet transform (FDCT). Consequently, before performing feature detection, appropriate denoising methods must be introduced into the pictures. The image is saved in digital form using an ADC to transform the analog sensor current fluctuation to digital form [22]. Although there are other techniques for denoising images, the curvelet transform (CT)-based method is used because it can detect long edges, which is necessary for bridge identification [23].

Gu et al. [24] proposed a method that deals with hierarchy algorithms for extracting the bridge over water in optical images which are taken as input. They extract the river regions that the bridges are included to decrease the omission of bridge image by identifying the edge [25]. The coarse water bodies are first extracted from the optical picture using an iterative threshold, followed by the removal of the noise regions and the inclusion of the lacking areas the application of k-means clustering with spatial integrity and texture data. Segment the image using an iterative threshold that has been amplified. In the aerial photograph, the sea is often darker than the land, and the level of gray is less than a positive degree [26]. Moreover, the bridges stand out in the removed river areas. Finally, geometric data and the frequency of bridges over rivers are used to validate the bridges. The outcomes demonstrate how well this method performs in both aerial optical pictures captured by unmanned aerial vehicles and satellite images from Google Earth for the extraction of bridges.

Fu et al. [27] proposed a methodology to create bridge knowledge models. The hypothesis and test phases of the stream are based on top-down knowledge. Using techniques such as water segmentation, ROI extraction with morphological operators and connectivity signatures, and identification of candidate regions, hypothesis [4] was approximated. Gray characteristics are used in the testing procedure to authenticate potential bridges, and parameters like the bridge's coordinates and azimuth may be retrieved. For the pre-treatment, fuzzy threshold segmentation is used [28]. Hough transform, the morphological opening operation, is used to eliminate the pseudo-waters, and smooth water's edge does not affect positioning accuracy because the positioning is just done roughly. For reducing target detection uncertainty by utilizing multi-band picture information from several spectral ranges.

Using multispectral pictures, Chaudhuri and Samal [25] suggested a system for detecting bridges over aquatic bodies. An eight-type classification of land cover is initially performed on a multispectral picture using a majority must be granted reasoning based on the multi-seed supervised classification method. The identified image is then divided into three categories: background, water, and concrete. Then, using a knowledge-based technique that employs a five-step process to utilize the spatial arrangement of bridges and their surroundings, bridges are identified in this tri-level picture. With the use of geometric restrictions and a recursive scanning approach, a river extraction module locates the rivers. We determine the potential bridge pixels by using a neighborhood operator and the spatial dimension of a typical

bridge. Based on their connection and physical characteristics, these prospective bridge pixels are then categorized into potential bridge segments.

Han et al. [29] proposed a method for bridges over a river. It proposes a rapid detection algorithm by using satellite imagery. Usually, the river has high reflectivity and its gray value is greater than that of a background which is called histogram thresholding. The gray mean variant distinguishes water from land. The texture analysis is about the gray level co-occurrence matrix (GLCM) method. Feature extraction uses the histogram-fitted curve of both object and background. Here, the feature selection presents with the assigned seeds and is used by merging a pixel into the nearest neighboring seed region [30] presents an automatic seed region algorithm used in the color image segmentation. As the river land is found by using GLCM, then the detection is based on edges that are parallel and the sides relate to the land. The river image is processed using morphological methods [31], and parallel line detection and the recognized lines are detected using Hough transformation [32]. The main features extracted from bridges have long, parallel edges, and the bridge over the river has two sides that are next to the river and two sides that are connected to the land.

3 An Overview of Bridge Detection Techniques on Land

Yu and Nishio [6] proposed a methodology that uses long-distance datasets to identify the type of bridges using the ResNet-50 network. Multi-level bridge inspection consists of three levels of inspection that are types of bridges, components of bridges, and parts of bridges. Images for the dataset have been taken from the drone. These images are trained by using computer vision technology. The model uses the convolution neural network architecture. The type of bridge is classified using ResNet-50, and the part of the bridge is detected by object detection using YOLOv3; then, the instance segmentation is done by using Mask R-CNN. The accuracy of the model is tested by the evaluation metrics, and performance is calculated by using accuracy, recall, precision, and AUC [33]. After training the dataset, there is a conclusion that the indexes of the suspension bridges were slightly lower than the cable-stayed bridges. This model easily finds out the arched and cable-stayed bridges and some difficulty in suspension bridges. The bridge parts are detected by using a bounding box using the LabelMe program.

Yang et al. [7] proposed a method that describing the imaging characteristics. The road tracking is done by the Roberts [9], Sobel, LoG, and Canny edge detection algorithm; then the edge vectorization is developed by binarization images and it continues to remove the invalid line segment removal. In the edge extraction, there will be some other issues like billboards, traffic signals, and vehicles; these all are removed by the invalid line segment removal. In this line detentions, if any parallel lines are detected, then these are bridges in this way bridges are detected. Using the detected edges, the unwanted vector lines are removed and only the parallel edges are detected. In this proposed method, edge detection is the feature that has been

recognized first and the outer edge-by-edge detection [34]. Edge vectorization is detected using the edge binarization, and then the removal of invalid lines is detected and removed. If any parallel lines are detected, then these lines are described as the bridges. The length of the bridge is detected as the point in the plane by the distance of the two object calculations. We can easily find out the distance between the two points. The proposed method uses the VS2008 + ArcGIS Engine environment.

Nogueira et al. [35] proposed a method that offers information regarding the status of these constructions and support potential decision-making processes; it is essential to identify bridges in large infrastructure projects. Typically, human agents must find the bridges into large-scale datasets by examining each image, which is a laborious and time-consuming operation. In You Only Look Once (YOLO) method [36], a multi-objective feature that accepts an input picture and returns bounding packing containers and their respective instructions is employed to teach the network. YOLO basically turns every image into a rectangular shape of dimension S*S. For each grid location, the team creates N boundary containers and the related private chats, which are utilized to filter the detection concepts. This uses the faster R-CNN (regions with CNN features), a cutting-edge deep learning-based technique for numerous object cognizance and identification applications [37]. Two components make up the framework of this method. The first one is made up of a VGG16 that has been pre-trained using the ImageNet dataset and is in charge of producing location suggestions. The second module consists of region proposal network (RPN), which creates bounding container predictions using the previously proposed areas.

Sithole and Vosselman [38] proposed a method based on the principle of Light Detection and Ranging (Lidar). Automated filtering has been used in air-borne laser scanning. The algorithm is intended to find bridges in point clouds that have had all of their objects removed, or bare earth point clouds. Buildings, grass, automobiles, lampposts, and electricity wires are among the eliminated objects. The created bare earth point cloud is eventually utilized to create digital terrain models (DTMs). At first, segmentation is used then detection is done, later preprocessing is done, then selecting seed points, and then the identification of points in a bridge. Then, different tests are done like simulated data, Nijmegen 1, Kruithuis, Reineveld, Geerweg, Nijmegen 2, and Stuttgart all done on the dataset and will detect the output of that resultant image.

There is a detailed analysis of algorithms and mathematical models used in research articles which are described in Table 1.

4 Discussions

In general, this paper proposes extraction techniques like water body extraction and separating the water bodies and land using the Canny edge detection. And also bridge detection can be done using different algorithms like fast R-CNN, denoising techniques, feature extraction, and also by using parallel line detection for the detection of the parallel bridges. Here, the differentiation between two types of bridges are

Table 1 Analysis of Algorithms and Mathematical Models

Research work–year	Mathematical model/ Algorithm	Purpose of usage	Formula	Limitation
Yu and Nishio [6]—2022	ResNet-50 network	It is a pre-trained network to classify images	Accuracy, precision, recall, F1–score, and AUC	Few Types of bridges and bridge components are classified
	YOLOv3	Identifies the objects in the images		
	Mask R-CNN	Used for image segmentation		
Tang and Dong [10]—2022	ResNet and DarkNet	To obtain low/ mid/high-level features	Attention mechanism, frequency	Small bridges detection is difficult when there are different environments
	YOLOv3	Identifies the objects in images		
	Faster R-CNN with FPN	To reduce the overall detection time		
Guo et al. [14]—2021	Faster R-CNN	To accurately and quickly predict the location	Multitask loss function, segmentation XLoss, inclination, fusion mask, precision, recall	It detects only with the aerial image dataset
	Cascade R-CNN	For multi-stage object detection		
	DetectoRS	To detect objects using a Switchable Atrous Convolution, a recursive feature pyramid		
Yang et al. [7]—2021	Canny edge detection	Identify a broad variety of edges	Threshold	Only parallel brides are detected correctly but not ring-shaped and irregular bridges
	Sobel edge detection	Calculate the gradient of image intensity at each pixel		
	Robert edge detection	Computes an image's 2-D spatial gradient measurement		
	Laplacian of Gaussian (LoG)	Useful for finding edges		

(continued)

Table 1 (continued)

Research work–year	Mathematical model/ Algorithm	Purpose of usage	Formula	Limitation
Chen et al. [18]—2021	Adaptively spatial feature fusion	Increases the prominence of features while decreasing the impact of background features	Feature map, feature vector, weighted vector, SoftMax function, gradient density, entropy loss function	Limited range of object detection
	Gradient harmonizing mechanism	To solve the problem of sample imbalance		
	Single shot detector	For target detection		
	Non-maximum suppression	Remove redundant candidate boxes		
Nogueira et al. [35]—2019	YOLO	Identifies the objects in the images	—	Only few deep learning models are used
	Faster R-CNN	Used for image segmentation		
	VGG16	Deep learning model		
	Region proposal network (RPN)	Bridge region extraction		
Malini and Moni [21]—2015	Geographic information systems (GIS)	Automatically detected for terrain mapping	MATLAB, First, denoising, multiscale transform, namely curvelets, curvelet coefficients	Noise interruption can't produce perfect bridge detection
	Fast discrete curvelet transform (FDCT)	Scale for translation		
	Unequally spaced Fourier transform (USFFT)	Scale for translation		
	Curvelet transform (CT)-based method	Bridge region extraction		
Gu et al. [24]—2011	k-means clustering	Removal of the noise regions	Mathematical morphology, local binary patterns (LBP)	Bridges are detected when there is more region of water
	The RGB value	To distinguish the bridge pixels		
	Edge mending algorithm	Extract water regions		

(continued)

Table 1 (continued)

Research work–year	Mathematical model/ Algorithm	Purpose of usage	Formula	Limitation
Fu et al. [27]—2009	Fuzzy threshold segmentation	It is used for pre-treatment, i.e., segmentation	Gray threshold	It does not detect the bridges near mountain region
	Connectivity sign	It is used for eliminating discrete noises		
	Improved Hough transform	For the geometric feature extraction		
Chaudhuri and Samal [25]—2008	Seed point detection	Removes outliers from training samples	Elongatedness, directional and percentages of water index, search radius	Only parallel bridges with few meters of height are detected
	Statistical parameter extraction algorithm	To find the major subclusters in each class and calculate their first-order statistics		
	Minimum-distance logical-based classifier	To categorize an image's unknown pixels		
Han et al. [29]—2007	Knowledge base algorithm	Used to cluster input objects	Histogram thresholding, angular second moment, correlation, entropy, contrast, homogeneity	Consider only some features of water body bridge detection
	Rapid cluster detection algorithm	Used for grouping objects based on similarities		
	Gray level co-occurrence matrix	It examines the spatial relationship among pixels		
	Seeded region growing algorithm	Used for color image segmentation		
Sithole and Vosselman [38]—2006	Segmentation-based filtering	To detect ground segments	—	This is applicable for the variety and different bridges

discussed: bridges over water and bridges on land. By comparing these two, bridges on land is giving some less efficiency as it has more noise when compared to bridges over water and it has light disturbance to find the curved and different shaped bridges while compared to parallel bridge. We have summarized, in this paper, the best techniques of detecting and extraction of bridges and described several methods that can be applied for the Google Earth Images including using object detection algorithms and convolutional neural networks like YOLOv3 and YOLOv5 algorithms. We proposed to perform in two steps: (A) Gathering the dataset images from Google Earth (B) Detection of the bridges. As YOLOv5 has higher detection quality (mAP) and training is a single stage, we can expect better results.

5 Conclusion

Bridges have a significant influence on individuals who travel, commute to work, and so on. Bridges must be built securely and correctly to serve their purpose of providing a convenient method for people to travel from one location to another. Each article is designed for detecting bridges over land and water using different types of deep learning and machine learning techniques. Different types of extraction techniques like water extraction tasks, convolution neural networks, and deep learning techniques are used. The study aids in the selection of the algorithm to be utilized for detecting bridge extraction.

References

1. Trias-Sanz R, Loménie N (2003) Automatic bridge detection in high-resolution satellite images. In: International conference on computer vision systems (ICVS). Springer, Berlin, Heidelberg, pp 172–181
2. Du ZG, Lu L, Liang J, Li S, Yang S-L (2005) Recognition of bridge over water in air-plane image. J Wuhan Univ Technol 29(2):230–233
3. Wu H, Liu Z-K, Zhang R (2003) A study of bridge recognition from landsat TM images. J Remote Sens (Beijing) 7(6):478–484
4. Tang L-B, Zhao B-J (2007) A real-time recognition algorithm of bridges above water in aerial images. Acta Electonica Sin 35(3):511–514
5. Nie X, Zhao R, Zhang Y (2003) A new method for detecting bridges automatically. Xibei Gongye Daxue Xuebao (J Northwest Polytechnical Univ) (China) 21(5):599–602
6. Yu W, Nishio M (2022) Multilevel structural components detection and segmentation toward computer vision-based bridge inspection. Sensors 22(3502):1–18
7. Yang W, Gao X, Zhang C, Tong F, Chen G, Xiao Z (2021) Bridge extraction algorithm based on deep learning and high-resolution satellite image. Sci Program 2021(9961963):1–8
8. Zhang X, Yang Y, Li Z, Ning X, Qin Y, Cai W (2021) An improved encoder-decoder network based on strip pool method applied to segmentation of farmland vacancy field. Entropy 23(435):1–18
9. Sholeh HA, Mulyani Y, Saptama HD (2018) Studi perbandingan pengenalan karakter aksara lampung dengan metode deteksi tepi roberts dan sobel. Inovasi Pembangunan J Kelitbangan 6(3):261–272

10. Tang R, Dong G (2022) Bridge-over-water detection via modulated deformable convolution and attention mechanisms. EURASIP J Adv Sig Process 2022(56):1–15
11. 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 (CVPR), pp 770–778
12. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
13. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2117–2125
14. Guo H, Zhang R, Wang Y, Yang W, Li H-C, Xia G-S (2021) Accurate bridge detection in aerial images with an auxiliary waterbody extraction task. IEEE J Sel Top Appl Earth Obs Remote Sens 14:9651–9666
15. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, vol 28, pp 1–9
16. Cai Z, Vasconcelos N (2018) Cascade R-CNN: delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6154–6162
17. Qiao S, Chen L-C, Yuille A (2021) DetectoRS: detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10213–10224
18. Chen L, Weng T, Xing J, Li Z, Yuan Z, Pan Z, Tan S, Luo R (2021) Employing deep learning for automatic river bridge detection from SAR images based on adaptively effective feature fusion. Int J Appl Earth Obs Geoinf 102:102425
19. Chen L, Cui X, Li Z, Yuan Z, Xing J, Xing X, Jia Z (2019) A new deep learning algorithm for SAR scene classification based on spatial statistical modeling and features re-calibration. Sensors 19(2479):1–22
20. Xing J, Sieber R, Kalacska M (2014) The challenges of image segmentation in big remotely sensed imagery data. Ann GIS 20(4):233–244
21. Malini S, Moni RS (2015) Multiresolution image denoising for detection of bridges in satellite images. In: 2015 international conference on control, instrumentation, communication and computational technologies (ICCICCT). IEEE, pp 457–461
22. Gributs CEW (2004) Photon time-of-flight and spectroscopic characterization of scattering samples using analysis of self-similarity
23. Raju C, Reddy TS, Sivasubramanyam M (2016) Denoising of remotely sensed images via curvelet transform and its relative assessment. Procedia Comput Sci 89:771–777
24. Gu D-Y, Zhu C-F, Shen H, Hu J-Z, Chang H-X (2011) Automatic bridge extraction for optical images. In: 2011 sixth international conference on image and graphics. IEEE, pp 446–451
25. Chaudhuri D, Samal A (2008) An automatic bridge detection technique for multispectral images. IEEE Trans Geosci Remote Sens 46(9):2720–2727
26. Frasson RPDM, Wei R, Durand M, Minear JT, Domeneghetti A, Schumann G, Williams BA, Rodriguez E, Picamilh C, Lion C, Pavelsky T et al (2017) Automated river reach definition strategies: applications for the surface water and ocean topography mission. Water Resour Res 53(10):8164–8186
27. Fu Y, Xing K, Huang Y, Xiao Y (2009) Recognition of bridge over water in high-resolution remote sensing images. In: 2009 WRI world congress on computer science and information engineering. IEEE, pp 621–625
28. Pal SK, King RA, Hashim AA (1983) Automatic grey level thresholding through index of fuzziness and entropy. Pattern Recogn Lett 1(3):141–146
29. Han Y, Zheng H, Cao Q, Wang Y (2007) An effective method for bridge detection from satellite imagery. In: 2007 2nd IEEE conference on industrial electronics and applications. IEEE, pp 2753–2757

30. Shih FY, Cheng S (2005) Automatic seeded region growing for color image segmentation. Image Vis Comput 23(10):877–886
31. Maragos P (1987) Tutorial on advances in morphological image processing and analysis. Opt Eng 26(7):623–632
32. Cha J, Cofer RH, Kozaitis SP (2006) Extended Hough transform for linear feature detection. Pattern Recogn 39(6):1034–1043
33. Sundaram M, Mani A (2016) Chapter 5: face recognition: demystification of multifarious aspect in evaluation metrics. Intech, pp 75–92
34. Liu Y, Cheng M-M, Hu X, Wang K, Bai X (2017) Richer convolutional features for edge detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3000–3009
35. Nogueira K, Cesar C, Gama PH, Machado GL, dos Santos JA (2019) A tool for bridge detection in major infrastructure works using satellite images. In: 2019 XV workshop de visão computacional (WVC). IEEE, pp 72–77
36. Pandey P, Pandey MM (2021) Research methodology: tools and techniques. Bridge Center
37. Krishna ST, Kalluri HK (2019) Deep learning and transfer learning approaches for image classification. Int J Recent Technol Eng (IJRTE) 7(5S4):427–432
38. Sithole G, Vosselman G (2006) Bridge detection in airborne laser scanner data. ISPRS J Photogramm Remote Sens 61(1):33–46

Vegetation Change Detection of Multispectral Satellite Images Using Remote Sensing

G. Sai Geethika, V. Sai Sreeja, T. Tharuni, and V. Radhesyam

Abstract Change Detection is depicted to compare the spatial representation of two points in time, with variations in the variables of interest causing changes. Remote sensed data of Landsat 8 satellite imagery can be used to detect changes in multispectral images. Vegetation change detection plays a prominent role in tracking the alteration of vegetation in selective areas. The vegetation change analysis of a specific area for a given time period involves a lot of information that can be used for predicting the impact of change over years. Multispectral images for vegetation change of Landsat 8 satellite for a specific location can be obtained by stacking selective bands together. Over the years, due to urbanization, the vegetative index of cities dropped drastically. To take necessary measures, the impact must be analyzed. To overcome this problem, we are using vegetation change detection analysis. The image stacking is performed using QGIS (Quantum Geographic Information System) software which is facilitated using various image enhancement options. The location Vijayawada is tracked for detecting change over a time period of eight years using the NDTS (Normalized Difference Between Time Series) algorithm followed by comparison with the PCA and K-means algorithm. This paper gives a detailed visualization of the results acquired in this project using metrics like RMSE and PSNR.

Keywords Landsat 8 · Remote sensed data · Multispectral images · Satellite imagery · Normalized difference between time series (NDTS) · PCA (principal component analysis) · K-means · Vegetative index · QGIS (quantum geographic information system)

G. S. Geethika (✉) · V. S. Sreeja · T. Tharuni · V. Radhesyam
Department of Information Technology, VR Siddhartha Engineering College, Vijayawada, India
e-mail: saigeethikagudapati@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_25

1 Introduction

Satellites gather an enormous amount of data every day. This data comes into use when the data is processed to gain useful insights. The essence of satellite imagery is that it gives the benefit of accessing remotely sensed data [1]. Remote sensed data is the relevant land cover data that is accessed anywhere irrespective of time and location. There is no requirement of visiting a particular location to explore the data. The data which is gathered in such a way can be used for analyzing the vegetation cover over a period of time. This information gives us the insights to predict cases like the increase or depletion of vegetation. The scenarios where depletion of vegetation cover is observed can be used for researching the drastic impacts on nature as a result.

Change detection techniques are not limited to land cover changes. The selective bands give us the detail of what is being detected. A few other observations that can be explored using change detection techniques are changes in land cover, and water bodies like rivers, lakes, and oceans. Predicting the differences using computer-based algorithms makes the process of finding the change faster and more efficient with improvements and enhancements that can be accommodated easily. The ecosystem stays balanced when there is a significant index of vegetation and forest cover.

Over 26% of the vegetation area was lost to habitation between the years 2005 and 2020. It is clear that the approaching era will be undesirable if these patterns of land use change persist. The majority of the land is covered by built-up areas (industries, commercial areas), as opposed to agricultural areas, however, agricultural areas shouldn't be thought for conversion to built-up areas [2]. Digital agricultural applications are crucial for tracking remote harvest and assessing the state of farmlands. As a result of their effectiveness in identifying land cover components, high-resolution pictures have drawn much more attention [3]. This estimate can be applied mainly to urban areas where the vegetation index decreases due to the increase in population. But it is not limited to any region.

The NDVI approach is commonly used to detect changes in the amount and distribution of vegetation, as well as in land use and land cover. Thanks to advancements in the spatial and spectral (from broadband range to small range) resolution of remote sensing data, it is now possible to work at the micro level [4].

1.1 Literature Study

According to Sumanta Bid, it is observed that NDVI technique with specific thresholds can detect changes in the vegetation of a place using remote sensing [4]. You Y, Cao J, Zhou W produced research article focused on different kinds of change detection techniques that can be used for urban change detection [5]. Song A, Choi J, Han Y, Kim Y research paper included change detection in hyperspectral imagery by using convolutional neural networks [6, 7]. Previous research on change detection

for vegetation on multispectral images is either primarily focused on a single algorithm like NDVI or focused on urban changes and hyperspectral images. But there is a need to detect vegetation change accurately using satellite images and compare it to other techniques to identify the best technique.

Our proposed study provides extended research by comparison study of K-means and NDVI for vegetation change detection on multispectral satellite images and determines the performance of both algorithms primarily for detecting accurate changes in vegetation over years.

2 Proposed Work

Change detection identifies the spatial changes induced by man-made or natural processes in multi-temporal satellite images. Remote sensing for change detection can help with urban planning and expansion by considerably improving land utilization. The objective of the project is to gather Landsat 8 satellite images and apply NDVI and compare results with PCA and K-means between 2 timestamps of the Vijayawada area.

2.1 Datasets

The dataset is taken from earth explorer [8] with USGS credentials. The images are located with a specified coordinate set enclosed in circular shape. The image radius considered for this project is 250 m near Vijayawada which can be observed in Fig. 1. The datasets that are available are displayed as search results. There are alternative variations for each dataset which include the year, cloud cover, area covered, and temperature constraints.

The selected dataset can be viewed with a footprint market on the map which can be observed in Fig. 2 and the specification can be viewed in Table 1. The image is displayed as a tile on the map. When the dataset is downloaded, the bands that are available for the Landsat 8 satellite are displayed [9]. To download the data, USGS account credentials are necessary. The data collected is Landsat Collection 2 with Level-2 data including Landsat 8–9 OLI (Operational Land Imager)/TIRS C2 L2 as specifications. The image data with feature class as vegetation features with minimum cloud cover is selected to acquire accurate results.

2.2 Data Extraction and Pre-processing

QGIS acts as GIS (Geographic Information System) software which allows different users to explore and edit spatial data. Beyond that, it allows composing and exporting

Fig. 1 Image location view in earth explorer

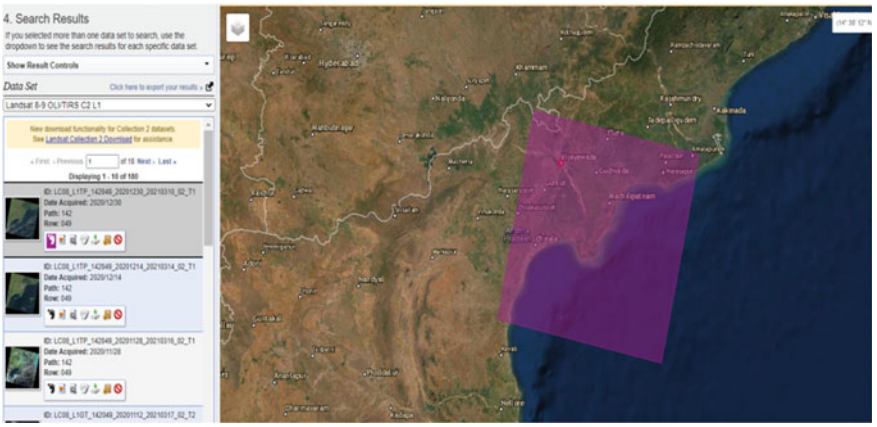


Fig. 2 Image location tile view in earth explorer

Table 1 Feature details of the dataset

Feature	Value
Geocoding method	Feature (GNIS)
Name	Vijayawada
Country	India
Center longitude	80.5991
Radius	250 m

various kinds of maps. The images are selected and pre-processed using QGIS Semi-automatic Classification Plugin (SCP) [10]. The images are cropped according to required coordinates to map a specific location.

2.3 Design Methodology

The multispectral images that are collected in the area of Vijayawada with timestamps 2013 and 2021 are collected as eleven different bands individually as Landsat Collection 2 Level 1 data which can be viewed in Fig. 3. The specification for level 1 data is considered as Landsat 8–9 OLI/TIRS C2L1. The bands that correspond to different change detection are listed in Table 2.

For vegetation analysis, bands 4, 5, 6 are considered. These image bands are stacked using QGIS software and pre-processed to remove noise. Our first change detection approach is calculating NDVI, and second approach is PCA and K-means.

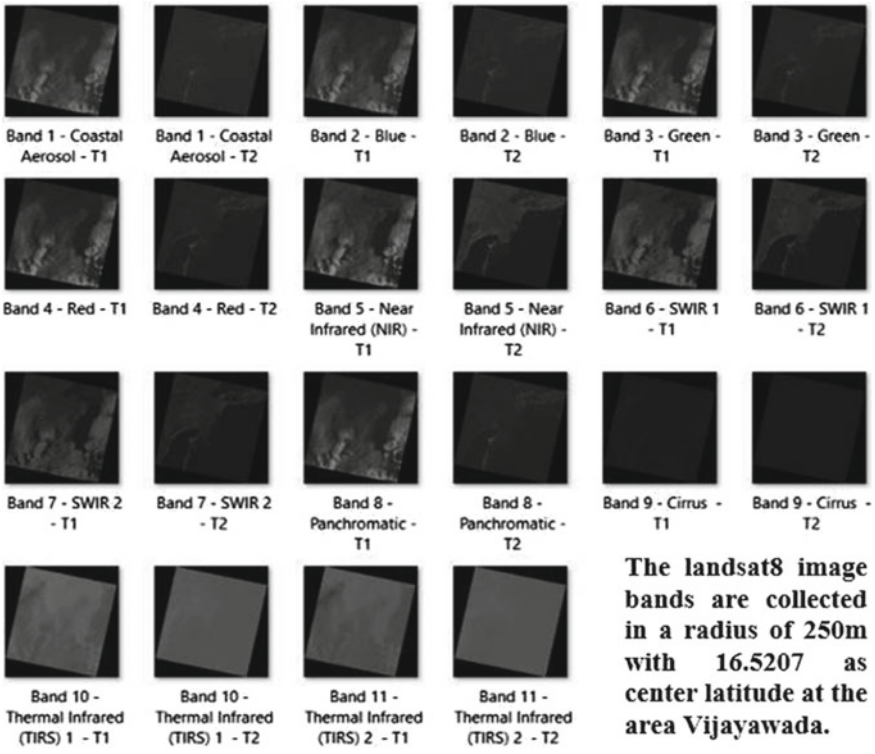


Fig. 3 Image bands of two timestamps

Table 2 Landsat 8 band designations

Sr. no.	Bands	Wavelength (μm)
1	Coastal aerosol	0.43–0.45
2	Blue	0.45–0.51
3	Green	0.53–0.59
4	Red	0.64–0.67
5	Near Infrared	0.85–0.88
6	SWIR1	1.57–1.65
7	SWIR2	2.11–2.29
8	Panchromatic	0.50–0.68
9	Cirrus	1.36–1.38
10	TIRS1	10.60–11.19
11	TIRS2	11.50–12.51

2.4 Procedure

NDTS

First step is importing libraries that are needed to open Geo TIFF format images. The libraries required are osgeo, gdalconst, NumPy, SciPy, IPython, matplotlib which are built-in libraries in python. Load the datasets after pre-processing using QGIS SCP plugin. The datasets used here are the stacked images. The images are loaded, and the files are named to apply the functions that are available in the libraries. Editing spatial data in QGIS can be enabled using digitized tools [11].

Plot the histograms for both bands. By taking multiple data points and organizing them into logical ranges or bins for the pixel values, the histogram condenses the given picture data series into an easily understood visual. NDTS can be computed in vegetation as NDVI with 0.1 as threshold value.

$$\text{NDTS} = \frac{(I_{t2} - I_{t1})}{(I_{t2} + I_{t1)}} \tag{1}$$

Here, I_{t2} represents Image at timestamp t_1 (2013) with red reflectance and It_2 represents Image at timestamp t_2 (2021) with near infrared reflectance.

The NDVI approach is a straightforward arithmetical indicator that may be applied to remote sensing readings to determine whether or not the target or object being examined has significant vegetation. The array values of two images are used to find NDVI by using 0.1 as threshold value to identify the best satellite image differentiating using the algorithm which is specifically applied to calculate Normalized Difference Vegetation Index in the images. The connection between fluctuations in vegetation growth and spectral variations rate has been extensively studied using the NDVI (Normalized Difference in Vegetation Index). Determining the growth of green vegetation and spotting changes in the vegetation are both useful [12].

The image files are obtained as an output after calculation of NDTS. The square of values is calculated for comparison with the original image. The square of NDTS is calculated to increase scope of detecting change between marginal values that are estimated during NDTS calculation and remove noise using 3×3 mode filter. This improves the image accuracy and resolution to display changes.

Display the changes along with the detected output graph. The output is displayed in the form of a histogram and change map. The change map indicates the change using two colors. The black color signifies no change, and the white color indicates change observed. NDVI signifies vegetation change detection in two images.

Principal Component Analysis (PCA)

The primary purpose of PCA is to reduce dimensionality. A high number of variables are condensed into a smaller number in order to minimize the dimensionality of large data sets while retaining the majority of data.

PCA includes the following operations:

The first action is standardization. This stage involves normalizing the range of continuous beginning variables such that each one contributes equally to the analysis. Mathematically, the mean for each value of each variable may be subtracted, and it can be divided by standard deviation.

$$Z = \frac{\text{value} - \text{mean}}{\text{standard deviation}} \tag{2}$$

Computation of the covariance matrix. In this stage, we try to figure out how the variables in the given input dataset vary from mean about each other, or if there is any link between them.

$$\begin{pmatrix} \text{cov}(a,a) & \text{cov}(a,b) & \text{cov}(a,c) \\ \text{cov}(b,a) & \text{cov}(b,b) & \text{cov}(b,c) \\ \text{cov}(c,a) & \text{cov}(c,b) & \text{cov}(c,c) \end{pmatrix} \tag{3}$$

Calculate the eigenvectors and eigenvalues of the covariance matrix to find the primary components. By linearly integrating the initial variables, principal components are formed as completely new variables. *n* major components are given from n-dimensional data. The ten main components are thus provided by 10-dimensional data. The second component, however, has less information than the first component, since PCA concentrates the most information on to the first component.

The last step is to select the feature vector. The major components of this feature vector are quite important. Low-importance principal elements may be discarded. We can determine the primary components in terms of importance by computing the eigenvectors and sorting them by their eigenvalues in descending order.

K-Means Clustering

By using the K-Means clustering method, an informative index is divided into K unique, non-covering groups and it can be used for change detection [13]. Characterize the number of bunches you require (K) before using K-Means grouping. The K-implies calculation will then assign each perception to one of the K groups [14].

The K-Means uses a centroid that minimizes the idleness between the points. It can be represented by below equation

$$\sum_{i=0}^{n} \min\left(\left|\left|x_i - \mu_j\right|\right|\right)\left(\left|\left|x_i - \mu_j\right|\right|\right) \tag{4}$$

Determine the number of clusters (K) and then randomly assign K different centroid locations. Finding the Euclidean distance across each point and the centroid is the following step. Each point should be assigned to the closest cluster before the cluster mean is determined as the new centroid. After the new point is assigned, the new centroid's position (X, Y) is:

$$\frac{X = (x_1 + x_2 + x_3 + x_4 + \cdots + x_{n-1} + x_n)}{n} \tag{5}$$

$$\frac{Y = (y_1 + y_2 + y_3 + y_4 + \cdots + y_{n-1} + y_n)}{n} \tag{6}$$

2.5 System Architecture

The architecture of the entire change detection system can be seen in Fig. 4.

3 Results and Observations

Different band combinations can be observed in Table 3. The bands selected for vegetation change detection are Red, Near Infrared, and Short-wave Infrared 1. The wavelength required for the detection of change ranges from 0.64 to 1.65 μm. The image resolution is around 30 for every image band of Landsat 8 [15].

The image features are displayed in the form of a dense peak with values from negative value range to positive value range. A change map indicates the change between two images at different timestamps. From the analysis of the vegetation index in the area of Vijayawada, there is a significant decrease in the vegetation of the city. The reports suggest that this decrease in vegetation index is due to urbanization and the development of cities due to the increase in population over the past few years.

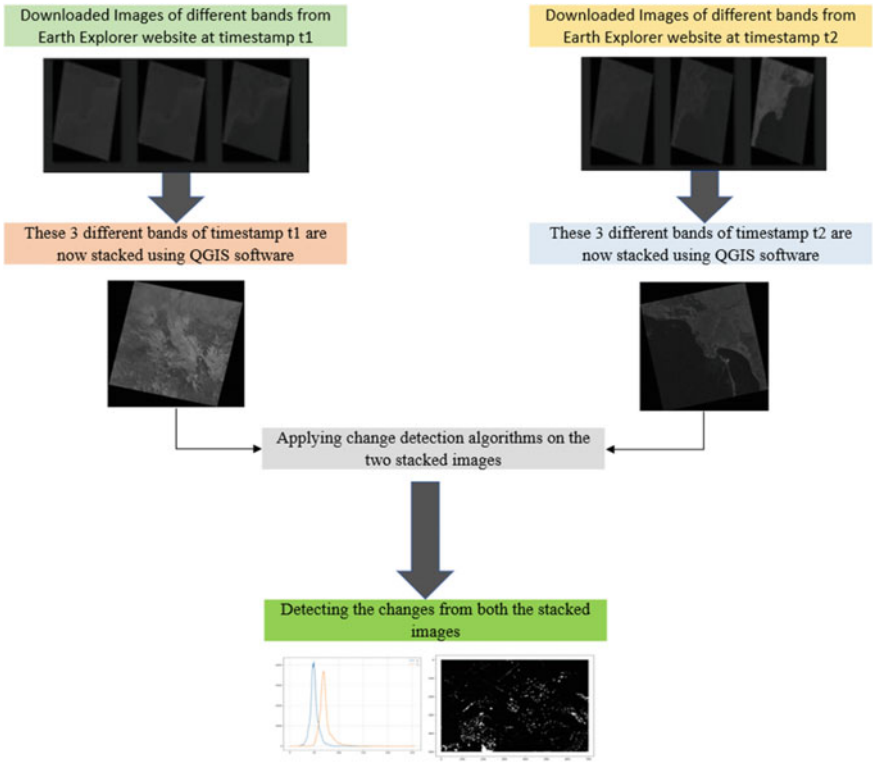


Fig. 4 System architecture diagram

Table 3 Bands associated with landscape

Property	Bands
Natural color band	4, 3, 2
False color (urban) band	7, 6, 4
Color infrared (vegetation) band	5, 4, 3
Agriculture band	6, 5, 2
Atmospheric penetration band	7, 6, 5
Healthy vegetation band	5, 6, 2
Land/water band	5, 6, 4
Natural with atmospheric removal band	7, 5, 3
Short-wave infrared band	7, 5, 4
Vegetation analysis band	6, 5, 4

This caused a hike in agricultural area removal and deforestation which decreased the vegetation index to a lower rate. The change map and associated graph results convey where the exact change is observed which gives the estimate of the vegetative index between the past few years.

RMSE is very helpful to measure a model's performance, during training, cross-validation, or even monitoring after deployment. The root mean square error is popularly used metrics for this. It is a reasonable rating scale that adheres to some of the most popular statistical hypotheses and is simple to understand. $y(i)$ is the ith measurement, $y\wedge(i)$ is its corresponding forecast, and N is the total number of data points.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \| y(i) - y \wedge (i) \|^2}{N}} \tag{7}$$

Image compression quality is also contrasted by the mean square error (MSE) and peak signal-to-noise ratio PSNR. PSNR indicates a measure of the peak error, whereas the MSE represents the cumulative squared error among original and compressed images. The error is inversely correlated with the value of MSE.

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{R^2}{\mathrm{MSE}} \right) \tag{8}$$

The histogram presented in Fig. 5 shows how the images are differentiated with each other. Blue curve indicates stacked image 1 and orange curve indicates stacked image 2. The change maps in Figs. 6 and 7 indicate particulary where the change is detected. White indicates change observed and black area indicates no change. The evaluation metrics obtained by calculating RMSE (Root Mean Square Error) is 0.0043071182 for NDTS and 0.0151108205 for K-means. It is used to calculate the variation between source image and obtained image. The metrics obtained by calculating PSNR (Peak Signal–Noise Ratio) can be seen in Table 4, which is 47.31626752546268 for NDTS and 36.41423705747562 for K-means. If PSNR is higher, then the image is of higher quality. If RMSE is higher, the produced image will be of lesser quality. Hence NDTS shows best change results compared to K-means.

4 Conclusion

The project mainly deals with the change of vegetation. The analysis is done using remote sensing techniques with the aid of satellite imagery which gives accurate results in less amount of time. The extended and depleted vegetation gives the idea of how the land cover is being used over the years. As the estimation of change using change detection algorithms overcomes the problem of identifying the location where the plantation is required, it helps various organizations to take steps toward

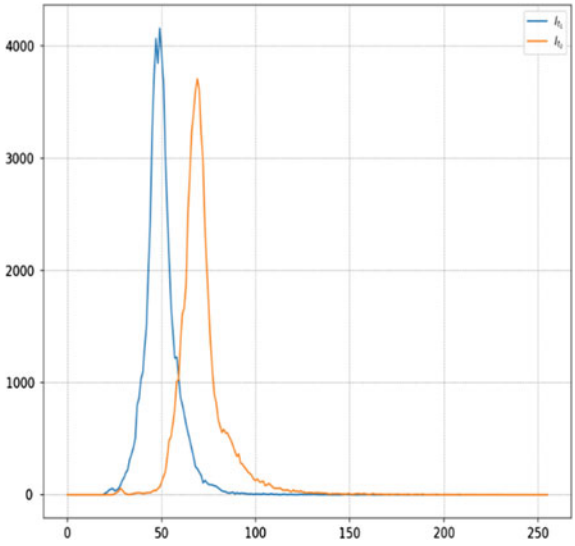


Fig. 5 Histogram depicting change between two images

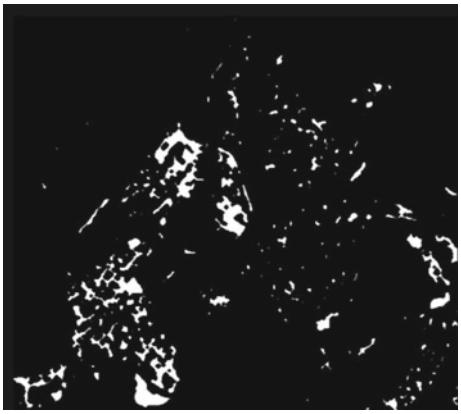


Fig. 6 Change map obtained by NDTS

growth and control over vegetation. The project mainly dealt with vegetation change detection in multispectral images, and it can be concluded that NDTS performs better than PCA k-means for vegetation change detection.

The future study of this project can be extended for land cover change detection and water body change detection. Research using change detection algorithms can help to assess the situation of increase in water level over the years through global warming. But the process can be more perspicuous if the work is extended to hyperspectral images. The algorithms that are newly employed using complex deep neural networks can give a presumable spike in accuracy and decrease the time taken for the process.

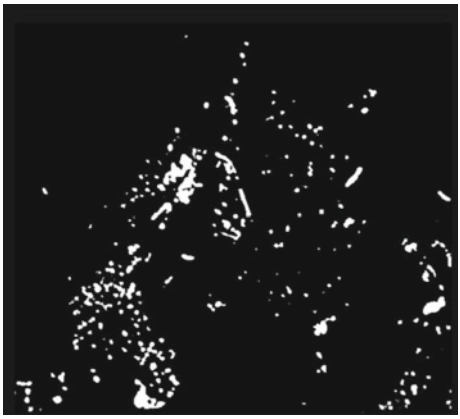


Fig. 7 Change map obtained by PCA K-means

Table 4 Metrics obtained

Algorithm/metrics	RMSE	PSNR
NDTS	0.0043071182	47.31626752546268
PCA and K-means	0.0151108205	36.41423705747562

References

1. Navalgund RR, Jayaraman V, Roy PS (2007) Remote sensing applications: an overview. Curr Sci, pp 1747–1766
2. Prasad AS, Ramamurthy C, Krishna KL (2022) Land use and land cover change and sustainability assessment of Vijayawada city by RS&GIS. In: IOP conference series: earth and environmental science, vol 982, No 1. IOP Publishing
3. Joseph S (2021) Accurate segmentation for low resolution satellite images by discriminative generative adversarial network for identifying agriculture fields. J Innov Image Process 3(4):298–310
4. Bid S (2016) Change detection of vegetation cover by NDVI technique on catchment area of the Panchet Hill Dam, India. Int J Regul Gov 2(3):11–20
5. You Y, Cao J, Zhou W (2020) A survey of change detection methods based on remote sensing images for multi-source and multi-objective scenarios. Remote Sens 12(15):2460
6. Mateen M, Wen J, Akbar MA (2018) The role of hyperspectral imaging: a literature review. Int J Adv Comput Sci Appl 9(8)
7. Song A et al (2018) Change detection in hyperspectral images using recurrent 3D fully convolutional networks. Remote Sens 10(11):1827
8. Bensi P et al (2007) A new earth explorer-the third cycle of core earth explorers. ESA Bull 131:30–36
9. Ridwan MA et al (2018) Applications of landsat-8 data: a survey
10. Tobias MM, Mandel AI (2021) Literature mapper: a QGIS plugin for georeferencing citations in Zotero. Air Soil Water Res 14:11786221211009208
11. Graser A (2016) Learning Qgis. Packt Publishing Ltd
12. Gandhi GM et al (2015) Ndvi: vegetation change detection using remote sensing and gis–A case study of Vellore District. Procedia Comput Sci 57:1199–1210
13. Vignesh T, Thyagharajan KK, Ramya K (2019) Change detection using deep learning and machine learning techniques for multispectral satellite images. Int J Innov Technolo Expl Eng 9(1S):90–93

14. Lv Z et al (2019) Novel land cover change detection method based on K-means clustering and adaptive majority voting using bitemporal remote sensing images. Ieee Access 7:34425–34437
15. He G et al (2018) Generation of ready to use (RTU) products over China based on Landsat series data. Big Earth Data 2(1):56–64

Performance Evaluation of Neural Networks-Based Virtual Machine Placement Algorithm for Server Consolidation in Cloud Data Centres

C. Pandiselvi and S. Sivakumar

Abstract Cloud computing, an on-demand model that provides IT services as a utility, has become increasingly popular in recent years. To the contrary, the cloud's resources are housed in data centres that have a major impact on the environment due to their high energy consumption. As a result, consolidating servers into fewer physical locations is a significant method for increasing data centre efficiency and reducing energy waste. In this work, we propose a technique for selecting and deploying servers and virtual machines using artificial neural networks. It forecasts user demand, evenly distributes work if the server is overloaded, trains a feed forward neural network with back propagation learning, uses a selection algorithm to determine which virtual machine to use, and finally uses a cross-validation algorithm to ensure that the placement was accurate. Cloud computing providers like Amazon (EC2), Microsoft Azure, and Google Cloud Storage provide the data used to measure the efficacy of server consolidation efforts. The algorithm's experimental results are compared to those of other algorithms published by the same author, both in exact and heuristic optimization methods, that aim to achieve the same goals.

Keywords Server consolidation · Neural network · Virtual machine placement · Energy efficiency

C. Pandiselvi · S. Sivakumar (✉)
Cardamom Planters' Association College, Affiliated to Madurai Kamaraj University, Bodinayakanur, India
e-mail: sivaku2002@yahoo.com

C. Pandiselvi
e-mail: pandiselvi.c@cpacollege.org

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_26

1 Introduction

Cloud computing is a rapidly expanding technology that enables the delivery of IT services to end users on a utility basis. Those services are provided by massive, virtualized data centres. For a cloud data centre to meet the demands of its customers, it must allocate its limited physical machines (PM) to accommodate as many virtual machines (VM) as the customers request, taking into consideration resources like processing power, random access memory, and storage space. Furthermore, increasing user demand has led to a rise in energy consumption in cloud data centres, which is becoming an increasingly pressing issue [1]. The studies have looked into ways to reduce data centre energy consumption by enhancing the data centres underlying hardware technology's performance.

Server consolidation is a method used by data centre energy management to lessen the amount of PM and, ultimately, the number of idle servers that must be powered down [2]. Choosing the consolidated target server for deploying the VM necessitates the use of energy management strategies. By employing server consolidation to minimize the necessary number of physical machines, it is possible to decrease the number of virtual machines. In order to determine the difficulty of assigning VMs to the optimal PM, the server consolidation algorithm (SCA) [3] can be used.

By monitoring whether the PM is overloaded, SCA in the cloud data centre can determine whether or not virtual machines (VMs) should be moved away from the PM in question and into the hands of a less taxed one. In any other case, all VMs must be migrated off it before the PM can be put into a power-saving, low-power, or idle mode. In order to decide which virtual machines (VMs) on the overworked host should be moved, the VM selection algorithm can be used. To optimize resource utilization and energy consumption, assign the chosen VM to the active PMs using the VM placement algorithm [4].

Several problems, including performance degradation, unanticipated fluctuations, time-outs, longer response times, and VM failures, can be brought on by the SCA approach [5]. The three phases of server overloaded/under loaded detection, virtual machine selection, and virtual machine placement in SCA must be optimized in order to ensure the service is of a high quality whilst lowering energy consumption and increasing resource utilization [6]. Therefore, a new idea is proposed to hybrid the neural network in SCA. In this chapter, an enhanced server consolidation algorithm (ESCA) based on a neural network (NN) is proposed, the back propagation algorithm used to train NN such that, it predicts the users demand, equally distributes the workload across all the nodes to train feed forward NN. If there exists under loaded server, it will be put in idle else, the overloaded server is taken as a training data, and mean square error (MSE) value is calculated. By using the minimum MSE value, an intelligent decision-making selection algorithm is used to select the VM, and a placement algorithm is then used to place and cross-validate the placement of VM, resulting in efficient optimization [7].

ESCA algorithm performance is assessed using datasets from Amazon EC2, Microsoft Azure, and Google Cloud. A variety of instance types, geared to match

various use cases, are available through Amazon EC2. In Microsoft Azure, there is an instance type that offers one or more scalable instance sizes to accommodate the demands of various workloads. The infrastructure of Google is present in Google Cloud, and it is utilized to create and run virtual computers. The experimental findings unmistakably show that the suggested ESCA algorithm places VM effectively by meeting the optimization goals.

To address the inefficient usage of server resources, reduce execution time, and manage resources efficiently, the following are some of the key contributions of this work:

1. A selection algorithm based on neural networks is presented to choose a VM from a server that is overloaded.
2. A placement algorithm based on neural networks is suggested to deploy and cross-validate the VM for effectiveness.
3. A new idea is proposed to hybrid the neural network selection and placement algorithm for enhanced server consolidation in cloud data centres.
4. The ESCA algorithm is compared to the algorithms with same objectives and showed that the VM placed effectively in proposed algorithm.

2 Related Works

Although many researchers have previously studied the server consolidation problem of virtual machine mapping, this work draws attention to some of the closest research in the context of server consolidation from a neural network perspective. In cloud data centres, some servers are overloaded for processing the user request services, some are under loaded, and some are totally idle. Turning off these idle servers and efficiently selecting and placing servers, transferring them from overloaded server to under loaded server, can also significantly reduce energy consumption. The work will not only save energy, it also utilizes the resources. But the problem lies in managing these idle servers, overloaded servers, and under loaded servers efficiently.

Abohamama et al. [8] have a server consolidation with migration and selection system that has decreased the number of migrations. Heuristics-based linear programming (LP) is developed to control migrations by including a constraint. A stable workload prevents a VM from migrating; instead, VMs with variable capacity are moved to lessen the need for real selection of servers. This will directly reduce the amount of electricity used.

Pandiselvi and Sivakumar [9], a migration control strategy is suggested along with the three VM selection methods. These methods use a migration control mechanism to optimize performance by reducing the migrations for virtual machines and energy usage. It performs better than previous heuristic methods and further reduces network traffic. Ferreto et al. [10], Fuzzy logic has been used in a VM selection method. The author advised using the fuzzy VM selection with migration control technique to make an informed choice based on changing workload. The suggested fuzzy logic algorithms offer a variety of benefits that help with handling uncertainty in the real

world and getting the best outcome through wise decision-making. The proposed technology saves energy whilst preserving QoS and has fewer SLA violations than existing methods.

Beloglazov et al. [11], a linear technique was described for forecasting CPU usage. The linear regression function predicts both an under loaded host and future CPU use, which is useful for overload detection. The suggested method is used in Cloud Sim, and the outcomes show a substantial decrease in energy costs and the achievement of a high degree of QoS. In [12], Beloglazov et al., the same authors suggest lowering the active set of physical servers based on workload needs by using a reinforcement learning-based server consolidation method. This approach uses an agent to identify the optimum response. Based on the required resources at the time, it decides which host should be put in active mode or sleep mode. When compared to alternative dynamic consolidation methods already in use, the suggested technique effectively lowers energy use and the number of SLA violations.

3 Materials and Methods

The NN operate based on the learning principle, which means that they must first undergo training before they can process input to produce the intended output. These networks are skilled to process any job by modifying the weights, the value of link between the processing parts. After training, NN can be utilized to simulate different nonlinear applications and forecast any new situation [13]. The anticipated output will then be produced at the productivity layer at the conclusion of the learning process. As a result, it produces superior results when a prediction and validation are made appropriately. This section discusses the validation-based placement algorithm for VM and the prediction-based selection method for VM that is used in the ESCA algorithm.

3.1 NN-Based Selection Algorithm

A proposed NN-based selection technique aids in making intelligent decisions whilst choosing a VM from a loaded PM [14]. The server overload PM, which was obtained from the preceding server overload detection phase, is provided as an input.

To choose the required VM from a loaded PM, training is done using feed forward and back propagation techniques. The information is initially transmitted from one end to the other using the feed forward technique, and then the back propagation learning algorithm trains the NN by propagating information backward to the hidden layer and subsequently to the input layer. The correlation coefficient and standard deviation are calculated repeatedly until the goal output is not met with the lowest possible error rate. Then, again it feed forwards to the productivity layer to select the VM. It helps in the increase in the performance of the system, as shown in Fig. 1.

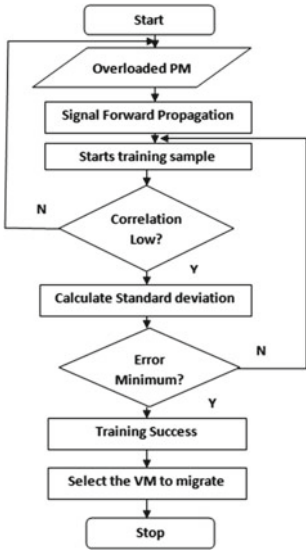


Fig. 1 Sequence diagram for neural network-based selection Algorithm

Algorithm 1. Neural network based selection algorithm

***Input**: Overloaded PMi(x)*
***Output**: Selected Vmi to be migration in VMi(m)*
1. Input overloaded PMi(x);
2. VMi(m) = GetMigratableVMi(m);
3. UMatrix (VMi) = UtilizationMatrix (VMi);
4. Metric(ni) = Cor-Co (UMatrix (VMi));
5. For each VMi(j) of Metric(ni)
6. CPUhisi( ) = GetCPUHist(j);
7. SD(i) = SD (CPUhist) by equ (1);
8. COR(i) = Cor-Coeff(i);
9. Outputneural = trainingdata (SD(i), COR(i));
10. If the mse of Outputneural is minimum then VMi(s) = j;
11. End;
12. Return VM(s);

The selection algorithm is illustrated in Algorithm 1. The overloaded PM serves as the algorithm's sole input. The earlier phase detects the overloaded PM: detection of overload. The GetMigratableVMi(m) function pulls all the VMs that are currently placed on that PM after having the PM(x) at step-1 and step-2 (x). The usage matrix of the VM is calculated by the UMatrix (VMi)function at step 3. Step 4 of the function calculates the correlation coefficient, which states that the likelihood of a server being overloaded increases with the correlation between the resource demands of the applications running on it. The correlation coefficient is based on the UMatrix (VMi) and is stored at Metric (n).

At steps 5 and 6, the function CpuHistory is used to retrieve the CPU usage history of I for each VMi(j). Using the standard deviation, SD Eq. (1) from CPUhist, SD(i), or standard deviation, is determined at step 7. Correlation for this VM will be retrieved at step 8. At step 9, the neural network will receive the training data and use the feed forward and back propagation learning approach to produce a VM output with the lowest possible MSE value. For migration, the VM with the lowest mean square error will be chosen. The chosen VM is finally returned in step 13 for placement.

Correlation

The possibility of a server being overloaded increases with the correlation between the resource usage of the apps operating on it. From step 4, the VM that can cause the server to get overwhelmed is described. A VM that has a low correlation with another VM should be moved.

Standard Deviation

If the SD is high in dynamic server consolidation, it means the CPU request varies frequently, whereas if it is low, it means the VM is consistently utilizing resources.

$$\mathrm{SD} = \sqrt{1}/n \sum_{i=1}^{n} \left(\mathrm{CPU_{id}} - \mathrm{CPU_{avg}}\right)^2 \tag{1}$$

CPU_i represents the *i*th CPU capacity, CPU_{avg} represents the average CPU capacity, and n represents the number of CPU capacity resources used. Equation (1) can be used to calculate the SD of CPU utilization for a certain VM.

3.2 NN-Based Placement Algorithm

NN-based placement algorithm is utilized to fit the most VM into the fewest PM [15]. The fundamental goal of placement is to conserve energy and fully avoid resource use possible. The neural network back propagation training algorithm is executed to cross-validate the placed VM using false placement. The placement method uses the best-fit strategy, which places the elements in the order they came. The next item should be put where its capacity is closest. The placement algorithm is used to distribute the most VM into a limited number of PM after determining whether the item does not meet the capacity of the current PM [16]. For effective resource usage and energy efficiency in cloud data centres, the NN is used to cross-validate the placement of virtual machines to determine whether they are suited for the PM or not, as depicted in Fig. 2.

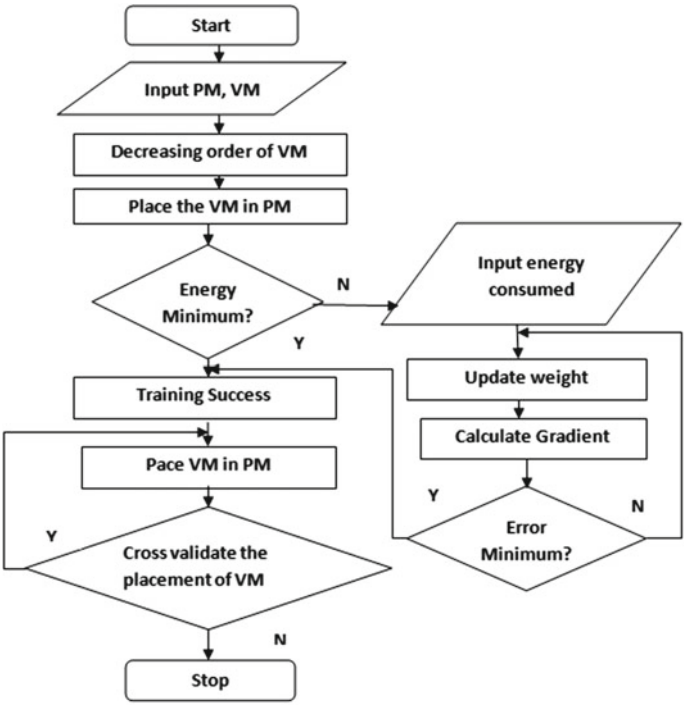


Fig. 2 Sequence diagram for neural network-based placement algorithm

Algorithm 2 Neural network based placement algorithm

Input: *PMList, VMList*
Output: *Efficient placement of VM*
1. for each PM in PMList do
2. Sort VMList of cpu for utilizing in decreasingorder;
3. for each VM in virtualmachineList do
4. minEnergy←∞;
5. placed PM ← Null;
6. for each PM in physicalmachineList do
7. if PM has adequate resources for VM then
8. if PM is postponed then
9. Energy ← estimate Increase in Energy (PM, VM (cpu)) by equ (4);
10. if energy <minEnergy then
11. place the VM in PM
12.Input Energyconsumed
13. Updated_Weight= Createweightiness Linear ax+b and Quadratic ax^2+bx+c
14. Epoch_Counter_value =n; Propagation_Counter_value=0;
15. While Epoch_value<Propagation_value || Gradient_Value<Updated_Weight
16. If the mse_value is minimum then
17. place the VM in PM;
18. End if;
19. Test_Set= energyConsumption;
20. Simulate (Test_Set, MSE, Placed VM, False Placement);
21. Evaluate Execution time of the algorithm;

The algorithm 2 shows how the placement algorithm-based NN operates by arranging the incoming virtual machine list entries according to CPU capacity utilization in ascending order. The PM checks for resource availability; if resources are not available, the PM is placed on a virtual machine (VM); this works as a best-fit method. The PM then determines the increase in energy usage following the deployment of the present VM. The PM with the smallest increase in energy use is subsequently assigned the VM. As indicated in Fig. 2, the input layer of the NN receives the energy used by each VM as input from I1 to In. The energy usage of every VM that has been put is then taken as raw data by a hidden layer that has been generated. The quadratic equation $ax2 + bx + c$ and the linear equation $ax + b$ are used to calculate the weight of the energy that has been consumed. Data processing occurs on the hidden layer between H1 and Hn. The neural network uses the energy consumed as training data. After that, the energy consumption of all the VM is added, and the mean of that number is used to calculate the gradient's value. It is then multiplied by the bias value, which is the network random number, to propagate the data in the feed forward NN.

The MSE is used as the performance metrics to determine error, and the linear and quadratic weights are then changed in accordance with the MSE in order to reduce error to the absolute minimum. Every VM constantly propagates this energy consumption with different uninformed constants (a, b, and c) in order to calculate the mean square error, which is then utilized to update weights.

It uses all available energy consumption using Eq.(4) data from all virtual machines to train the network until gradient and epoch requirements are not met. The one advancing pass in this three-layer feed forward NN is called the epoch. There have been 50 epochs taken. When the slope is satisfied or the specified number of epochs has been used, data propagation will halt.

The training of the neural network is finished when the gradient is satisfied. Consumed energy is passed during the initial iteration and will serve as a previous weight during the subsequent repetition.

The circulated energy consumption at the output layer is calculated by adding each time's new value to the preceding value. Epochs are terminated and the gradient will satisfy when the charge of the final element is greater than or equal to the slope value. The neural network training is carried out to determine the viability of the discovered solution and to cross-validate the inserted VM using false placement Eq. (3). There are two sub problems within the placement problem for energy efficiency. The first is to put the requested VM on the PM using Eq.(2), and the second is to make the current placement as effective as possible. However, in our NN-based placement approach, these issues are solved simply positioning the VM.

VM Placed

The logarithmic meaning of experimental output *xi* divided by *MSE_value* is 10 times multiplied by the placed VM. This parameter assists in keeping the correctly placed VM from turning false in the subsequent iterations, as indicated in Eq. (2)

$$\text{VM} = 10 * \log(xi)/\text{MSE\_Value} \tag{2}$$

False Placement

False placement is any placement that the execution algorithm does not expect, as well as any placement that does not take place because resources are not used effectively [17]. The outcomes of the ESCA are described by this parameter. The threshold value is used to find the false placement as described in Eq. (3).

$$\text{Threshold value} = \text{nweig}\big[\text{tweig}\big]X/100 \tag{3}$$

nweig (new weight) = Energy consumption in NN

tweig (total weight) = Total consumed Energy.

X is the percentage of threshold value like 25, 50, 75, 100%.

Energy Consumption

The energy consumption (ECP) by servers can be accurately described by a linear relationship between the energy consumption and CPU utilization [18]. In general, given a CPU utilization $v \in [0,1]$, let utilization of CPU v(t) be the function of time t, the energy consumed by the server can be denoted as:

$$\text{ECP}(v(t)) = \text{Estatic} + \text{Edynamic}.v(t) \tag{4}$$

where, v is the percentage of CPU utilization based on time (t), Estatic refers to static energy consumption which is independent of workload. Edynamic refers to the dynamic energy consumption based on time (t).

3.3 Enhanced Server Consolidation Algorithm (ESCA)

The proposed ESCA, based on NN, is mainly categorized into two sub problems, i.e., VM selection and VM placement. VM selecting and deploying servers on virtual machines, if the server is overloaded, trains a feed forward neural network with back propagation learning, uses a selection algorithm to determine which virtual machine to use, and finally uses a cross-validation algorithm to ensure the placement was accurate.

The dataset considered are the VM instance type provided by the Amazon EC2 [19], Microsoft Azure [20] and Google Cloud [21], as shown in Table 1. In this work, four different numbers of VM instance types have been chosen, including 25 VM, 50 VM, 75 VM, and 100 VM.

In ESCA, the input data is initialized and distributes the workload dynamically. Training is done in feed forward propagation. Depending upon the MSE error, the user request for resources is placed in PM.

If the error is not minimized, indicating that the server is overloaded, it selects the VM using a selection algorithm and allocates resources according to its demand using a placement algorithm. Thus, it always cross-validates the placement and replaces the VM, according to the current demand of resources, resulting in lower energy consumption and improved resource utilization, as shown in Fig. 3.

Table 1 Virtual machine instance type dataset

Cloud service provider	VM instances type	Number of VM instance types	CPU Capacity	Memory in GB
Microsoft Azure	1/H1V2	25	1	3.5
	2/H2V2	50	2	7
	3/H3V2	75	4	14
	4/H4V2	100	8	28
Google Cloud	1-C1-STD	25	1	3.75
	2-C1-STD	50	2	7
	3-C1-STD	75	4	15
	4-C1-STD	100	8	30
Amazon EC2	AEC2.small.1	25	1	2
	AEC2.large.2	50	2	8
	AEC2.xlarge.3	75	4	16
	AEC2.2xlarge.4	100	8	32

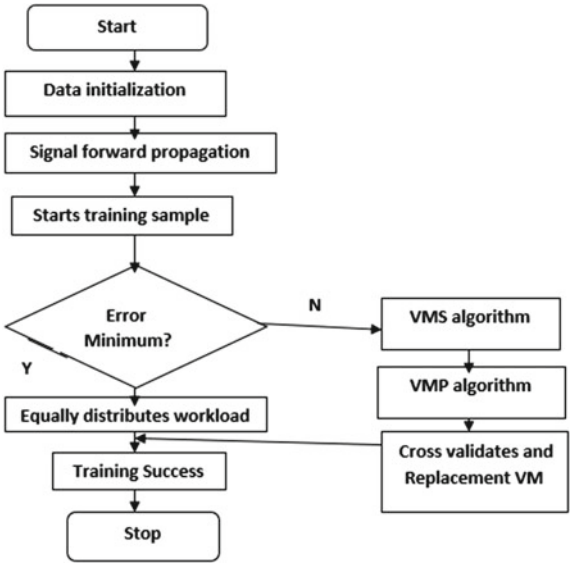


Fig. 3 Sequence diagram of an enhanced server consolidation algorithm

Algorithm 3 Enhanced Server Consolidation Algorithm

Input: *PMList, VMList*
Output: *Energy usage, Resource utilization and execution time*
1.Training network starts (network, epoch)
2.For epoch in search range
3.Forward propagate(network)
4. error ← estimated error;
5. if error <minError then
6. allocate the VMList in PMList;
7. else
8. select VMList using algorithm.1;
9. place VMList using algorithm.2;
10. backward propagate the VMList placement;
11. Training success;
12. allocate the VMList in PMList;
13. End;

The algorithm 3, which is provided, explains the fundamental idea of the increased server consolidation strategy. At first, servers are formed, virtual machines are assigned to the available servers, and virtual machines are given CPU capacity. Check for underload at each defined time period, identify the server that is being used less, put that server in sleep mode, and move the virtual machine to another active server. Locate the overloaded detection, choose the VM that will be migrated using Algorithm 1, and put the VM that will be placed on the available servers using Algorithm 2. Calculate the resource usage, energy use, and algorithm execution time at the conclusion of the simulation experiment.

4 Results and Discussion

Cloud computing enhances its performance by using an efficient server consolidation algorithm based on performance metrics. The metrics include execution time, energy consumption, resource utilization, reliability, scalability, and many others. The ESCA algorithm intends to attain the consolidation of servers with minimum energy consumption, execution time, and maximum resources utilization. To illustrate the feasibility and efficiency of the ESCA, the performance results of the algorithm are analysed and presented in this section.

The experiments have been conducted to evaluate the performance of the ESCA in Python-Spider IDE simulator. In ESCA algorithm, the MSE value is calculated for VM selection algorithm and VM placement algorithm. To optimize the propagation error in ESCA, Mean Square Error (MSE) value using Eq. (5) is taken as a parameter.

It is calculated as follows:

$$\text{MSE} = 1/n \sum_{i=1}^{n} (\text{pv} - \text{ov})^2 \tag{5}$$

MSE is a measure to quantify the difference between pv predicted values and ov observed values of the quantity being calculated and n is the size of the sample set. Data passes across the neurons from the input layer to the hidden layer and then to the output layer as the neural network learns from a specific dataset. The NN will then be trained using a back propagation technique to reduce error rates and reach the desired value. It is regarded as a supervised learning method as a result. The experiment uses the epoch value of 50. The term "epoch" refers to how many times the algorithm has seen the whole training dataset. Therefore, one epoch is finished each time the algorithm encounters the dataset to compute the output.

According to Fig. 4, it has taken six epochs to get the necessary output after the data has been passed through the algorithm both forward and backward. Figure illustrates the performance evaluation of minimum MSE value graph, which shows how the mean square error rate decreases gradually as the network learns. To identify the optimal selection of VM using NN model, the minimum value of MSE was employed.

The training process was conducted using a back propagation algorithm as the optimization procedure. The optimal MSE value is calculated as 201.02 using Eq. (5). The minimum MSE value of the VM is selected and migrated using a selection algorithm and cross-validation of placement in a placement algorithm. Minimizing MSE is a key criterion in selecting VM and placing VM. The number of resources utilized and energy consumed by the resources are calculated in ESCA algorithm, as shown in Table 2.

The number of VM instance types is taken as 25, 50, 75, and 100 VMs. The three datasets are implemented individually in ESCA algorithm, the resource utilization in numbers and energy consumption in KHW is obtained. Therefore, for ESCA

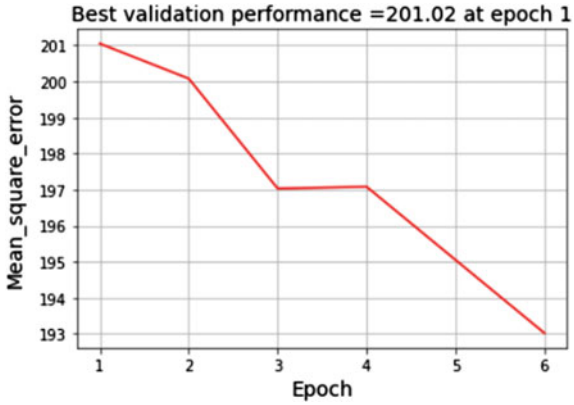


Fig. 4 Minimum mean square value

Table 2 Simulation result of enhanced server consolidation algorithm

Number of VM instance types	Amazon EC2		Google Cloud		Microsoft Azure	
	Resource Utilization	Energy Consumption	Resource Utilization	Energy Consumption	Resource Utilization	Energy Consumption
25	3	150.45	2	55.89	5	53.11
50	5	148.89	4	102.67	3	110.32
75	4	135.89	2	120.88	6	137.97
100	2	128.70	3	150.23	1	185.12
Total	**14**	**563.93**	**11**	**429.67**	**15**	**486.52**

algorithm, the maximum resource utilization is obtained in Microsoft Azure dataset and the minimum energy consumption is obtained as 429.67 in Google cloud dataset.

To determine the time complexity for a trained neural network with three layers, i, j, and k, where I stands for the number of input nodes, j for the number of hidden nodes, and k for the number of output nodes. The t training with n epochs, the result of the duration of network depends on the structure and numbers of layers in a neural network is $O(nt*(ij + jk + ki)$. Big O formula in the ESCA algorithm is used for faster and more efficient memory and timewise. The Big O is a mathematical operation that expresses the number of training and number of epochs increases the number of steps increase in algorithm. In ESCA algorithm, the time complexity for Amazon EC2 is 1.0 s, for Google Cloud is 2.0 s, and for Microsoft Azure is 1.0 s, hence the time complexity is less and same for Amazon EC2 and Microsoft Azure dataset. Experiments are carried out to compare the performance of ESCA with BFRU algorithm in [22] and VMBP algorithm in [23]. All three algorithms share the same objectives; therefore, they are compared to three different metrics namely the resource utilization in numbers and the energy consumption in KWH (Kilowatt per hour) and execution time in seconds.

In Best-fit resource utilization algorithm (BFRU), the constraint programming technique is applied for virtual machine placement in cloud environments. It is beneficial for combinatorial search problem, where solution must satisfy the constraints on relation between variables. The constraints are applied to allocate VM to PM. The goal of the BFRU algorithm is to lessen the amount of PM and energy use. The virtual machine bin packing (VMBP) algorithm is used to find actual mapping of VM to PM. It is possible to maximize the resource utilization by tightly packing the VM required to be running on to the least number of possible PM. The ESCA-based NN is used to improve resource utilization and lower data centre energy demand. The resource utilization required for placement of VM and the energy consumption in KWH and execution time of the ESCA, BFRU, and VMBP algorithms are given in Table 3. The comparison is done with Amazon EC2 datasets having 25,50,75 and 100 VMs.

Figure 5 shows the simulation outcome of the resource use of ESCA, BFRU, and VMBP for the Amazon EC2 dataset. The findings demonstrate that, in comparison to BFRU and VMBP, ESCA boosts resource consumption. The horizontal axis shows the various sets of VMs with various instance types that were taken into consideration for the trials. The vertical axis displays the actual resource usage as determined by algorithms for ESCA, BFRU, and VMBP.

Figure 6 shows the energy simulation results for the ESCA, BFRU, and VMBP for the Amazon EC2 dataset. It is discovered that the proposed ESCA algorithm consumes much less energy when compared to BFRU and VMBP. Since ESCA uses only 429.67 Kwh of energy at a minimum, the research's goal is met with this low energy usage.

Table 3 Results of resource utilization and energy consumption results for Amazon EC2

Cloud service provider	Number of VM instance types	Resource utilization			Energy consumption in KWH			Execution time in Sec		
		BFRU	VMBP	ESCA	BFRU	VMBP	ESCA	BFRU	VMBP	ESCA
Amazon EC2	25	4	2	3	160.38	175.88	150.45	2.2	1.8	1.0
	50	3	3	5	170.66	172.96	148.89			
	75	3	1	4	185.56	160.88	135.89			
	100	0	0	2	189.08	152.00	128.90			
	Total	10	6	14	649.22	661.72	563.93			

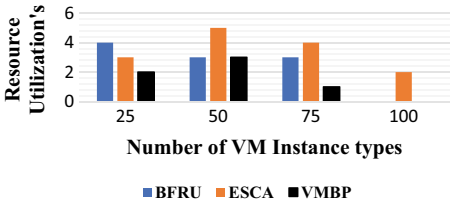


Fig. 5 Simulation results of resource utilization for the Amazon EC2 data sets

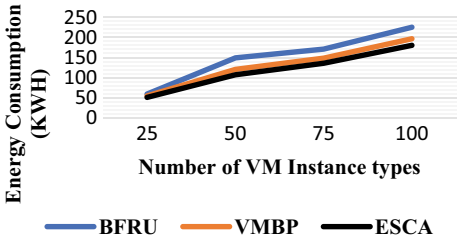


Fig. 6 Simulation results of energy consumption for the Amazon EC2 data sets

Similarly, the resource utilization and the energy consumption in KWH using ESCA, BFRU, and VMBP algorithms for Google cloud data sets are given in Table 4. The energy consumption of Google cloud datasets implemented with ESCA are 55.98, 102.67, 120.88, and 150.23. The simulation result of energy consumption of ESCA, BFRU, and VMBP for Google Cloud dataset is shown in Fig. 7. The results explore that BFRU and ESCA increases the resource utilization than VMBP.

The horizontal axis shows the various sets of VMs with various instance types that were taken into consideration for the trials. The vertical axis displays the actual resource usage as determined by algorithms for ESCA, BFRU, and VMBP.

The simulation result of energy consumption of ESCA, BFRU, and VMBP for Google Cloud dataset is shown in Fig. 8. When comparing ESCA with BFRU

Table 4 Results of resource utilization and energy consumption results for Google Cloud data sets

Cloud service provider	Number of VM instance types	Resource utilization			Energy consumption in KWH			Execution time in Sec		
		BFRU	VMBP	ESCA	BFRU	VMBP	ESCA	BFRU	VMBP	ESCA
Google Cloud	25	3	1	2	78.91	60.76	55.98	2.5	1.7	2.0
	50	2	3	4	129.56	120.45	102.67			
	75	1	2	2	150.41	138.44	120.88			
	100	5	4	3	201.55	166.21	150.23			
	Total	11	10	11	560.43	485.86	429.67			

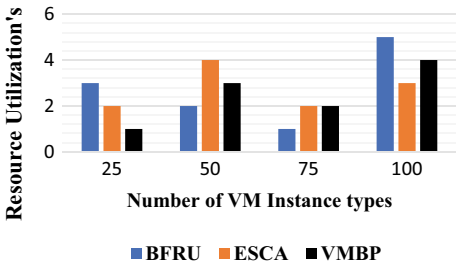


Fig. 7 Simulation results of resource utilization for the Google cloud data sets

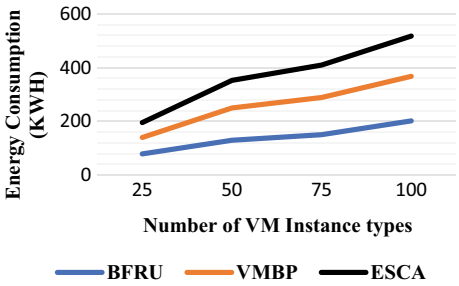


Fig. 8 Simulation results of energy consumption for the Google cloud data sets

and VMBP, it is found that the ESCA algorithm significantly reduces the energy consumption. Minimum energy consumption by ESCA is 429.67 Kwh.

Similarly, the resource utilization and the energy consumption in KWH using ESCA, BFRU, and VMBP algorithms for Microsoft Azure data sets are given in Table 5. Figure 9 shows the simulation outcome of the resource use of ESCA, BFRU, and VMBP using the Microsoft Azure dataset. The findings demonstrate that, in comparison to BFRU and VMBP, ESCA boosts resource consumption. The horizontal axis shows the various sets of VMs with various instance types that were taken into consideration for the trials. The real resource use of ESCA, BFRU, and VMBP is shown on the vertical axis. Figure 10 illustrates the energy simulation outcome for ESCA, BFRU, and VMBP using the Microsoft Azure dataset. It is discovered that the energy usage of the suggested ESCA algorithm is much lower when compared to BFRU and VMBP. A minimum of 486.52 Kwh of energy is used by ESCA. The investigations are done to analyse the effect of ESCA algorithm in overall resource utilization of all the active servers and the total energy consumption of all the active PM and the time taken for placing a VM, it is observed that the minimum time has been taken for placing a VM when compared with BFRU algorithm and VMBP algorithm.

The ESCA method, which is based on neural networks, is adaptable to the changing cloud environment. While neural networks may need some time to adapt to

Table 5 Results of resource utilization and energy consumption results for Microsoft Azure data sets

Cloud service provider	Number of VM instance types	Resource utilization			Energy consumption in KWH			Execution time in Sec		
		BFRU	VMBP	ESCA	BFRU	VMBP	ESCA	BFRU	VMBP	ESCA
AZURE	25	4	3	5	57.67	56.01	53.11	2.3	1.8	1.0
	50	3	2	3	147.25	125.67	110.32			
	75	5	4	6	150.57	138.86	137.97			
	100	2	0	1	235.11	198.55	185.12			
	Total	14	9	15	147.65	129.77	486.52			

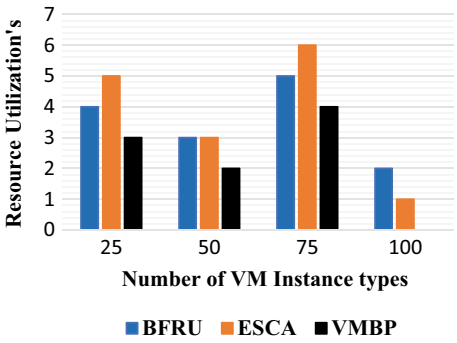


Fig. 9 Simulation results of resource utilization for the Microsoft Azure data sets

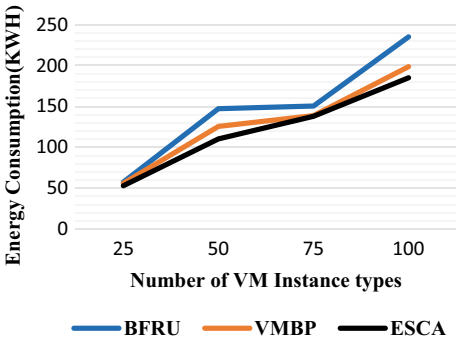


Fig. 10 Simulation results of eergy consumption for the Microsoft Azure data sets

sudden and abrupt changes, their strength lies in their ability to do so effectively, given the ever-changing nature of information. To reduce error rates and attain the desired value, the network is trained using the back propagation approach in the ESCA algorithm. Thus, the ESCA algorithm's time complexity with the Amazon EC2, Microsoft Azure, and Google cloud datasets is 1.0 s, 2.0 s, and 1.0 s, respectively, to reach the goal result. The training value is therefore the smallest possible value, and the epoch value is obtained as 50. When compared to the BFRU method and the VMBP algorithm, the ESCA algorithm has a greater time complexity. Consequently, the proposed algorithm achieves significant results with neural network.

5 Conclusion

Server consolidation techniques are a dynamic research area that has inspired a lot of excitement about how to efficiently map VM to PM, so that resources can be used to their maximum capabilities. In this research work, a new idea to hybrid a neural network and server consolidation namely an enhanced server consolidation algorithm with neural network is proposed, in which neural network predicts the

demand and thus place the resources according to that demand reduces the energy consumption of the data centres and it achieved high resource utilization by the way of using minimal number of PM. After running the simulation and comparing the results with the current BFRU and VMBP algorithms, it was discovered that using neural networks yielded better energy savings and system performance than the traditional strategy of consolidating servers.

References

1. Ahmad RW, Gani A, Hamid SHA, Shiraz M, Yousafzai A, Xia F (2015) A survey on virtual machine migration and server consolidation frameworks for cloud data centres. J Netw Comput Appl 52:11–25
2. Varasteh A, Goudarzi M (2015) Server consolidation techniques in virtualized data centres. IEEE Sys J
3. Liu X, Wu J, Virtual machine consolidation with minimization of migration thrashing for cloud data centers. In: Hindawi, Mathematical problems in engineering, vol 2020, Article ID 7848232, 13 pages
4. Zolfaghari R, Sahafi A (2021) Application of virtual machine consolidation in computing systems. Elsevier, Sustain Comput Inform Syst 30:100524
5. Halder K, Bellur U, Kulkarni P (2012) Risk aware provisioning and resource aggregation-based consolidation of virtual machines. In: Proceedings of IEEE 5th international conference CLOUD, pp 598–605
6. Prevost JJ, Nagothu K, Kelley B, Jamshidi M (2013) Optimal update frequency model for physical machine state change and virtual machine placement in the cloud. In: Proceedings of IEEE 8th international conference SoSE, pp 159–164
7. Vijaya C, Srinivasan P (2020) Hybrid technique for server consolidation in cloud computing environment. In: Cybernetics and information technologies, vol 20, no 1. ISSN: 1311-9702; Online ISSN: 1314-4081
8. Abohamama AS, Alrahmawy M, Elsoud MA. (2020) A hybrid energy Aware virtual machine placement algorithm for cloud environments. Elsevier Expert Syst Appl
9. Pandiselvi C, Sivakumar S (2018) A review of virtual machine placement algorithm in cloud data centres for server consolidation. Int J Eng Res Comput Sci Eng (IJERCSE) 5(3):182–188
10. Ferreto NMAS, Calheiros RN, De Rose CAF (2011) Server consolidation with migration control for virtualized data centres. Future Greener ComputSyst 27(8):1027–1034
11. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centres for Cloud computing. Future Greener Comput Syst 28(5):755–768
12. BeloglazovA, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual Machines in Cloud data centres. Concurrency Computer Practical Exp 24(13)
13. Monil M, Qasim R, Rahman RM (2014) Incorporating migration control in VM selection strategies to enhance performance. Int J Web Appl (IJWA) 6(4):135–151
14. Monil M, Rahman RM (2016) VM consolidation approach based on heuristics, fuzzy logic, and migration control. J Cloud Comput 5(1):1–18
15. Farahnakian F, Liljeberg P, Plosila J (2013) LiRCUP: linear regression-based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: 39th Euromicro conference on software engineering and advanced applications (SEAA), pp 357–364
16. Farahnakian F, Liljeberg P, Plosila J (2014) Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning. In: 22nd Euromicro international conference on parallel, distributed and network-based processing (PDP), pp 500–507

17. Xing X, Liu B, Ling D (2015) Neural network PID control based placement mechanism for cloud computing. Appl Math Inf Sci 9(2):789
18. Fathima A, Javid N, Sultana T (2018) Virtual machine placement via bin packing in cloud data centers. MDPI Electron 7:389. https://doi.org/10.3390/electronics/7120389
19. Amazon EC2, https://aws.amazon.com/ec2/instance-types/
20. Microsoft Azure, https://azure.microsoft.com/en-us/products/open-datasets/
21. Google cloud,https://cloud.google.com/datasets
22. Pandiselvi C, Sivakumar S (2018) Constraint programming approach based virtualmachine placement algorithm for server consolidation in cloud data centre. IJCSE 6(8)
23. Pandiselvi C, Sivakumar S (2021) Performance of particle swarm optimization bin packing algorithm for dynamic virtual machine placement for the consolidation of cloud server. In: IOP conference series: material science and engineering (Indexed by Scopus)

Low-Resource Indic Languages Translation Using Multilingual Approaches

Candy Lalrempuii and Badal Soni

Abstract Machine translation is effective in the presence of a substantial parallel corpus. In a multilingual country like India, with diverse linguistic origins and scripts, the vast majority of languages need more resources to produce high-quality translation models. Multilingual neural machine translation (MNMT) has the advantage of being scalable across multiple languages and improving low-resource languages via knowledge transfer. In this work, we investigate MNMT for low-resource Indic languages—Hindi, Bengali, Assamese, Manipuri, and Mizo. With the recent success of massively multilingual pre-trained models for low-resource languages, we explore the effectiveness of using multilingual pre-trained transformers—mBART and mT5 on several Indic languages. We perform fine-tuning on the pre-trained models in a one-to-many and many-to-one approach. We compare the performance of multilingual pre-trained models with multiway multilingual translation trained from scratch using a one-to-many and many-to-one approach.

Keywords NMT · Multilingual neural machine translation · Low resource · Mizo · Indic · BLEU

1 Introduction

Machine translation (MT) quality for bilingual language pairs has shown notable improvements with neural approaches like neural machine translation (NMT). Translation between pairs of high-resource languages with large corpora has made rapid advancements in terms of translation quality. However, low-resource languages still have a long way to go, given that NMT systems are frequently resource-intensive.

C. Lalrempuii (✉) · B. Soni
National Institute of Technology, Silchar, Assam, India
e-mail: candy_rs@cse.nits.ac.in
URL: http://cs.nits.ac.in/badal/

B. Soni
e-mail: badal@cse.nits.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_27

Due to the shortage of publicly available parallel data, researchers' attention has been directed to utilizing readily accessible resources, such as monolingual data. Another reason for employing monolingual corpora is that they are far more abundant than parallel corpora. They facilitate language model development for specific languages. Unsupervised NMT (UNMT) exploits the monolingual data for translation without any requirement for parallel data between the language pairs. Initially, UNMT assumes that no supervisory signal exists between the language pairs. Training solely depends on cross-lingual signals obtained through cross-lingual word embeddings. However, the shared representation between the languages restricts translation performance for distant languages. Furthermore, translation often requires multiple systems to be trained when numerous language pairs are incorporated into the MT system. This further led to the development of single NMT systems that can simultaneously train multiple languages, which can translate to and from *N* languages that are present during the training.

Multilingual neural machine translation (MNMT) has gained traction in recent years, which has shown improved translation quality for resource-constrained languages. MNMT accomplishes the task of translating multiple languages using a single model. In MNMT, the transfer of knowledge is facilitated by the joint training of diverse language pairs, which has been found beneficial for low-resource languages. Moreover, it utilizes available linguistic resources, streamlines the overall training process, and thus improves the generalizability and translation quality as a result of exposure to multiple languages. MNMT training is often carried out through transfer learning (i.e., high-resource to low-resource languages), pivot language, or multiway translation, where a single model simultaneously trains multiple languages for one-to-many, many-to-one, and many-to-many translations.

Recent success in large pre-trained language models like mBERT [1], mBART [2], and mT5 [3] has led to significant gains on tasks such as multilingual translation. Both these models employ an encoder-decoder architecture with several objectives like span corruption, and permutations. Low-resource languages have been found to benefit from these massively multilingual MT models. Motivated by the advantages of large pre-trained models, we leverage seq2seq multilingual pre-trained models like mBART and mT5 and explore their effectiveness on Indic languages. We specifically focus on Indic languages like Manipuri (*mni*), Assamese (*asm*), Bengali (*bn*), and Hindi (*hi*) in this study. Mizo and Manipuri belong to the Tibeto-Burman language family, while Assamese, Bengali, and Hindi belong to the Indo-Aryan language family. During the fine-tuning from pre-trained models, *lus, mni*, and *asm* are unseen languages that were not present during the pretraining stages. We also train a multilingual NMT system in a one-to-many and many-to-one approach from scratch. We evaluate the performance of the models using bilingual evaluation understudy (BLEU) and Character n-gram F-score (ChrF).

2 Related Works

Neural machine translation (NMT) has recently become one of the most prominent approaches for machine translation owing to its simplicity and robustness. The standard bilingual MT supports the translation of bilingual language pairs, often resulting in multiple systems for each language pair. Since the collection of bilingual resources is a laborious task, many researchers have resorted to using other linguistic resources, such as monolingual data. Multiple efforts have been made to improve the quality of machine translation results, including unsupervised approaches with back-translation and denoising autoencoders, semi-supervised and supervised approaches. Furthermore, transfer learning has been utilized to transfer knowledge in terms of learned parameters from high-resource to under-resourced language pairs.

There has been a recent surge in works incorporating multiple languages as opposed to only bilingual pairs for translation. Multi-task learning-based MT approach was proposed in which a single shared encoder for the source languages and separate decoders for each target language were used [4]. Similar to this architecture, a multiway NMT with an added attention mechanism was introduced using a many-to-many translation for the languages involved [5]. A single framework for multilingual systems was subsequently proposed to simplify the training process. A single model multilingual system that applied language-specific coding on subword tokenized input was introduced [6]. Google's MNMT system also uses a single model, which added a target language tag at the start of each sentence input sentence prior to training [7]. This also allowed zero-shot translation for unseen languages not present during the training. Furthermore, massively multilingual systems extended the support to more than 100 languages for training. A many-to-many translation model was trained using 102 languages in high-resource settings and 59 languages in low-resource settings [8]. The authors reported that the massive multilingual many-to-many models outperform the bilingual models for similar settings, while low-resource MT also benefits from the large models.

Some works on MT for Mizo languages have been reported. A comparative study on translations using PB-SMT and NMT was performed on English, and Mizo language pairs where the authors performed a detailed analysis based on fluency and adequacy [9]. Another work reported on NMT using various configurations and using the transformer architecture [10]. A transformer-based NMT system has also been reported with error analysis performed on statistical and modern neural approaches [11]. The tonal nature of the language has been explored, and an English-Mizo corpus has been published recently [12].

3 Background

This section describes the background of multilingual NMT.

3.1 Multilingual Neural Machine Translation

The main objective of multilingual translation systems is to enable translation to and from multiple languages. MNMT training can be carried out in different ways, including multiway translation [4, 5, 7], pivot language-based translation [13], and transfer learning [13] for low-resource languages. The multiway translation is typically a single model that can be used to devise one-to-many, many-to-one, or many-to-many translation systems. For source languages s_i, target languages t_i and $i \in N$ ($i = 1$ to N) where i is the selected language, and N is the number of languages used during training. The parallel data for the languages are denoted as $C(s_i)$ and $C(t_i)$. The training objective is to maximize the log-likelihood for the training data in all languages given as:

$$L_\theta = \frac{1}{N} \sum_{i=1}^{N} N^{C(s_i), C(t_i)}.\theta \tag{1}$$

Transfer learning is used for low-resource languages, which benefits from transferring the learned parameters from a high-resource language pair. A pivot language or a shared language pair is also used when no parallel corpus exists for a given language pair.

3.2 Multilingual Pre-trained Models

Self-supervised pretraining, such as BART [14] and T5 [15], has been widely used in a variety of language processing tasks. For MNMT, models like mBART [2], and mT5 [3] have been built upon BART and T5, respectively.

mBART uses a transformer-based encoder–decoder architecture that pre-trains on N number of languages. For monolingual data $D = D_1, D_2, ..., D_N$ and noise function g, it aims to predict the uncorrupted input text X given that $g(X)$. The objective is to maximize the log-likelihood as:

$$L_\theta = \sum_{D_i \in D} \sum_{X \in D_i} \log P(X|g(X); \theta), \tag{2}$$

where i is the selected language. The noise function g performs span text corruption and permutation of sentence order. mBART was originally trained in 25

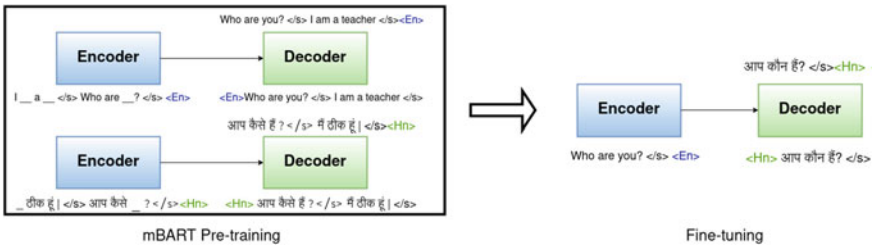


Fig. 1 Multilingual fine-tuning from pre-trained mBART model

languages. mBART50 is an extension carried out by pretraining on 50 languages which we also use in our experiments. Figure 1 depicts a bilingual fine-tuning performed by initializing an MT system with a large pretrained mBART model. mBART pretraining trains multiple unsupervised monolingual data, which is used to fine-tune downstream MT tasks. Fine-tuning is done on bilingual pairs English-Hindi, which is also extensible to multilingual pairs.

mT5 pushes the limit further by pretraining on the Common Crawl dataset of 101 languages. It closely follows the T5 encoder-decoder architecture using transformer models. Span corruption is also applied during pretraining, where spans of input text are masked.

Using these multilingual pre-trained models as an initialization for downstream tasks such as machine translation has shown to be beneficial. Moreover, in addition to fine-tuning on bilingual pairs, multilingual fine-tuning can be done to enable the pre-trained models to carry out the multilingual translation. Multilingual fine-tuning can be performed by adding the language tokens to the source and target side, which are then fine-tuned on systems initialized with the pretrained models.

4 Experimental Setup

In this section, we present the dataset description and experimental setup for multiway translation and multilingual fine-tuning on pre-trained models.

4.1 Data Description

We perform experiments on a dataset obtained from PMI dataset[1] which consists of English and 13 Indian languages. Of these, we focus on English (*en*), Manipuri (*mni*), Assamese (*asm*), Bengali (*bn*), and Hindi (*hi*). Since the Mizo language is not included in the PMI dataset, we use a general domain dataset for both training

[1] https://data.statmt.org/pmindia/v1/.

Table 1 Detailed statistics of parallel corpora for PMI dataset

Languages	Number of sentences $en \rightarrow xx$
mni	7419
asm	9732
bn	29584
hi	56831
lus	61738

and testing. For Mizo (*lus*), we use the parallel dataset manually collected from government websites and online blogs. However, the monolingual dataset is in-domain, which is crawled from online news websites like Times of Mizoram[2] and Zalen.[3] For the rest of the languages, monolingual data is also obtained from the PMI dataset. The statistics of parallel data are presented in Table 1. We use 500 and 1000 sentences for the validation and test sets, respectively, which are subsampled from the parallel dataset.

4.2 Data Preprocessing

During data preprocessing, the following steps were performed:

- Using Moses[4], we preprocess English and Mizo since Mizo also uses the Latin script similar to English. For other Indic languages, we use the Indic NLP library[5]. We perform normalization and tokenization on the data.
- We perform subword tokenization based on Byte Pair Encoding (BPE) using SentencePiece [16]. The vocabulary size is fixed at 32k.
- For many-to-one translation in a multilingual setting, we added a language tag at the source side indicating the target language as introduced in [7]. We do not perform transliteration into a unified script.

4.3 Training Settings

We conduct the experiments in two settings—multiway translation and fine-tuning on multilingual pretrained models like mBART and mT5.

[2] https://www.timesofmizoram.com/.

[3] https://zalen.in/.

[4] http://www.statmt.org/moses/.

[5] https://github.com/anoopkunchukuttan/indic_nlp_library.

1. Multiway Translation: For multiway translation, we carry out many-to-one translation using Fairseq [17]. We perform multilingual translations from Indic languages to English. A transformer model consisting of six layers and a total of 512 hidden units for both encoder and decoder is employed for training. The optimization is performed using Adam optimizer, where the learning rate is initially set to 0.001 with a dropout rate of 0.3. The models are trained using batches of 1024 tokens with 100k steps of training. We perform validation on the model using the BLEU and perplexity score on the validation set. The BPE tokenized subwords are rejoined during testing.

2. Multilingual Pre-trained Models: During fine-tuning of pretrained mBART, we use the mBART50, which has been pre-trained on 50 languages using Fairseq. We fine-tune the model on the unseen bilingual language pair English-Mizo. We employ the Adam optimizer with a 0.0001 learning rate, a dropout of 0.3, and an attention dropout of 0.1. For a maximum of 80K steps, we keep the maximum token count at 256. We also carry out fine-tuning on the mT5 pre-trained model for bilingual English-Mizo pair using Simpletransformers library[6] which has been built on the Huggingface [18] transformers. We utilize the mT5-base model and set the maximum sequence length to 100, the maximum steps to 100K, and the dropout rate to 0.1.

All of the models are trained on a single NVIDIA Quadro P4000 GPU.

4.4 Evaluation Measures

To assess the quality of translation results, we employ two automated evaluation metrics: bilingual evaluation understudy (BLEU) and Character n-gram F-score (chrF).

1. Bilingual Evaluation Understudy (BLEU): BLEU [19] is a modified n-gram precision score used to measure the similarity of translated output to reference translations. It counts the number of n-gram matches in the reference translation of the n-grams in the translated output, normalizing the n-grams by the number of occurrences in the translated output. For all our experiments, we use SacreBLEU [20] on detokenized translation outputs, n-gram matches up to 4-grams on a scale of 1–100.

2. Character n-gram F-score (chrF): We also use chrF [21] as an automatic evaluation measure to quantify the quality of translated output sentences. chrF is computed as character n-gram precision and recall, which is arithmetically averaged over all the n-grams. The character n-gram order is 6 and computed on a scale of 1–100.

5 Experimental Results and Discussions

This section gives the findings from our experiments utilizing the evaluation metrics from Sect. 4.4. Tables 2 and 3 present the automatic evaluation results in terms of BLEU and chrF scores, respectively, in both forward and backward directions for

[6] https://simpletransformers.ai/.

Table 2 BLEU scores for multiway translation and fine-tuning from pretrained **mBART** and **mT5** models

Model	lus		mni		asm		bn		hi	
	←	→	←	→	←	→	←	→	←	→
One-to-many (O–M)	–	15.6	–	9.2	–	10.5	–	12.1	–	16.3
mBART50 (O–M)	–	21.1	–	13	-	15.4	–	16.2	–	23.6
mT5 (O–M)	–	20.2	–	14.8	–	16.2	–	15.5	–	26.9
Many-to-one (M–O)	19.3	–	16.2	–	17.4	–	16.3	–	24.3	–
mBART50 (M–O)	23.4	–	17.2	–	19.3	–	25.8	–	31.7	–
mT5 (M–O)	24.5	–	16	–	20.1	–	24.6	–	32.4	–

Table 3 chrF scores for multiway translation and fine-tuning from pretrained **mBART** and **mT5** models

Model	lus		mni		asm		bn		hi	
	←	→	←	→	←	→	←	→	←	→
One-to-many (O–M)	–	33.46	–	24.32	–	27.14	–	30.06	–	35.43
mBART50 (O–M)	–	39.71	–	33.54	–	34.91	–	34.55	–	41.27
mT5 (O–M)	–	40.49	–	35.04	–	33.25	–	31.86	–	44.82
Many-to-one (M–O)	39.1	–	34.64	–	40.2	–	35.18	–	43.09	–
mBART50 (M–O)	40.85	–	37.93	–	38.21	–	43.14	–	47.12	–
mT5 (M–O)	42.11	–	35.02	–	40.45	–	41.90	–	48.53	–

each setting. The performance of the model with the highest BLEU and chrF scores is the mBART50 model in both settings. mT5 multilingual fine-tuning also comes close to mBART where for languages like *hi* and *asm*, mT5 performs relatively better. We note that, in general, the performance of fine-tuning from the pre-trained systems is superior to that of the multiway translation model. This is also consistent with previous works [22]. Languages *lus, mni*, and *asm* are unseen languages that were not present during the pretraining of mBART and mT5. However, fine-tuning using the pre-trained models on the MT task shows significant improvements in performance for both seen and unseen languages. This indicates that large multilingual models pre-trained on multiple languages learn from exposure to various languages during training, and translation knowledge is transferred further to the translation task during fine-tuning.

All languages used in our experiments-parallel and monolingual-are in-domain, except for *lus*, for which we employ parallel data from a generic domain for training and testing. The multilingual out-of-domain fine-tuning, however, appears to be helpful in knowledge transfer despite the languages not being present during pretraining since we see improvements in BLEU and chrF scores even for out-of-domain data.

6 Conclusions and Future Scope

In this paper, we present the results of multilingual NMT on a few Indic languages, using multiway translation and multilingual fine-tuning from large pre-trained models, namely mBART and mT5. We observe that multilingual fine-tuning significantly outperforms multiway translations trained from scratch without any pretraining. We perform a one-to-many and many-to-one translation for each setting. Our experimental results show that translation of knowledge from pre-trained models is beneficial, even for unseen languages. Furthermore, multilingual fine-tuning is also beneficial for unseen, out-of-domain data. In the future, we will explore the significance of language similarity in enhancing MNMT quality and addressing language divergences.

References

1. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers), Minneapolis, Minnesota, pp 4171–4186
2. Liu Y, Gu J, Goyal N, Li X, Edunov S, Ghazvininejad M, Lewis M, Zettlemoyer L (2020) Multilingual denoising pre-training for neural machine translation. Trans Assoc Comput Linguist 8:726–742
3. Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, Barua A, Raffel C (2021) mT5: a massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, (Online). Association for Computational Linguistics, June 2021, pp 483–498
4. Dong D, Wu H, He W, Yu D, Wang H (2015) Multi-task learning for multiple language translation. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, Vol 1 (Long Papers). Association for Computational Linguistics, Beijing, China, pp 1723–1732
5. Firat O, Cho K, Bengio Y (2016) Multi-way, multilingual neural machine translation with a shared attention mechanism. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, San Diego, California, pp 866–875
6. Ha TL, Niehues J, Waibel A (2016) Toward multilingual neural machine translation with universal encoder and decoder. In: Proceedings of the 13th international conference on spoken language translation, Seattle, Washington DC. International Workshop on Spoken Language Translation, Dec 8–9, 2016

7. Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, Thorat N, Viégas F, Wattenberg M, Corrado G, Hughes M, Dean J (2017) Google's multilingual neural machine translation system: enabling zero-shot translation. Trans Assoc Comput Linguist 5:339–351
8. Aharoni R, Johnson M, Firat O (2019) Massively multilingual neural machine translation. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, Vol 1 (Long and Short Papers), Minneapolis, Minnesota, pp 3874–3884
9. Pathak A, Pakray P, Bentham J (2022) English-mizo machine translation using neural and statistical approaches. Neural Comput Appl 31:7615–7631 Mar
10. Lalrempuii C, Soni B (2020) Attention-based english to mizo neural machine translation. In: Machine learning, image processing, network security and data sciences. Springer Singapore, pp 193–203
11. Lalrempuii C, Soni B, Pakray P (2021) An improved english-to-mizo neural machine translation. ACM Trans Asian Low-Resour Lang Inf Process 20
12. VMXEP Khenglawt, Laskar SR, Pal S, Pakray P, Khan AK (2022) Language resource building and English-to-mizo neural machine translation encountering tonal words. In: Proceedings of the WILDRE-6 workshop within the 13th language resources and evaluation conference, Marseille, France, European Language Resources Association, pp 48–54
13. Dabre R, Chu C, Cromieres F, Nakazawa T, Kurohashi S (2015) Large-scale dictionary construction via pivot-based statistical machine translation with significance pruning and neural network features. In: Proceedings of the 29th Pacific Asia conference on language, information and computation, Shanghai, China, pp 289–297
14. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2020) BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th annual meeting of the association for computational linguistics, (Online), pp 7871–7880
15. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(140):1–67
16. Kudo T, Richardson J (2018) SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations, Brussels, Belgium, pp 66–71
17. Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, Grangier D, Auli M (2019) Fairseq: a fast, extensible toolkit for sequence modeling. In: Proceedings of NAACL-HLT 2019: demonstrations
18. Wolf T, Debut L, Sanh J, Chaumond V, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush A (2020) Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, (Online), pp 38–45
19. Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics, ACL'02, Stroudsburg, PA, USA, pp 311–318
20. Post M (2018) A call for clarity in reporting BLEU scores. In: Proceedings of the 3rd conference on machine translation: research papers, Belgium, Brussels, pp 186–191
21. Popović M (2015) chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the 10th workshop on statistical machine translation, association for computational linguistics, Lisbon, Portugal, pp 392–395
22. Ramesh G, Doddapaneni S, Bheemaraj A, Jobanputra M, AK R, Sharma A, Sahoo S, Diddee H, J M, Kakwani D, Kumar N, Pradeep A, Nagaraj S, Deepak K, Raghavan V, Kunchukuttan A, Kumar P, Khapra MS, (2022) Samanantar: the largest publicly available parallel corpora collection for 11 indic languages. Trans Assoc Comput Linguist 10:145–162

DCC: A Cascade-Based Approach to Detect Communities in Social Networks

Soumita Das, Anupam Biswas, and Akrati Saxena

Abstract Community detection in social networks is associated with finding and grouping the most similar nodes inherent in the network. These similar nodes are identified by computing tie strength. Stronger ties indicate higher proximity shared by connected node pairs. This work is motivated by Granovetter's argument that suggests that strong ties lie within densely connected nodes and the theory that community cores in real-world networks are densely connected. In this paper, we have introduced a novel method called *Disjoint Community detection using Cascades (DCC)* which demonstrates the effectiveness of a new local density-based tie strength measure on detecting communities. Here, tie strength is utilized to decide the paths followed for propagating information. The idea is to crawl through the tuple information of cascades toward the community core guided by increasing tie strength. Considering the cascade generation step, a novel Preferential Membership method has been developed to assign community labels to unassigned nodes. The efficacy of *DCC* has been analyzed based on quality and accuracy on several real-world datasets and baseline community detection algorithms.

Keywords Social network analysis · Community detection · Information diffusion · Similarity measures

S. Das (✉) · A. Biswas
Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India
e-mail: soumita_rs@cse.nits.ac.in

A. Biswas
e-mail: anupam@cse.nits.ac.in

A. Saxena
Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: a.saxena@tue.nl

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_28

1 Introduction

Online social networks (OSNs) consist of inherent modular structures called communities where, nodes within a community are densely connected, and, nodes between communities are sparsely connected. Moreover, OSNs are predominantly used for information sharing because of its ability to connect geographically distant users. As information sharing occurs through social contacts, so the underlying network structure plays an important role in information propagation. Studying and analyzing the connections of the underlying network structure is vital for solving the problem of information diffusion and hence community detection. In OSNs, the strength of the connections shared by users is different. Numerous local similarity measures have been proposed to compute the strength of these connections using local neighborhood similarity, such as Jaccard Index (JI), Preferential Attachment (PA), and Salton Index (SA). These local similarity measures are particularly beneficial in community detection because it has low time complexity. For e.g., (α, β) algorithm utilizes JI to identify communities.

Social network analysis is predominantly associated with analyzing the interaction patterns among people, states, or organizations. These interactions among users help to reveal various important details of the underlying network structure [1]. The interactions in OSNs are dependent on the relationships shared by the connected users. These relationships are analyzed using tie strength measure. Strong ties cover densely knitted networks [2], and this idea is used to design a novel tie strength measure which is contingent on the neighborhood density of connected node pairs. Next, the tie strength is utilized to guide the interactions among individuals. Basically, DCC utilizes the interaction paths to reach the core of communities. Studies suggest that community cores are most densely connected [3]; so, if we start the diffusion process from any node and approach toward the community core, the tie strength goes on increasing. Tracing all the interaction patterns is used for ensembling groups of similar nodes. Therefore, our work shows the effectiveness of the proposed tie strength measure and information diffusion strategy on the identification of optimal communities. In this paper, the primary contribution is the introduction of a cascade-based method called Disjoint Community detection using Cascades (DCC) which shows the significance of tie strength, neighbors of neighbors, and information diffusion for detection of communities.

The rest of the paper is organized as follows. Section 2 discusses about the related work, Sect. 3 briefs about the proposed cascade-based community detection method, Sect. 4 discusses about the experimental setup, Sect. 5 discusses about the result analysis, and Sect. 6 concludes the paper.

Algorithm 1: Disjoint Community Detection Using Cascades

Input: $G(V,E)$, p, U
Output: $C = \{c_0, c_1, .., c_k\}$: set of communities
Procedure DCC ($G(V, E)$, p, U) **:**
- $A \leftarrow$ empty list
- $U \leftarrow$ list of all nodes
- $C_l \leftarrow$ empty list : stores lists of cascades
- $C \leftarrow$ empty list : stores lists of communities
- $p \leftarrow$ select any random node from U
- remove(p,U)
- add(p,A)
- $path\_length = 1$
- $q \leftarrow$ find_maxts(p, $\Gamma(p)$)
- **while** *len(U)>0* **do**
  - $r \leftarrow$ find_maxts(q,$\Gamma(q)$)
  - **if** $NS(p, q) <= NS(q, r)$ **then**
    - store p in A
    - remove (q, U)
    - $path\_length++$
    - $t \leftarrow q$
    - $q \leftarrow r$
  - **else if** $path\_length > 1$ **then**
    - store q in A
    - store A in C_l
    - make empty A
    - $t \leftarrow r$
  - **else**
    - $t \leftarrow q$:start new process with q
  - **if** *t in A and U not empty* **then**
    - $p \leftarrow$ select any random node from U
- Assign community labels to all respective cascades in C_l and store in C
- **while** *U not empty* **do**
  - $PM(u, c_j)$ computed using equation $\forall\, u \in U, \forall\, c_j \in C$
  - C updated with addition of unassigned nodes to respective communities
- **if** *two communities say, $c_1 \in C$, $c_2 \in C$ share atleast 1 node* **then**
  - Merge(c_1,c_2)

return C

Algorithm 2: Neighborhood Similarity

Input: $G(V,E)$, p, q
Output: ts: Tie strength value
Procedure NS (p, q) **:**
- **if** $\Gamma(p) == 1$ *or* $\Gamma(q) == 1$ **then**
  - $ts = 0$
- **else**
  - $ts = \frac{\rho p,q}{|\chi pq|}$
- **return** ts

2 Related Work

Increase in the size of social media users has made social network analysis very complex. Therefore, community detection task has been introduced to reduce the complexity of the original network in a substantial manner. Moreover, there are several potential applications of communities in OSNs such as it is used in rec-

Algorithm 3: Find Maximum Tie Strength

Input: $G(V, E)$, p, $\Gamma(p)$
Output: q: neighbor of p sharing maximum tie strength with p
Procedure `find_maxts(`$G(V, E)$, p, $\Gamma(p)$`)`**:**

```
max = 0
for q in Γ(p) do
    if NS(p, q) > max then
        max = NS(p, q)
return q
```

ommendation systems, trend analysis in citation networks, evolution of communities in social media, discovering fraudulent telecommunication network activities, and dimensionality reduction in pattern recognition. Therefore, several community detection techniques have been introduced till date to identify communities which are primarily classified into several approaches based on graph partitioning, clustering, modularity optimization, random walk, and diffusion community [4, 5]. Spectral bisection method is a graph partitioning technique which divides the graph into clusters based on density of links within a cluster and between clusters [6, 7], $Gdmp2$ [8] is a clustering technique where set of similar nodes are grouped together. It is usually of two kinds such as hierarchical clustering [9] and partitioning method of clustering [10, 11], Greedy-modularity (GM) [12], $Kcut$ [13] are modularity maximization techniques which are based on partitioning the graph based on the best modularity value [14–19], Diffusion Entropy Reducer (DER) [20] uses random walk technique where communities are detected by adopting a walker where the overall time is dependent on the density of communities [21, 22], Label Propagation Algorithm (LPA) [23] utilizes diffusion community method where similar nodes are grouped by propagating same action, property, or information in a network.

3 Proposed Method

The social contacts shared by an individual is indicative of some similarity possessed by the corresponding individuals, but mere connection is not enough to determine the most similar nodes present in the network. DCC addresses the role of tie strength and cascades in the identification of communities inherent in a network. In this section, we shall discuss the preliminary concepts that would be used throughout this paper followed by the discussion of DCC algorithm in detail.

3.1 Preliminaries

Suppose, we consider a graph $G(V, E)$ where V refers to set of nodes, E refers to set of edges. For any node $v \in V$, set of neighbors of v is denoted by $\Gamma(v)$, degree of node v is indicated by $\mid \Gamma(v) \mid$. Then, for a connected node pair $(v, u) \in V$, number of connections shared by common neighbors of v and u is indicated by $\mid \sigma(vu) \mid$.

Definition 1 (*Unprocessed Node*). Given a graph $G(V, E)$, a node $p \in V$ is an unprocessed node, if p is not yet activated during the diffusion process.

Definition 2 (*Common Neighborhood*). Given a graph $G(V, E)$ and a connected node pair say, $e_{p,q} \in E$, then common neighborhood is used to find the neighboring nodes related to p and q. Common neighborhood of (p, q) pair is defined by

$$\begin{aligned}\rho_{p,q} = {} & |\Gamma(p) \cap \Gamma(q)| + |\Gamma(p) \cap \Gamma(z)| + \left|\Gamma(q) \cap \Gamma(z)\right| + |\sigma_{pq}| \\ & + |\Gamma(w) \cap \Gamma(z)|, \quad \forall (w, z) \in (\Gamma(p) \cap \Gamma(q)) \\ & if\ e_{w,z} \in E,\ w \neq z,\end{aligned}$$

which indicates that $NS(p, q)$ is dependent on the degree exhibited by p and q. If either of the node's degree is 1, then $NS(p, q) = 0$.

Definition 3 (*Neighborhood Similarity*). Given a graph $G(V, E)$, neighborhood similarity of an edge say, $e_{p,q} \in E$ indicates the tie strength of p and q. It is defined by

$$NS_{p,q} = \frac{\rho_{p,q}}{\mid \chi_{p,q} \mid}, \tag{1}$$

where numerator term refers to common neighborhood of (p, q) pair and denominator term indicates number of nodes belonging to common neighborhood of p and q.

Definition 4 (*Cascade*). A cascade is a tuple $(p, find\_maxts(p, \Gamma(p), T)$ which contains information about a node p, neighbor of p with which p shares maximum NS score indicated by $find\_maxts(p, \Gamma(p))$ at a certain time T.

It is important to understand how cascades are generated during the diffusion process and how these are used for identifying communities present in the network. Therefore, it is required to understand the DCC algorithm to obtain a concrete idea of the community detection process. The details of the DCC algorithm are discussed below.

3.2 Disjoint Community Detection Using Cascades

It is a cascade-based Disjoint Community detection approach. DCC comprises three steps. Firstly, cascades are generated by computing and comparing tie strength based on neighborhood similarity measure. Secondly, Preferential Membership method is proposed to assign community labels to the unprocessed nodes, and thirdly, merging step where communities sharing common nodes are merged. Let us now try to understand each of the steps with the help of pseudocodes and pictorial example.

Cascade Generation: The path followed during information diffusion process is dependent on a novel tie strength measure called neighborhood similarity (NS). Tracing the path generated during the diffusion process results in a set of cascades as shown in Fig. 1. Let us try to understand the cascade generation step illustrated in the first while loop in Algorithm 1 with the help of the cascade generation example on the simple graph as shown in Fig. 1. Suppose, node 13 (indicated by red colored node) initiates the diffusion process. Then, node 13 tries to activate its maximum NS value neighboring node obtained using $Find\_maxts$. The illustration of $Find\_maxts$

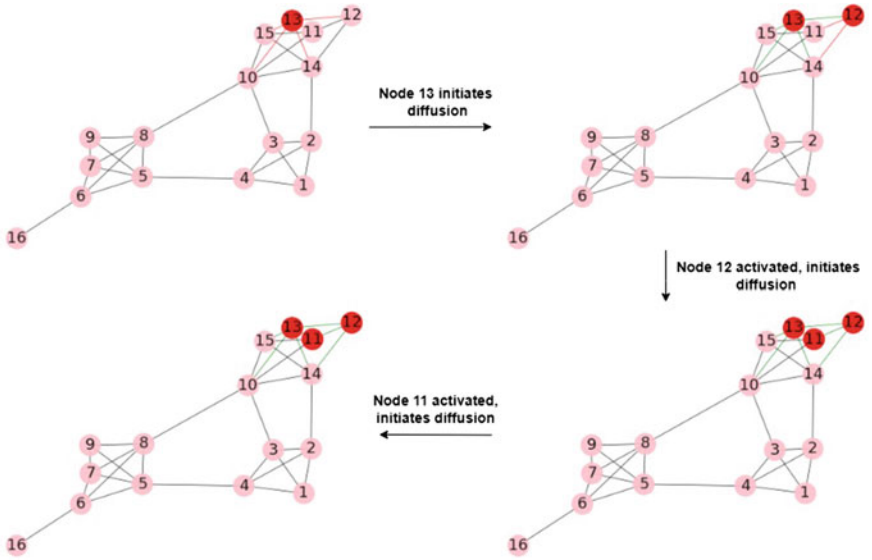


Fig. 1 Demonstration of cascade generation step of DCC algorithm using a simple graph. Peach colored nodes indicate inactive nodes, red colored nodes indicate active nodes, red arcs represent edges where active nodes try to activate their inactive neighbors, green arcs represent edges propagated once. Initially, node 13 initiates diffusion process, computation of $Find\_maxts(13, \Gamma(13))$=12 with $NS(13, 12) = 0.8$, so node 12 activated and initiates diffusion, computation of $Find\_maxts(12, \Gamma(12))$=11, $NS(12, 11) = 1.0$ and $NS(13, 12) <= NS(12, 11)$, node 11 is activated and initiates diffusion, computation of $Find\_maxts(11, \Gamma(11))$ gives no neighbor of node 11 that shares greater tie strength than $NS(12, 11)$, cascade obtained is [13, 12, 11]

is shown in Algorithm 3. Computation of $Find\_maxts(13, \Gamma(13))$ gives node 12 with $NS(13, 12) = 0.8$. Next, node 12 is activated. Next, node 12 tries to activate its neighboring nodes indicated with red arcs. The task is to identify the neighboring node say r such that $Find\_maxts(12, \Gamma(12)) = r$ (say) and $NS(13, 12) <= NS(12, r)$. We find r=node 11 with $NS(12, 11) = 1.0$, and hence, node 11 is activated, indicated with red color. Now, node 11 tries to find its maximum NS value neighbor such that its tie strength is greater than or equal to $NS(12, 11)$. But, no such suitable neighboring node is obtained, and hence, the cascade obtained is [13, 12, 11]. Next, all cascades for the remaining unprocessed nodes are obtained by repeating the above mentioned procedure. At the end of the cascade generation step, a list of cascades are obtained which are assigned with corresponding community labels. Next, labels are assigned to remaining unlabeled nodes using Preferential Membership (PM).

Definition 5 (*Preferential Membership*). Given graph $G(V, E)$, set of communities C; then, Preferential Membership is used to assign community membership $c_j \in C$ to an unlabeled node, $p \in V$ when $\arg\max_j \; PM(p, c_j), \; \forall c_j \in C$. It is defined by

$$PM(p, c_j) = \sum_{\substack{\Gamma(q) \neq p, \\ q \in \Gamma(p), \\ q \in c_j}} \frac{| \Gamma(p) \cap \Gamma(q) |}{| \Gamma(p) | \times | \Gamma(q) | .} \tag{2}$$

Nodes that are yet to be labeled are assigned with corresponding community labels using Eq. (2). Let us try to understand the membership assignment with an example. Suppose, we assign a community label c_1 to the cascade [13,12,11] as obtained from the cascade generation process. Now, node 10 (say) is one of the unprocessed node, then using Eq. (2), we compute $PM(10, c_1)$. Considering this equation, we select node 13 which is one of the neighbors of node 10. Moreover, node 13 is also in c_1. Neighbors of node 13 is indicated by

$$\Gamma(13) = \{10, 12, 11, 14, 15\}.$$

Next, to compute Eq. (2), we need

$$\begin{aligned}
\Gamma(12) &= \{11, 13, 14\}. \\
\Gamma(11) &= \{10, 12, 13, 15\}. \\
\Gamma(14) &= \{2, 10, 12, 13, 15\}. \\
\Gamma(15) &= \{10, 11, 13, 14\}.
\end{aligned}$$

Now, putting these values in Eq. (2), we obtain

$$\begin{aligned}
PM(10, c_1) &= \frac{| \Gamma(10) \cap \Gamma(12) | + | \Gamma(10) \cap \Gamma(11) | + | \Gamma(10) \cap \Gamma(14) | + | \Gamma(10) \cap \Gamma(15) |}{\Gamma(10) \times \Gamma(13)} \\
&= \frac{3 + 2 + 2 + 3}{6 \times 5} \\
PM(10, c_1) &= 0.333.
\end{aligned}$$

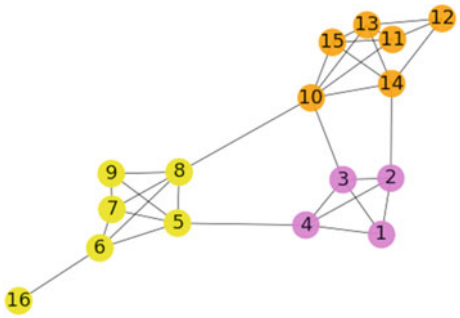


Fig. 2 Communities obtained by DCC algorithm on karate dataset. Three different colors indicate three different communities obtained by incorporation of DCC algorithm

Merging: Merging is incorporated to obtain the final community set. Communities sharing at least one common node are merged. The final set of communities obtained by DCC algorithm on the example graph is shown in Fig. 2. Therefore, incorporation of DCC algorithm gives three set of communities.

4 Experimental Setup

In this section, we shall discuss about the experimental setup. Here, experiments are conducted to evaluate the comparative performance of DCC with respect to the baseline community detection algorithms. Evaluation is carried from three perspectives such as community detection algorithms, real-world datasets, and evaluation metrics. We have selected community detection algorithms that are based on network structure, modularity optimization, random walk, and neighborhood information of nodes.

Community detection algorithms: Algorithms are based on diffusion such as Label Propagation Algorithm (LPA) [23]; modularity maximization-based algorithms such as Greedy-modularity (GM) [12] and $Kcut$ [13]; random walk-based algorithm such as Diffusion Entropy Reducer (DER) [20] and $Gdmp2$ [8] based on clustering nodes. These algorithms are selected to analyze and compare the performance of DCC in terms of modularity, neighborhood information of nodes, and cascade information. Moreover, the evaluation of communities is carried in two perspectives such as quality and accuracy. Evaluation of community quality is performed in terms of number of internal and external connections. Quality evaluation does not require ground truth information. Whereas, accuracy evaluation requires ground truth information. The following evaluation metrics have been considered for our experimentation purpose.

Evaluation metrics: Quality metrics based on internal connections only such as NGM, Modularity Density, Z Modularity; external connections-based quality metrics such as Cut_Ratio have been used. Moreover, accuracy metrics such as Normalized

Table 1 Dataset statistics

Dataset	# Nodes	# Edges	Avg. degree	Dataset	# Nodes	# Edges	Avg. degree
Riskmap [24]	42	83	3.95	Dolphin [25]	62	159	5.12
Karate [26]	34	78	4.58	Strike [27]	24	34	3.16
Football [28]	115	613	10.66	Sawmill [29]	36	37	3.44

First column contains dataset details, # $Nodes$ refers to number of nodes, # $Edges$ refers to number of edges, and Avg. degree indicates average degree of the graph

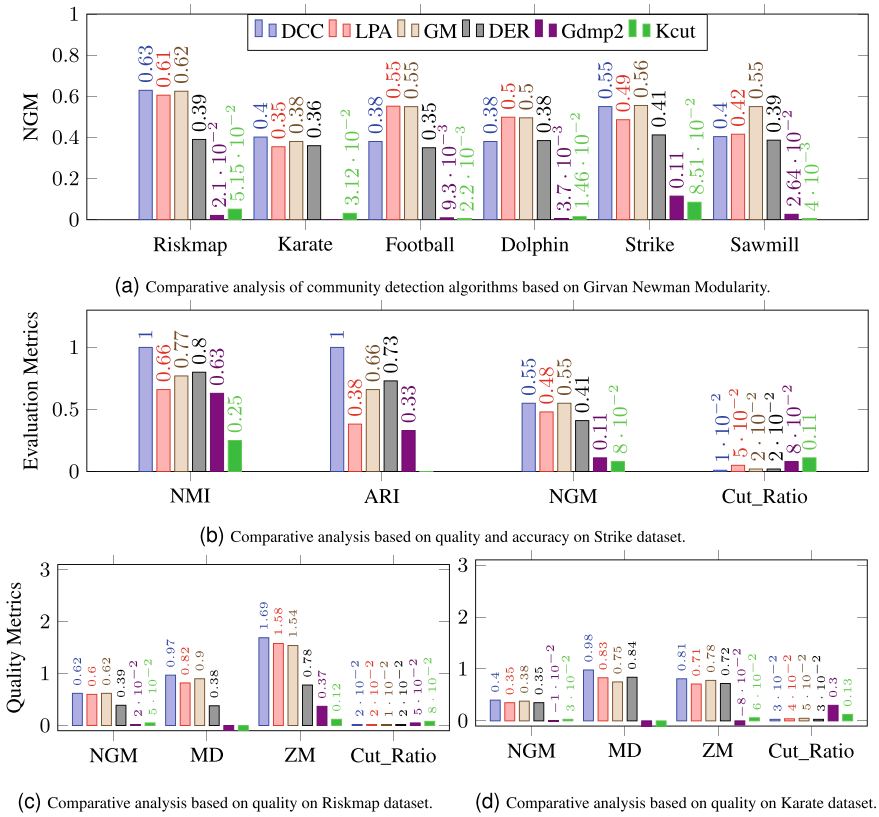
Mutual Information (NMI) and Adjusted Random Index (ARI) have been used [14, 30–32]. Next, the above mentioned community detection algorithms are tested on several real-world datasets such as riskmap, karate, football, dolphin, strike, and sawmill are summarized in Table 1. These datasets are publicly available in online repositories such as SNAP [33]. The reason to select these datasets is availability of ground truth information and for ease of performance evaluation by visualization.

5 Result Analysis

The comparative results of DCC algorithm with respect to the baseline algorithms considered in this paper have been represented in Fig. 3. Before discussing about the results obtained by incorporation of several evaluation metrics, let us first try to interpret the result of DCC on karate dataset. DCC gives three set of most densely connected communities on karate dataset. From this result, we can say that DCC works excellently to identify all groups of densely connected nodes irrespective of the size of such groups.

Let us try to comprehend the results of DCC obtained by incorporation of several evaluation metrics one by one. Firstly, if we consider the result represented in Fig. 3a, DCC gives the best Newman Girvan Modularity (NGM) score on riskmap, karate, and strike datasets. Whereas, the results on football, dolphin, and sawmill are comparative low. The reason for this is that DCC algorithm explicitly identifies the densely connected group of nodes without considering the number of nodes in the corresponding group. Hence, low modularity value does not infer low quality communities. The good performance of DCC is also justified by the results obtained on the remaining quality metrics on football, dolphin, and sawmill network.

Consider the results represented in Fig. 3b, clearly DCC gives the maximum NMI, ARI, NGM score, and minimum Cut_Ratio score as compared to the baseline algorithms on strike dataset. Therefore, from these results, it is obtained that DCC shows excellent performance in terms of quality and accuracy on strike dataset. Similarly, the results of DCC in terms of these quality and accuracy metrics on other datasets are quite good. Also, results based on different variants of modularity such as NGM, MD, ZM, and results based on Cut_Ratio on riskmap and karate dataset as shown in Fig. 3c and d respectively are implication of the excellent performance of DCC. Also, though we have not used any modularity-based optimization con-



(a) Comparative analysis of community detection algorithms based on Girvan Newman Modularity.

(b) Comparative analysis based on quality and accuracy on Strike dataset.

(c) Comparative analysis based on quality on Riskmap dataset.

(d) Comparative analysis based on quality on Karate dataset.

Fig. 3 Comparative analysis based on different evaluation metrics on real-world datasets, NGM: Newman Girvan Modularity, MD: Modularity Density, ZM: Z Modularity

cept in *DCC* algorithm, but the excellent modularity results are self-explanatory of the effectiveness of neighborhood similarity measure and Preferential Membership method.

6 Conclusion

In this paper, a novel tie strength guided cascade generation approach for community detection called *DCC* has been developed. Depending on cascades that are generated, a new method called Preferential Membership has been designed. The interpretation of communities obtained by *DCC* algorithm assures its ability to identify densely connected communities irrespective of the size of such communities. We have considered six real-world datasets, five baseline algorithms, four quality evaluation metrics, and two accuracy metrics for performance evaluation. The results given

by *DCC* confirm effectiveness of the proposed tie strength measure, cascade generation strategy, and Preferential Membership method. In the future, we shall examine the performance of *DCC* on large real-world networks and synthetic networks and examine its performance.

References

1. Das S, Biswas A (2021) Deployment of information diffusion for community detection in online social networks: a comprehensive review. IEEE Trans Comput Soc Syst 8(5):1083–1107
2. Van der Leij M, Goyal S (2011) Strong ties in a small world. Rev Network Econ 10(2)
3. Yanchenko E, Sengupta S (2022) Core-periphery structure in networks: a statistical exposition. arXiv preprint arXiv:2202.04455
4. Chand S, Mehta S (2017) Community detection using nature inspired algorithm. In: Hybrid intelligence for social networks. Springer, pp 47–76
5. Khan BS, Niazi MA (Aug 2017) Network community detection: a review and visual survey. arXiv preprint arXiv:1708.00977
6. Pothen A (1997) Graph partitioning algorithms with applications to scientific computing. In: Parallel numerical algorithms. Springer, pp 323–368
7. Barnes ER (1982) An algorithm for partitioning the nodes of a graph. SIAM J Algebraic Discrete Methods 3(4):541–550
8. Chen J, Saad Y (2010) Dense subgraph extraction with application to community detection. IEEE Trans Knowl Data Eng 24(7):1216–1230
9. Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction, vol 2. Springer
10. Hlaoui A, Wang S (2004) A direct approach to graph clustering. Neural Networks Comput Intell 4(8):158–163
11. Bezdek JC (2013) Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media
12. Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111
13. Ruan J, Zhang W (2007) An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In: Seventh IEEE international conference on data mining (ICDM 2007). Omaha, NE, USA, IEEE, pp 643–648
14. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
15. Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
16. Boettcher S, Percus AG (2002) Optimization with extremal dynamics. Complexity 8(2):57–62
17. Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582
18. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press
19. Jin D, Yu Z, Jiao P, Pan S, He D, Wu J, Yu P, Zhang W (2021) A survey of community detection approaches: from statistical modeling to deep learning. IEEE Trans Knowl Data Eng
20. Kozdoba M, Mannor S (2015) Community detection via measure space embedding. Advances in neural information processing systems, p 28
21. Hughes BD et al (1995) Random walks and random environments: random walks. Oxford University Press
22. Zhou H (2003) Distance, dissimilarity index, and network community structure. Phys Rev E 67(6):061901

23. Cordasco G, Gargano L (2010) Community detection via semi-synchronous label propagation algorithms. In: 2010 IEEE international workshop on: business applications of social network analysis (BASNA). IEEE, pp 1–8
24. Cheng J, Leng M, Li L, Zhou H, Chen X (2014) Active semi-supervised community detection based on must-link and cannot-link constraints. PloS one 9(10):e110088
25. David L, Karsten S, Oliver JB, Patti H, Elisabeth S, Steve MD (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405
26. Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473
27. Michael JH (1997) Labor dispute reconciliation in a forest products manufacturing facility. For Prod J 47(11/12):41
28. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
29. Michael JH, Massey JG (1997) Modeling the communication network in a sawmill. For Prod J 47(9):25
30. Miyauchi A, Kawase Y (2016) Z-score-based modularity for community detection in networks. PloS one 11(1):e0147805
31. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174
32. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
33. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, Accessed 27 March 2021

Fault Classification and Its Identification in Overhead Transmission Lines Using Artificial Neural Networks

Kathula Kanaka Durga Bhavani and Venkatesh Yepuri

Abstract In modern power systems, fault classification and placement are critical for improving protection schemes, service reliability, and reducing line outages. However, due to the mutual coupling effect, it confronts usual issues in fault identification in double-circuit lines. Although several methods have been shown to be accurate in locating double-circuit line faults, they have limitations in certain situations such as a lack of synchronization between the measuring ends, identification of the type of fault, data before and after a fault, three-phase faults, and so on. This study offers an artificial neural network technique that uses a probabilistic neural network (PNN) for fault classification and a generalized regression neural network (GRNN) for fault localization to address some of these challenges. To categorize and detect the fault, the proposed technique leverages the fundamental current phasor magnitudes obtained at both circuit endpoints of the double circuit. To validate the proposed method, a 200 km overhead transmission line is simulated using MATLAB/SIMULNK for all sorts of faults by varying location of the fault, resistance of the fault, and inception angle of the fault.

Keywords 200 km overhead transmission line · Fault location · Neural networks (NN) · Current phasor magnitude · MATLAB/SIMULINK

1 Introduction

Transmission lines are structures that are used in power systems to deliver electrical energy across great distances. Transmission lines can be configured as single-circuit transmission lines, double-circuit transmission lines, or multi-terminal transmission lines based on the tower structure and conductor design. In modern power systems, double circuits are used where greater consistency and security is required.

K. K. D. Bhavani · V. Yepuri (✉)
Department of Electrical and Electronics Engineering, Swarnandhra College of Engineering and Technology, Seetharampuram, West Godavari District, Andhra Pradesh 534280, India
e-mail: venkatesh.yepuri555@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_29

This double-circuit lines enable the transfer of more electrical power with less cost. However, running two circuits in parallel will cause mutual coupling between the conductors and also will increase additional problems in protection schemes. Fault location is a method that aims to accurately locate the occurred fault. Quick identification of faults in transmission lines is needed, otherwise they can destroy the whole power system. The purpose of fault location is to improve service reliability, save time and money for repair, and reduce line outages and economic losses. There are various ways for locating faults in double-circuit transmission lines, including impedance-based methods [1–6], high frequency-based methods [7], travelling wave method [8] and knowledge-based methods [9–13]. Out of this, the direct, simple, economical way of calculating the fault location is achieved by using the impedance-based method. Liao et al. [2] used current measurements from only one or two branches, but in this algorithm, a more comprehensive fault classification is required. Ning et al. [3] suggested an algorithm that leverages synchronized voltage measurements using PMUs and harnesses voltage readings from one or more branches. Behnam et al. [4] proposed an impedance-based fault location based on negative sequence voltage, but this method does not hold good for three-phase faults. Kale et al. [7] used a combination of neural network and wavelet transform consisting of time frequency analysis of fault-generated transients. Jiang et al. [8] proposed an improved fault line identification method based on initial process of zero sequence currents and voltage travelling waves. Gracia et al. [9], ANN structure is applied for classification and localization of faults in overhead transmission lines. The best structure is carried out by a software tool called SARENEUR. Anamika et al. [10] discussed an algorithm to classify and locate the fault by using only single end data. Despite the fact that it is sensitive to distant in feed, it is not relevant to all faults.

All of the aforementioned approaches have certain drawbacks, such as the requirement for synchronization measurements, the categorization of fault types, the lack of before-and-after fault data, and the inability of some of them to find three-phase faults. ANN is utilized to categorize and find the error in order to get around some of these issues. PNN (probabilistic neural network) is used to categorize faults by utilizing pattern recognition. PNNs are a type of radial bias network that perform well for classification issues, whereas GRNNs (Generalized Regression Neural Networks) are used to locate faults. A GRNN employs the function estimate extracted from the training to approximate any arbitrary function between input and output vectors. Using MATLAB and SIMULINK, a conventional system of a 200 km, 400 kV double-circuit transmission line is simulated, tested, and the findings are determined to be correct.

The following is how this paper is structured. Section 1 provides an introduction, Sect. 2 describes the usage of artificial neural networks in power systems, and Sect. 3 discusses the suggested strategy. Section 4 compares the numerical simulation findings for a double-circuit transmission line using an Artificial Neural Network to those of Behnam et al. [4]. Section 5 closes with conclusions.

2 Methodology

In order to estimate functions that are normally unknown, artificial neural networks (ANNs), which are computer models based on biological neural networks, are utilized. These are a subset of learning algorithms and nonlinear statistical models. Due to its effectiveness at pattern recognition, classification, generalization, and fault tolerance, ANN is frequently utilized in power systems. Offline data may be used to train ANN. They have excellent features such as generalization capability, immune to noise, and robustness. Load forecasting, fault diagnosis/location, economic dispatch, security evaluation, and transient stability are now the most popular ANN implementation areas.

2.1 Proposed PNN and GRNN for the Classification of Fault and Its Location

This part develops ANN-based fault diagnosis and classification for long transmission lines in power systems. Here, classification and location are found by using current phasors as inputs. The current phasors, in three phases at four bus bars, are taken as total inputs and training is done by using the offline phasors for a large number of various fault conditions. The number of conditions increases the accuracy rate for fault location and classification.

2.2 Classification of Faults

Robust and precise algorithms are required because fault type classification plays a significant role in relay protection for transmission lines in power networks. So, a new method is proposed and implemented here by using probabilistic neural network. The PNN inputs here are the magnitudes of three-phase currents measured at both ends of line1 and line2. Thus, there are twelve inputs for each line. Figure 1 displays the block diagram of proposed PRNN. Input vector = [ia1, ib1, ic1, ia2, ib2, ic2, ia3, ib3, ic3, ia4, ib4, ic4].

PNN requires a significant number of fault patterns to understand the problem in order to determine the kind of fault. By altering distinct fault sites, fault resistance, and fault inception angles, 1407 fault patterns are used to train the PNN. Now, the

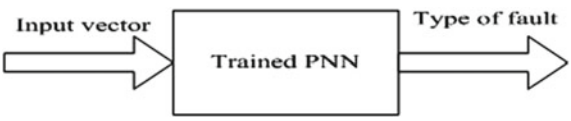


Fig. 1 Schematic of proposed probabilistic neural network using fault classifier

equivalent outputs for faults of types AG, BG, CG, AB, BC, CA, ABG, BCG, CAG, and ABC are 1, 2, 3, 5, 6, 7, 8, 9, and 10, respectively.

$$\text{Output Vector} = [1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9, 10].$$

When the aforementioned input and output data patterns are used to train the PNN, it is discovered that the PNN accurately categorizes all sorts of errors in each circuit.

2.3 Fault Location

A generalized regression neural network (GRNN) is used in the suggested technique to accurately find the caused fault since it is crucial to do so in order to preserve the efficient and reliable functioning of the power system. The current magnitudes of both healthy and faulty line at two ends serve as the inputs for GRNN. Figure 2 displays the block diagram utilizing the GRNN.

$$\text{GRNN input} = [\text{ia1},\ \text{ib1},\ \text{ic1},\ \text{ia2},\ \text{ib2},\ \text{ic2},\ \text{ia3},\ \text{ib3},\ \text{ic3},\ \text{ia4},\ \text{ib4},\ \text{ic4}]$$

So the total number of inputs is twelve. The GRNN is trained for 1407 different fault location conditions and training is done for every 3 km distance of total 200 km line length. Since an efficient training is provided here, it must be able to locate the fault for nearer and farer distance, also which will not be done in [4]. The output for the GRNN is the fault location distance.

$$\text{The output vector} = [\text{fault location distance}]$$

The following flow chart presented in Fig. 3 uses probabilistic neural networks and extended regression neural networks to provide clear information about fault location and fault categorization (PNN).

Figure 3 depicts the architecture for an ANN-based fault classifier and problem locator. When the input vector is fed to a previously trained GRNN network under various fault situations, the function between the input and output.

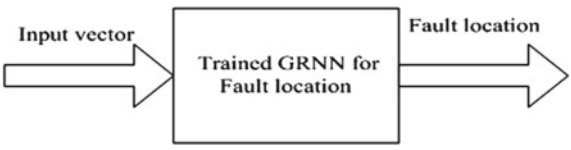


Fig. 2 Block diagram of GRNN-based fault classifier

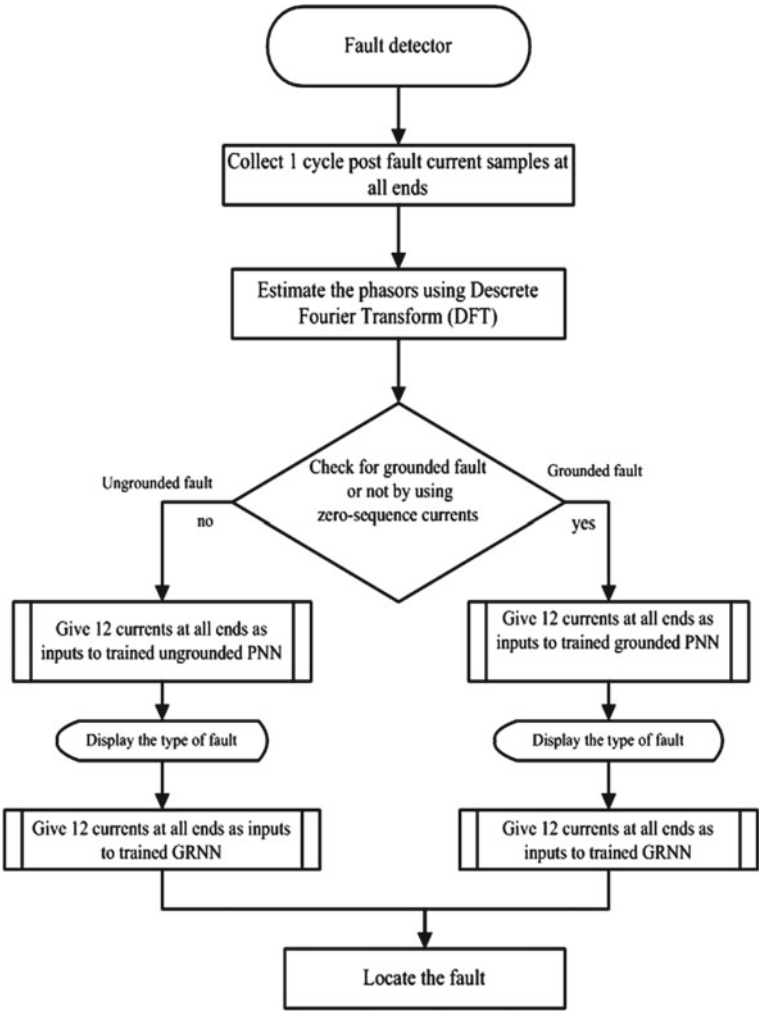


Fig. 3 Illustration for finding a defect in a double-circuit transmission line

3 Simulation Results

3.1 Unsymmetrical Fault (AG Fault)

In line-2, 100 km from the relay-S, we assume a single line-to-ground fault (AGF) with a resistance of fault about 0.01 ohms and an inception of fault time of 0.02 s. The 0.02 fault is shown as a three-phase current and voltage phasor in Fig. 4. Using voltage and current values collected from relays during inception of fault, it demonstrates three-phase supply with constant magnitude in no-fault conditions. A line-to-ground

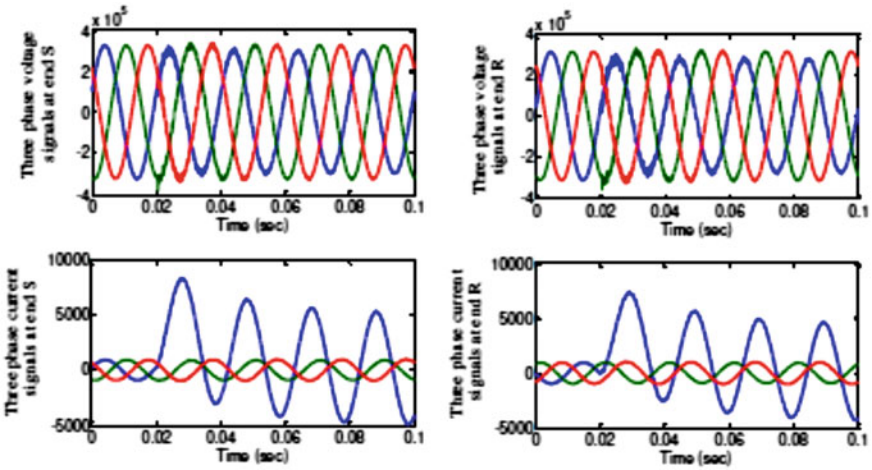


Fig. 4 Three-phase voltage and current signals at both ends for an AG fault 50 km from the relays, with a fault inception time of 0.02 s

fault develops in the "b" phase, and its magnitude suddenly rises close to the fault's initiation angle.

The MATLAB model performs several test cases by altering the fault type, fault resistance, and fault inception angle to simulate unsymmetrical faults using trained probabilistic neural networks and generalized neural networks. We found that it correctly categorizes and locates the error in each case. The discrepancy between the actual fault site distance and the projected fault location distance is displayed in Table 1. The error rate is under 0.2%.

It is evident from the following table that Generalized Regression Neural Network for Unsymmetrical Faults produces correct results.

3.2 Symmetrical Faults (ABC Fault)

We presupposed a 3-Ø defect (ABC) in line-2, located 100 km from the relay-S, with an inception time of 0.02 s, and a fault resistance of 0.01 ohms. Figure 5 depicts the fault that forms at 0.02 s as a 3-Ø current and voltage phasor.

With voltage and current values acquired from relays during inception of fault, it demonstrates three-phase supply with constant magnitude during no-fault situations. A three-phase supply with constant magnitude is seen in Fig. 5, when there is no fault. When a three-phase fault is produced, it manifests itself in all phases and its magnitude abruptly increases in all three phases at the fault inception angle. Many test cases are run on the three-phase fault for various fault resistance and fault inception angles for symmetrical faults utilizing the generalized neural network in the MATLAB model. According to Table 2, there is a very little discrepancy between

Table 1 ANN technique fault localization findings for various conditions of fault under unsymmetrical fault

S. no.	Type of fault	Fault resistance	Fault inception angle	Actual distance	Estimated distance	Error (%)
01	AG	0.01	0	50	49.1310	0.08
			90	50	49.4349	0.282
		50	30	50	49.0358	0.482
			60	50	50.0390	0.0195
02	AB	5	120	50	49.6073	0.196
			90	50	49.9907	0.0046
		25	180	50	50.0023	0.0015
			20	50	50.3830	0.1917
03	ABG	10	40	50	50.0855	0.042
			60	50	49.388	0.306
		100	0	50	50.4306	0.215
			100	50	49.2182	0.309

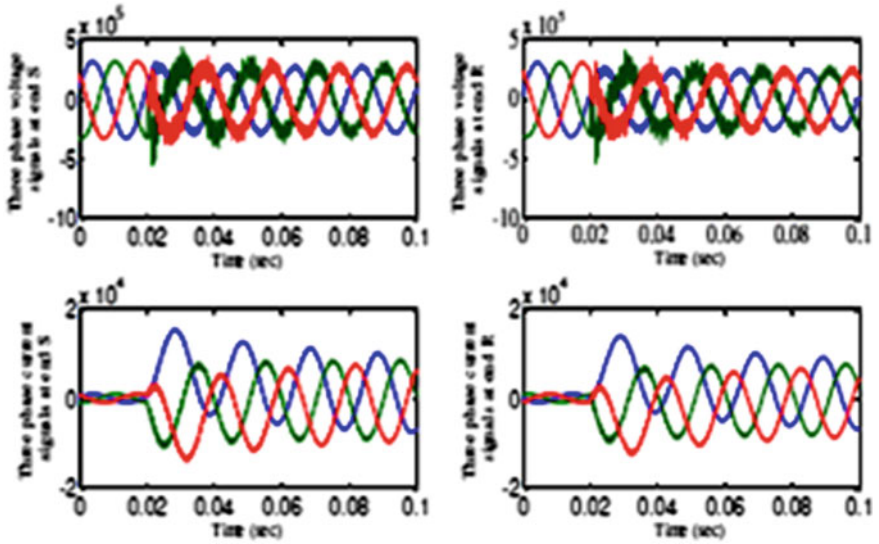


Fig. 5 3-Ø voltage and current signals at both ends for an ABC fault 50 km from the relays, with a fault inception time of 0.02 s

the actual fault site distance and the projected location of fault distance. The error rate is under 0.2%.

The aforementioned table demonstrates that the suggested ANN-based approach provides results that are quite accurate for symmetrical problems with a low error percentage. For both symmetrical and unsymmetrical errors, this ANN-based technique produces accurate findings.

The suggested technique provides correct localization findings for all types of defects in Table 3 for all case studies with a very low error percentage, i.e., within 0.2%. According to this strategy, faults can be found in both symmetrical and unsymmetrical fault situations.

Table 2 Fault location results for different fault situations under symmetrical fault using ANN method

S. no.	Type of fault	Fault resistance	Fault inception angle	Actual distance	Estimated distance	Error (%)
01	ABC	0.01	0	20	19.8988	0.0506
				72	72.5496	0.2748
			30	150	150.510	0.255
		50	90	80	79.6233	0.188
				75	75.3315	0.1675
			180	110	109.432	0.284
				150	150.577	0.2885

Table 3 Comparison of unsymmetrical and symmetrical fault locations

Type of fault	Actual distance (kms)	Fault resistance (Ω)	Inception angle (°)	Method in [4]	Error (%)	Proposed method	Error (%)
BG	40	0.01	0	37.93	1.03	39.90	0.05
		5	90	38.83	0.58	39.19	0.40
ABG	20	0.1	60	18.93	0.53	20.45	0.22
		50	120	19.05	0.47	19.86	0.07
BC	10	0.01	10	10.77	0.38	9.861	0.06
		25.0	170	9.113	0.44	9.772	0.11
ABC	20	0.01	30	27.39	3.69	20.00	0.00
		50	90	23.4	1.7	20.76	0.38

4 Conclusions

The mutual coupling effect makes it difficult and complex to locate faults in double-circuit wires. The need for synchronization and the lack of pre- and post-fault data are still problems with certain earlier techniques for identifying faults in double-circuit transmission lines. This study uses artificial neural networks to build a fault classification and localization method for double-circuit transmission systems. The proposed approach uses the latest phasor magnitudes from the ends of both lines. The methodology offers accurate results for diverse fault scenarios. It is shown how precisely the simulation in the MATLAB environment operates and how well it can find both symmetrical and non-symmetrical faults.

References

1. Saha MM (2001) New accurate fault location algorithm for parallel lines. In: 7th International conference on developments in power systems protection (DPSP 2001), IEEE, Netherlands, pp 407–410
2. Kang N, Liao Y (2009) New fault location technique for double-circuit transmission lines based on sparse current measurements. In: 41st North American power symposium, IEEE North America, pp 1–6
3. Kang N, Liao Y (2012) Double-circuit transmission-line fault location with the availability of limited voltage measurements. IEEE Trans Power Deliv 27(1):325–336
4. Mahamedi B, Zhu JG (2014) Unsynchronized fault location based on the negative-sequence voltage magnitude for double-circuit transmission lines. IEEE Trans Power Deliv 29(4):1901–1908
5. Kawady T, Stenzel J (2003) A practical fault location approach for double circuit transmission lines using single end data. IEEE Trans Power Deliv 18(4):1166–1173
6. Song G, Suonan J, Ge Y (2009) An accurate fault location algorithm for parallel transmission lines using one-terminal data. Int J Electr Power Energy Syst 31(2–3):124–129
7. Kale VS, Bhide SR, Bedekar PP (2009) Faulted phase selection on double circuit transmission line using wavelet transform and neural network. In: 2009 International conference on power systems, IEEE India, pp 1–6
8. Jiang B, Dong X, Shi S, Wang B (2015) Fault line identification of single line to ground fault for non-effectively grounded distribution networks with double-circuit lines. In: 2015 IEEE power and energy society general meeting, IEEE Denver, pp 1–5
9. Gracia J, Mazon AJ, Zamora I (2005) Best ANN structures for fault location in single- and double-circuit transmission lines. IEEE Trans Power Deliv 20(4):2389–2395
10. Jain A, Thoke AS, Koley E, Patel RN (2009) Fault classification and fault distance location of double circuit transmission lines for phase-to-phase faults using only one terminal data. In: 2009 International conference on power systems, IEEE India, pp 1–6
11. Jain A, Thoke AS, Patel RN (2009) Double circuit transmission line fault distance location using artificial neural network. In: 2009 World congress on nature and biologically inspired computing (NaBIC), IEEE India
12. Aggarwal RK, Xuan QY, Dunn RW, Johns AT, Bennett A (1999) A novel fault classification technique for double-circuit lines based on a combined unsupervised/supervised neural network. IEEE Trans Power Deliv 14(4):1250–1256
13. Dalstein T, Kulicke B (2015) Neural network approach to fault classification for high-speed protective relaying. IEEE Trans Power Deliv 10(2):1002–1011 (2015).

An Improved Way to Implement Round Robin Scheduling Algorithm

Kuldeep Vayadande, Aditya Bodhankar, Ajinkya Mahajan, Diksha Prasad, Riya Dhakalkar, and Shivani Mahajan

Abstract The Scheduling algorithm is primarily responsible for the system's effectiveness. The Round Robin (RR) algorithm is considerably more effective than the currently available ones. Every task in the Round Robin algorithm is given a set amount of time to complete the part of execution. The process needs to be taken out of ready queue if it is finished, otherwise it's added to the end of ready queue, and waits for the process's execution turn. In this chapter, we propose a technique related to the dynamic time quantum approach of Round Robin that works well. The dynamic time quantum (TQ) approach autonomously figures out the time quantum for each cycle based on a prescribed formula. In the proposed technique, TQ is calculated in each cycle and is compared to each process burst time for its execution. Further, the dynamic TQ reduced the Average Waiting Time, Average Turnaround Time, and Number of Context Switches.

Keywords Scheduling algorithm · Round robin · Dynamic time quantum · Context switching · Turnaround time

K. Vayadande · A. Bodhankar (✉) · A. Mahajan · D. Prasad · R. Dhakalkar · S. Mahajan
Vishwakarma Institute of Technology, Pune, India
e-mail: aditya.bodhankar21@vit.edu

A. Mahajan
e-mail: narendra.ajinkya21@vit.edu

D. Prasad
e-mail: diksha.prasad21@vit.edu

R. Dhakalkar
e-mail: riya.dhakalkar21@vit.edu

S. Mahajan
e-mail: vijay.shivani21@vit.edu

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_30

1 Introduction

Process scheduling refers to both the processes and techniques for assigning CPU resources to various processes as well as the methodology for allocating CPU resources to distinct processes. It is one of the basic properties an operating system to multitask. Maximizing CPU efficiency is the most basic and utmost goal in the process of CPU scheduling. Throughput, Number of context switching (NCS), Average Waiting Time (AWT), and Average Turnaround Time (ATT) of these processes are used to determine the efficiency.

- **Average Turnaround Time**: The reference of the term ATT is the mean of time consumed by processes waiting present in queue for execution till the CPU is allocated to them.
- **Average Wait Time**: AWT is described as the average time spent (usually in ns) by processes as they wait in ready queue for the CPU for being assigned to them for execution.
- **Throughput** can be described as the rate at which all procedures are being completed.
- **Time Quantum** (TQ) is the amount of time a preemptive multitasking system allows a process to run.

The switching between multiple processes for CPU is context switching. This helps the processes to execute fairly (Fig. 1).

CPU scheduling algorithms are divided into two types: preemptive algorithms and non-preemptive algorithms. Preemptive algorithms focus on priority of the processes. The process that is higher in priority has the power to block the process that is being

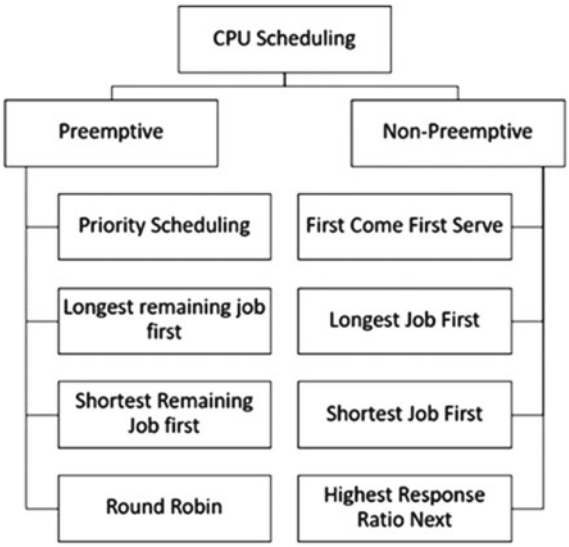


Fig. 1 CPU scheduling algorithms

executed and has low priority. Also, the non-preemptive algorithms allocate the CPU to the process until the completion of its execution [1].

In case, if there are numerous processes in a ready queue, choosing which one will receive the CPU is a key challenge. There are several Scheduling Algorithms available to handle this issue, which determine how processes are assigned to the CPU. These algorithms include Round Robin Algorithm of Scheduling, SP, SJF, and FCFS which are elaborated below:

- **First Come First Serve (FCFS)**: As implied by the name, FCFS executes the jobs present in ready queue having similar sequence as they are received. This method has a fault in that shorter processes usually arrive after longer ones, which concludes in long average waiting, turnaround times.
- **Shortest Job First (SJF)**: Here, the processes that entered the ready queue are originally arranged in line with their Burst Times (BT) in the ascending order. The CPU is then sequentially assigned to the processes. In most situations, this algorithm is superior to others. However, it is difficult realistically to have a prior knowledge of the BT, of all the processes in SJF.
- **Priority Scheduling (PS)**: Processes are given priorities and placed in a ready queue as part of PS. The OS allocates processor to the process that has highest priority. Later on, the processor is given to another task with a comparatively lower priority, in loops. Instead of concentrating on CPU efficiency restrictions, the PS method corresponds the level of priority of each operation. Therefore, it may produce the best or worse cases depending on the burst timings of the necessary operations. Both the FCFS and SJF algorithms are non-preemptive. However, PS can be anyone of both preemptive or non-preemptive. If a process can continue when a higher priority process arrives, it falls under the category of non-preemptive scheduling.
- **Round Robin (RR)**: One of the best algorithms for real-time and time-sharing systems is the RR algorithm. This algorithm's main building block is a brief time slice that is allotted to each process for the duration of operation. If in case, within this time, the execution ends or completes, the processor is then given the next process and the current process is withdrawn from the maintained queue. If the execution is not complete, the processor gets the next process to execute. The term "time quantum" refers to this little period of time, that is measured in nanoseconds or milliseconds. Determining a mean time quantum is a crucial problem since the RR method depends entirely on it.

In RR algorithm, the main concern is about choosing right TQ. If the value is too small, it leads the process to wait for more time and increase in Number of Context Switch (NCS) and if it is too large, it leads the algorithm to work as FCFS. So, it is important to choose correct TQ for the cycles to make the algorithm work efficiently. Many algorithms for RR have been proposed that choose TQ efficiently to reduce the AWT and ATT. Algorithms are also clubbed with other algorithms which reduce it too, but some algorithms lead to increase in complexity of algorithms.

2 Literature Survey

Bishat et al., in this paper [2], suggested and modified a method for the RR algorithm by computing the time slice which was based on the BTs of the different processes which were present in ready queue for execution. The methodology proposed by authors directly impacted the turnaround time (TAT), response time, WT, and numeric value of context switches. Additionally, extended the time quantum for processes that take somewhat longer to complete than the allotted time quantum cycle(s).

Sohrawordi et al., in this paper [3], leverage Dynamic TQ to boost the algorithm's effectiveness. The ready queue processes that made up the collection of burst times (BT) were used to determine the time quantum. In comparison to the original RR algorithm, the algorithm reduced the AWT by solving the fixed time quantum problem.

Alsulami et al., in this paper [4], observed a few diverse Round Robin (RR) algorithms like Adaptive RR, Best Time Quantum RR CPU scheduling, Optimal RR scheduling, and Improved RR scheduling algorithm. The AWT, ATT, average reaction time, and the amount of context flips are some of the metrics that the authors have used to test the performance of all these approaches. The simulation outcomes prove Because they output the lowest values of the performance parameters, Adaptive RR and Optimal RR scheduling employing Manhattan Distance algorithms are both far more effective to use.

Alaa et al., in this paper [5], proposed a model for Cloud Computing process scheduling using improved Round Robin algorithm. This algorithm used varying time quantum for round robin. The authors have given comparison between their proposed algorithm and the traditional RR algorithm and the Improved RR Algorithm with different time quantum. One among the most crucial concerns in a cloud environment is task scheduling, as tasks must be applied to the right virtual machines while taking multiple aspects into account at once. This makes the task scheduling problem an NP-complete problem. The RR CPU process scheduling method with variable time quantum is presented in this article. The suitable method has been shown to be superior to the established RR and IRRVQ algorithms. The outcomes demonstrate that the proposed approach has a shorter WT and TAT than the conventional RR and the IRRVQ.

Farooq et al., in this paper [6], focused on creating an effective RR algorithm utilizing the variable quantum unit of time. The goal was to minimize the executing duration of the algorithm as well as efficiency parameters such as normal WT and TAT. Reduced context switching, typical waiting and turnaround times, and increased operating system efficiency lead to improved embedded systems. Some of these systems have already been developed, but since sorting the processes is so challenging, they require other methods and operate much slower.

Tang et al., in this paper [7], introduced Lazy Round Robin, a variant of the original Round Robin utilized in real systems that is easier to construct and also has less run time overhead compared to the regular Round Robin algorithm. The main difference between the traditional Round Robin and the Lazy Round Robin is the reaction time

of the scheduler to the freshly given instances of tasks. The Round Robin scheduling algorithm determines if some lately released instance of task is execution eligible in the remnant part of the recent round, whereas, the Lazy Round Robin scheduling algorithm delays response of any of the task releases until conclusion of the recent round.

Rosita et al., in this paper [8], suggested one model on multicore processor systems. The researchers improvised the native Round Robin algorithm on multicore systems. The authors made calculations of time quantum referenced on average BT. The processes in the ready queue were transferred to each core for parallel execution. The given methodology decreases the average waiting-time for each process.

Balharith et al., in this paper [9], reviewed the types of Round Robin algorithms. The authors studied the classification of algorithms that was based on static and dynamic time quantum. The authors presented various aspects of the Scheduling tasks in processors. They mainly classified the types on basis of RR algorithm in CPU and RR algorithm in Cloud Computing. The survey presented by authors acted as base start for research for development of new algorithms in an improved manner.

Zongyu et al., in this paper [10], presented a model which makes use of cluster web servers effectively and fairly through load balancing is an essential task, especially with the development of the dynamic content and database-driven internet applications like e-commerce and corporate databases. The authors of this work developed the load balancing method, that uses the philosophy of prediction to load balancing, to solve this issue. The efficiency of the suggested load balancing technique has been checked using simulated data. The approach significantly lowers both the load range as well as the load variation in comparison to the RR scheduling and MRR scheduling.

Mora et al., in this paper [11], suggested a model for the RR algorithm named the Modified Media Round Robin Algorithm (MMRRA). Instead of emphasizing average waiting time, this method concentrates on average response time. The authors suggested a method that results in determination of time quantum that utilized the max BT as well as the median of the BT of all processes present inside the ready queue. For each cycle of this algorithm, the time quantum has been determined one time in a static way.

3 Proposed Approach

In this study, we have proposed the classic round robin's driving principle that every process should have an opportunity to run after a certain amount of time. The methodology will undoubtedly decrease the average wait time for high priority tasks. Depending on the collection of processes, the overall average waiting duration for every process present in the ready queue may decrease.

As mentioned above, the Time Quantum (TQ) selected for the processes determines how effective the RR algorithm is. A small TQ leads to increase in NCS, AWT, and ATT, whereas an increased TQ causes the algorithm to perform like FCFS. If

the TQ is not chosen correctly, the processes left with few units of BT are left in the waiting queue for their next turn which increases waiting time.

For example, if a TQ is 40 for a cycle and the remaining time of a process is 42 units, then the process has to wait for the next cycle to complete its execution. This led to increase in the waiting time of the processes. This directly affects the performance of the schedulers.

To overcome these problems, we propose an approach where the algorithm chooses TQ such that the processes can be completed with minimum Context Switches with reduced AWT and ATT. The algorithm is set up so that the TQ for the current process is changed to the remaining BT of the current process if there is 10% or less of the current TQ left in the process. The proposed approach tends to give more efficient output without using other algorithms (Fig. 2).

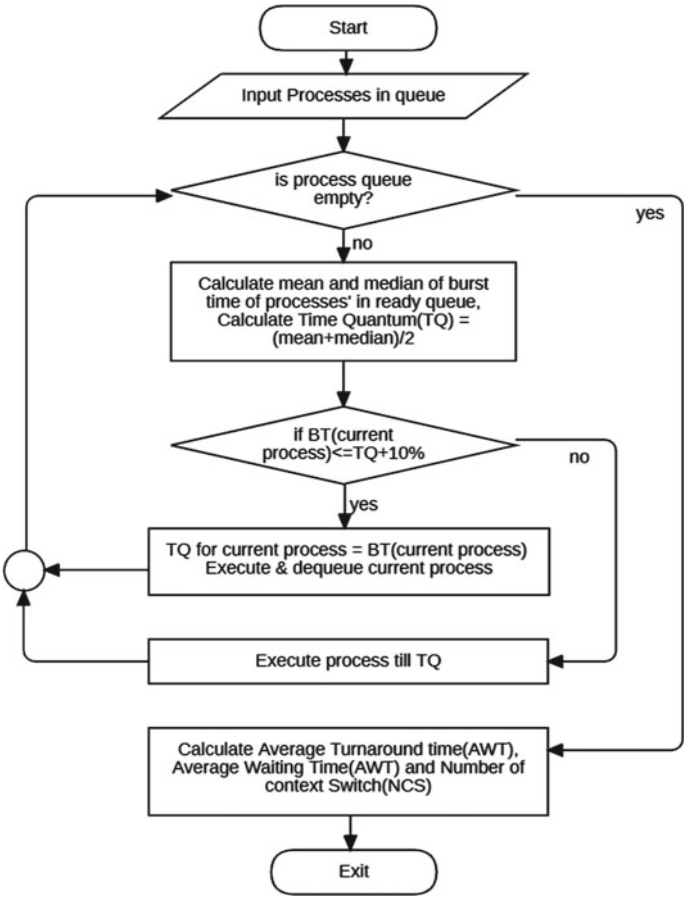


Fig. 2 Block diagram of proposed algorithm

Table 1 Algorithm

Step 1	Calculate mean and median
Step 2	Calculate TQ(TQ = (mean + median)/2)
Step 3	While in process queue repeat for each process (P) If BT(P) < = TQ then: Execute(P) till BT(P) Else if BT(P) < = TQ + 10% then: Execute(P) till BT(P) Else: Execute(P) till (BT(P)-TQ) If BT(P) is 0 then: Dequeue(P)
Step 4	Check if processes are in queue If YES then go to step 1 Else continue
Step 5	Calculate TAT for each P, WT for each process, AWT & ATT
Step 6	Exit

The model concentrates on operations that arrive to the ready queue at the same time. Initially, mean and median of BT of all processes are taken into consideration. In the first cycle, the average of both (i.e., mean and median) is used as the TQ. It is verified for each operation whether the remaining BT is 10% less than time quantum, equal to TQ, or less than or equal to TQ. If it is so, then the complete process is executed and dequeued from the ready queue or if it is greater than the processes is executed till TQ and then again enqueued into the ready queue.

Similarly, in the next cycles, the remaining time of the processes in ready queue is considered for calculation of the mean and median. The process continues till all processes are executed completely. The algorithm for same is discussed (Table 1):

4 Results and Discussion

For the analysis of the proposed algorithm, a hypothetical example is taken. Processes with same arrival time are taken into consideration for the work (Table 2).

For the evaluation of the proposed model, we have considered the following models.

Table 2 Hypothetical example

Process ID	Burst time
P1	31
P2	63
P3	40
P4	79
P5	75

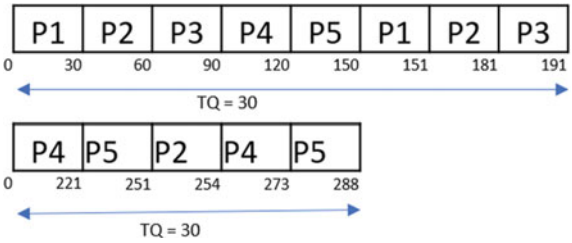


Fig. 3 Gantt chart—classic RR TQ = 30

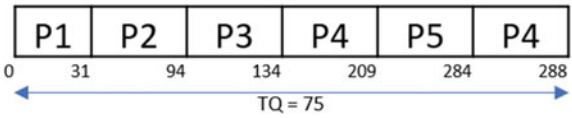


Fig. 4 Gantt chart—classic RR TQ = 75

4.1 Classic Round Robin with Time Quantum = 30

In classic RR algorithm, the TQ is fixed for each cycle. The lesser the TQ, the greater the number of Context Switch. The Gantt chart, for example, is as follow (Fig. 3):

For the example, the AWT is 173.8 and ATT is 231.4.

4.2 Classic Round Robin with Time Quantum = 75

Now, considering with maximum TQ, the algorithm acts like FCFS. The Gantt chart obtained is as follows (Fig. 4):

For TQ = 75, the AWT is 108.6 and ATT is 166.2.

4.3 Dynamic Round Robin with Controlled Preemption (DRRCP)

The DRRCP algorithm considers the median of BTs of the processes as TQ. The TQ is static for all cycles (Fig. 5).

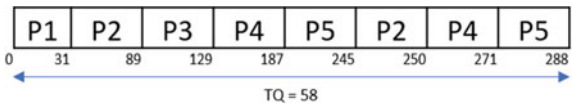


Fig. 5 Gantt chart—DRRCP

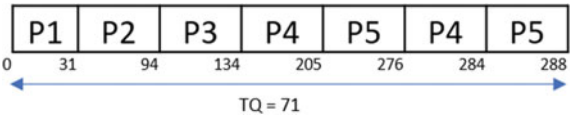


Fig. 6 Gantt chart—MMRRA

For this approach, the AWT obtained is 136.2 and ATT is 193.8. The TQ obtained was 58.

4.4 Modified Median Round Robin Algorithm (MMRRA)

The MMRRA algorithm considers the median and maximum burst time for calculation of TQ. The TQ is static for all cycles. The TQ is calculated as the square root of the product of median and maximum of all BT. The Gantt chart obtained for the example is as follows (Fig. 6):

For this approach, the AWT is 108.6 and ATT is 166.2. The Time Quantum for the algorithm obtained is 71.

4.5 Improved Round Robin Algorithm with Varying Time Quantum (IRRVQ)

The IRRVQ algorithm arranges the processes in ascending order of their Remaining BT. The smallest BT of them is considered as TQ for the current cycle. The algorithm leads to increase in context switches as it focuses on completion of the minimum process in the current cycle. The Gantt chart obtained for the algorithm is as follows (Fig. 7):

For this approach, the AWT is 140.4 and ATT is 198.0. The algorithm works on dynamic time quantum for the cycles, so the TQ are 31, 9, 23, 12, and 4 for each cycle, respectively.

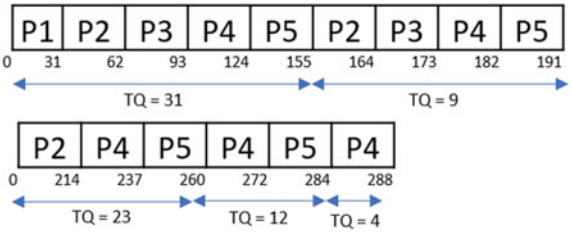


Fig. 7 Gantt chart—IRRVQ

4.6 *Proposed Approach*

The proposed approach selects TQ as average of both mean and median. The time quantum changes per cycle as well as within the cycle if required as discussed above. The Gantt chart obtained for the proposed approach is as follows (Fig. 8):

The approach provides an efficient approach with reduced AWT of 106.4 and ATT of 164.0. The time Quantum for cycle 1 were 60, 63 and for cycle 2 were 17 and 19.

The hypothetical example was evaluated on 5 different models with the proposed approach. The comparison of the algorithms is given as follows (Table 3):

On comparison, it is observed that the classic round robin algorithm with less TQ has the highest waiting time whereas the proposed algorithm has lowest. The comparison graph is shown below (Fig. 9):

Similarly, the classic round robin with less TQ has greater ATT and the proposed algorithm has lowest one. The comparison graph is as follows (Fig. 10):

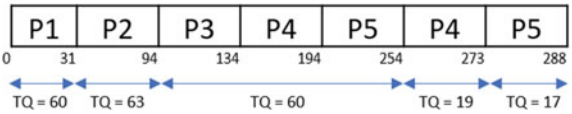


Fig. 8 Gantt chart—proposed algorithm

Table 3 Comparison of algorithms

Algorithm	TQ	AWT	ATT	NCS
Classic round robin	30	173.8	231.4	12
Classic round robin	75	108.6	166.2	5
MMRRA	71	108.6	166.2	6
DRRCP	58	136.2	193.8	7
IRRVQ	C1–31, C2–9, C3–23, C4–12, C5–4	140.4	198.0	14
Proposed approach	C1–60, 63, C2–17	106.4	164.0	6

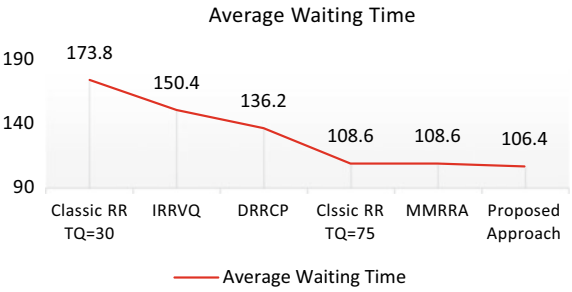


Fig. 9 Comparison of average waiting time (AWT)

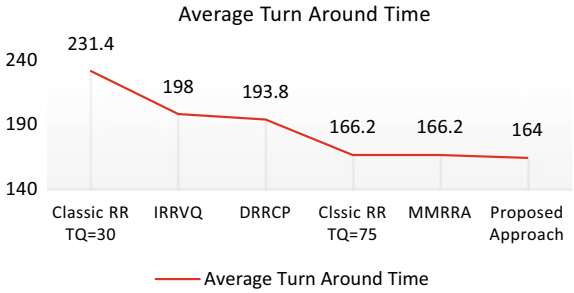


Fig. 10 Comparison of average turnaround time (ATT)

The Classic RR algorithm with higher TQ has lowest NCS but acts like FCFS algorithm, followed by the MMRRA and proposed algorithm. Whereas, the IRRVQ has highest NCS. The comparison is as follows (Figs. 11 and 12):

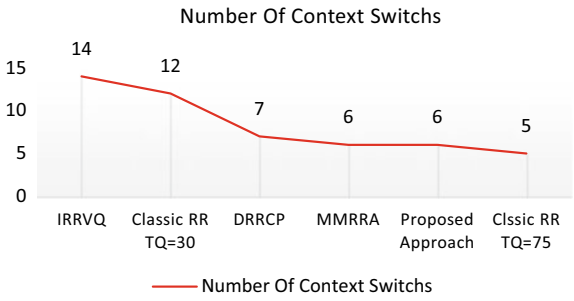


Fig. 11 Comparison of number of context switch (NCS)

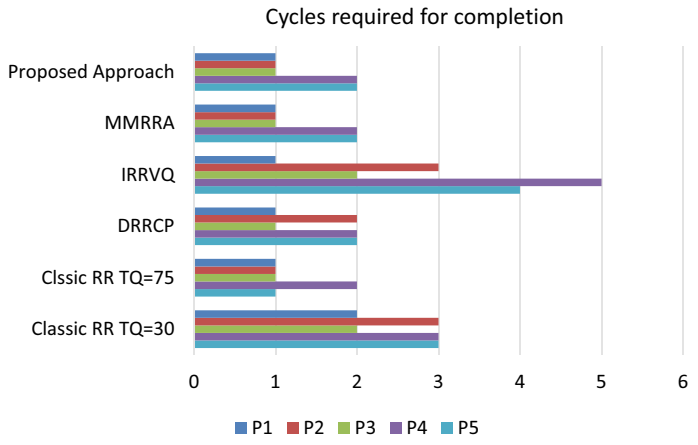


Fig. 12 Comparison of cycle required for completion

5 Conclusion and Future Scope

In the original RR algorithm, it is an improvement. The improvised algorithm will provide the necessary x + y units of time for the process's full execution if the cycle's quantum of time is x units and there is a process in the ready queue with an execution time of x + y units, where y is a reasonably tiny percentage of time. It has been determined that a process may operate within a specified range even if it exceeds the system's time quantum. The range, however, does not extend very far beyond the real-time quantum. This improves the overall performance of the system and its speed.

The current algorithm that was specifically developed is based entirely on the RR method. However, the algorithm can essentially be changed and combined with other algorithms to reduce the AWT, ATT, and NCS without altering the program's complexity, which is unquestionably quite considerable. Additionally, the algorithm's existing efficacy can be maintained even when it is specifically applied to processes with varying arrival times and priority.

References

1. Peter SA, Galvin B, Gagne G (2003) Operating system concepts, 6th edn. Wiley, Hoboken NJ
2. Bisht A, Ahad M, Sharma S (2014) Calculating dynamic time quantum for round robin process scheduling algorithm. Int J Comput Appl 98:20–27. https://doi.org/10.5120/17307-7760
3. Sohrawordi, Ehasn Ali, Palash Uddin and Mahabub Hossain. (2019); A modified round robin CPU scheduling algorithm with dynamic time quantum. Int J Adv Res 7(Feb):422–429
4. Alsulami AA, Al-Haija QA, Thanoon MI, Mao Q (2019) Performance evaluation of dynamic round robin algorithms for CPU scheduling. SoutheastCon 2019:1–5. https://doi.org/10.1109/SoutheastCon42311.2019.9020439
5. Alaa F, Zoulikha MM, Hayat B (2020) Improved round robin scheduling algorithm with varying time quantum. In: 2020 second international conference on embedded & distributed systems (EDiS), pp 33–37. https://doi.org/10.1109/EDiS49545.2020.9296452
6. Farooq MU, Shakoor A, Siddique AB (2017) An efficient dynamic round robin algorithm for CPU scheduling. In: 2017 international conference on communication, computing and digital systems (C-CODE), pp 244–248. https://doi.org/10.1109/C-CODE.2017.7918936
7. Tang Y, Guan N, Feng Z, Jiang X, Yi W (2021)Response time analysis of lazy round robin. In: 2021 design, automation & test in Europe conference & exhibition (DATE), pp 258–263. https://doi.org/10.23919/DATE51398.2021.9474242
8. Simarmata ER, Lumbantoruan G, Nainggolan R, Napitupulu J (2019) Round robin algorithm with average quantum dynamic time based on multicore processor. J Phys: Conf Ser. https://doi.org/10.1088/1742-6596/1361/1/012005
9. Balharith T, Alhaidari F (2019) Round robin scheduling algorithm in CPU and cloud computing: a review. In: 2019 2nd international conference on computer applications & information security (ICCAIS), pp 1–7. https://doi.org/10.1109/CAIS.2019.8769534
10. Zongyu X, Xingxuan W (2015) A predictive modified round robin scheduling algorithm for web server clusters. In: 2015 34th Chinese control conference (CCC), pp 5804–5808. https://doi.org/10.1109/ChiCC.2015.7260546
11. Mora H, Abdullahi SE, Junaidu SB (2017) Modified median round robin algorithm (MMRRA). In: 2017 13th international conference on electronics, computer and computation (ICECCO), pp 1–7. https://doi.org/10.1109/ICECCO.2017.8333325

E-Learning Paradigm in Cloud Computing and Pertinent Challenges in Models Used for Cloud Deployment

Dhaval Patel and Sanjay Chaudhary

Abstract The pedagogical science has improved immensely in each section of learning and education due to adoption of recent developments in Information and Communication Technologies (ICT) domain. Adopting web-based tools has aided to new thrust in imparting education through remote means. The deployment of cloud network is quickly aided to all disciplines of learning and at all levels. A user can access, from any remote location, a data stored over cloud platform using their computers, tablets or smartphones, and can also make perturbations in data on need basis. The method of E-learning becomes seamless when cloud-based techniques are employed. This also helps in cost cutting and cumbersomeness of data processing as the major chunk of services is operated by third party infrastructures. Both cloud-based and conventional E-learning when combined offers a greater advantage to teachers, students and institutes albeit with a clutch that the security becomes a major issue of concern. The present review is an analysis of perks of deploying cloud-based learning and the security concerns that come with it, as observed through various studies.

Keywords ICT · E-learning · Cloud computing · SaaS · PaaS · IaaS · MOOCS · NIST

D. Patel (✉)
Madhav University, Pindwara, Rajasthan 307026, India
e-mail: erdhavalpatelce@gmail.com

S. Chaudhary
Janardan Rai nagar Rajasthan Vidyapeeth University, Udaipur, Rajasthan, India
e-mail: schaudhary00@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_31

1 Introduction

Education as a vital tool for growth of any society needs to get evolved basis requirement of time and resources. The E-learning platform has emerged as convincingly cheap but effective means of holistic pedagog. All E-learning needs to have inculcation of information and communication technology (ICT) tools. Under the umbrella of ICT are included the learning skills given through internet-based tools, web application, smart devices, interactive methods, immersive learning tools and every other type of virtual means to fire imagination of learners [1, 2]. Cloud computing has added a new facet to ICT tools because it is very relatable to the whole idea of E-learning using the available resources and in a more effective manner. It has been suggested that the development in cloud technologies will hugely affect the overall knowledge and skill imparting in the pedagogical science [3]. The deployment of evolving, dynamic, interactive and virtual tool like Internet of Things (IoT) for efficient use of available resources of World Wide Web are hallmarks of cloud computing paradigm [4–6]. Other major benefit of adopting cloud platform is the manner it can be molded as per temporal and spatial needs of academic fraternity, institutions of learning and students. However, users are required to save their valuable data and need an access to them whenever demand arises, that scenario can lead to possible breach over internet [7]. A global survey conducted by Ecosystem AI in 2019 revealed that the most important objective of adopting the emerging methods of teachings is in the area of personalized learning followed by student profiling and then assessing the needs of students [8] (Fig. 1).

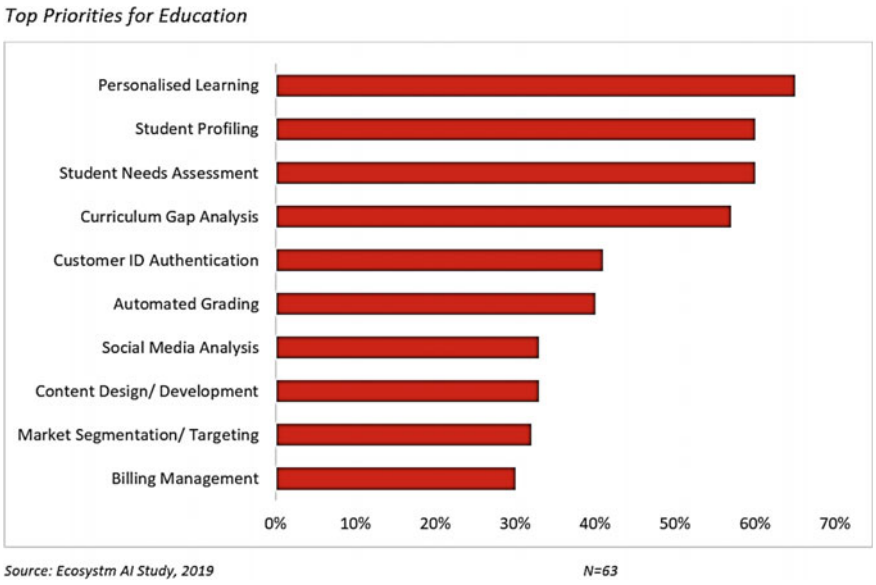


Fig. 1 An ecosystem AI study for education institutes [8]

2 Basics of E-Learning Domain

Electronic (E) or virtual learning was a connecting link that catered to both conventional and ICT-based learning [9]. The developments in IT services catering to ICT domain have caused paradigm shift in education and learning. So much so that the entire field of education is seeing a fundamental reshaping due to deployment of internet and allied services [10]. Both web-based technologies and ICT are getting utilized by institutes of higher learning, vocational activities and school for children [11]. The aggressive utilization of ICT technologies aided by cloud network is made by institutes of higher studies and for school going children [12]. Seeing the aggressive adoption of E-learning by higher learning institutes, schools, freelancers, professionals and even layman during the recent COVID-19 pandemic, it is suggested the future is very bright for E-learning tools, especially when cloud-based technologies are inculcated in them [13–16]. The cloud services can come to rescue in such scenarios of higher power servers, costly infrastructure and expert technicians with much less legal and financial burden [17, 18]. Ivy-league institutes like MIT and Harvard University have developed Massive Open Online Courses i.e. MOOCs forum for providing top quality educational materials to any student of any stream.

3 Cloud Paradigm

The cloud computing as defined by National Institute of Standards and Technology (NIST) is "a platform that givens an omnipresent, convenient, access to an internet based network on need basis, a network that has resources shared among them, such a network can be easily allocated without much supervision and interference from service providers [19]." Such a cloud platform has following facets (Fig. 2)

1. Five essential characteristics—Broader network, pooling of resources, flexibility, on-demand rendering of resources and a calibrated service.
2. Three models of delivery—Cloud computing providers offer their services according to three fundamental delivery models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [20].
3. Four modes of deployment—The private, public, hybrid and community modes are the four deployment models in cloud computing.

4 Three Major Models of Delivery Deployed on Cloud Services

The way in which cloud service serves their clients falls into three major model [21].

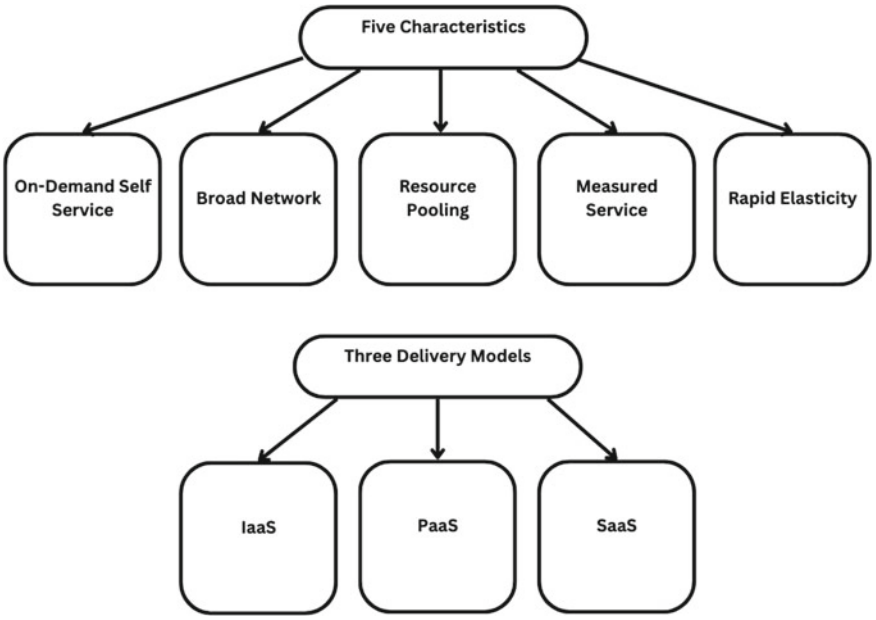


Fig. 2 Characteristics, delivery and deployment models in cloud-based services

4.1 Software as a Service (SaaS)

It is basically a web platform over the internet, which acts as distributor of software model. Since the user is permitted to access the cloud services based on their subscription status, it is also known as an "On-demand" service model. Being flexible in terms of its utilization, the SaaS is gaining traction among all major institutes and organizations [22]. SaaS is accessed over the internet or web browser and it is hosted, managed and secured by the service providers.

4.2 Platform as a Service (PaaS)

Basically, this platform is for creating, testing, running and controlling the application by expert programmers. In PaaS, the web is a basic tool to work for development of software both for expert users and novices [23]. The PaaS and SaaS are majorly similar with a difference that deployment is quicker and simpler for software distribution system in SaaS even while not buying the services.

4.3 Infrastructure as a Service (IaaS)

It is also known as Hardware as a Service. The IaaS skeleton is modulated by internet only, hence the cost of expenditure on actual servers is saved. Without being having a physical server, IaaS can provide frictionless personal network to do development over the network. But IaaS can only be used by network architectures (Fig. 3).

As shown in Fig. 4, the education systems of current era that use cloud platform as partner have developers, researchers, administrative staff, library, students, faculties and administration staff as major stakeholders.

5 Importance of Cloud Platform in E-Learning

The cloud platform and its allied services have importance in terms of scalability, affordability, convenience, inclusive learning, immersive learning, for education institutions [25, 26]. The benefits of cloud-based E-learning are categorized as follows

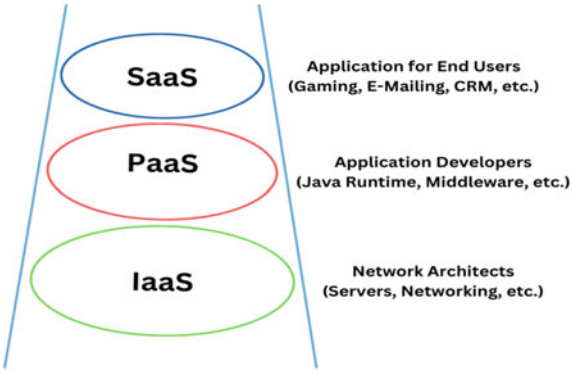


Fig. 3 Cloud service models [24]

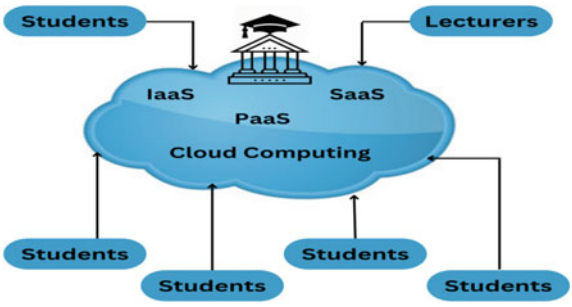


Fig. 4 Major stakeholders in cloud platform for E-learning

1. Benefits to Academic institutions—The innovation in science of teaching, pedagogics, has helped academic bodies to spread the knowledge through cloud computing in more efficient manner without incurring a consistent cost. The scalability, without compromising quality, has helped academic bodies globally to adopt cloud services in their E-learning domain [27].
2. Benefits to learners—The students, whether, kids, young people, working professional or researchers are the final beneficiaries of cloud-based learning. They can record the lectures, prepare notes, work on sophisticated platform, raise queries and write the exams using cloud platform in E-learning [28]. It has become increasingly uncommon for students from all around the globe to be part of a team that works on a collaborative assignment and performs exceedingly well [29, 30].
3. Benefits to instructors—Using cloud methods, the faculties and instructors can easily study, prepare and teach materials to a student group of heterogeneous nature. Even the process of publishing the intellectual property becomes more transparent for faculties when they adopt cloud platform for E-learning [31].

6 Drawbacks and Hurdles in Cloud-Based E-Learning Methods

Notwithstanding the huge benefits when cloud computing is incorporated in teaching methods, the drawbacks still lingers large [32, 33]. Apart from initial investment and adoption by different departments of an educational institution, the cloud models suffer from serious issues of security at all the levels, especially in the institutions where intellectual property theft is a major issue of concern [34–37]. The present review work is an attempt to understand the major security issues when cloud computing is incorporated in higher education institutes.

6.1 Issues with Software as a Service (SaaS)

Location of information and its reliability is a major issue in SaaS model of cloud computing. The end user has always a suspicion that their geographical location is compromised or not as they can't be sure whether Service Level Agreements are followed by cloud service providers. Services can be hired from trusted vendors e.g. CISCO, ACS, Microsoft etc.

6.2 Issues with Platform as a Service (PaaS)

PaaS has security issues too in its programming structure and scaled-up services to chosen users are also a tedious task. The security models worked out for one PaaS platform may not be so effective on other, howsoever better programming was done.

6.3 Issues with Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) provides better security for cloud service users comparing SaaS and Paas. Security of both physical and virtual environment is catered upon in IaaS, since they consider it as a primary responsibility. Attacks of cross-based type are more common when infrastructures are shared among users in IaaS model that is deploying virtualization methods.

7 Risks Associated with Security in Cloud-Based E-Learning

Academic organizations are eager to adopt cloud services which will cater to objective of education-for-all [38]. But the security issues in terms of data theft are a major issue of concern. The more robust security measures require initial investment which may not be affordable for a modest institute [39]. The seamless adoption with scalability can never be realized if security issues are persistent and costly to do away with [40, 41].

Cloud Platform: The virtual infrastructure needed to be deployed as a major skeleton of cloud services and physical instruments of application are two vital parameters needed to implement cloud platform for E-learning. However, this very process of virtualization can lead to security breach as all the virtual operations are performed on individual computers and, hence, chances of getting hacked are higher. The commonly found attacks on cloud platform are Denial of Service (DoS), incorporation of malware, attacks using botnets and theft of services [20, 42]. The standards for technical consideration of E-learning platform including IEEE Learning Object Metadata (IEEELOM), Shareable Content Object Reference Model (SCORM) or Instructional Media Services (IMS) have aggressively deployed Cloud platform in their execution. The *Dublin Core* Metadata Initiative (DCMI), which was developed in 1994, is a non-profit project that describes the elements of resources. Hence, it can also be applied for cloud-based learning. The current IEEELOM is result of collaboration between IMS and an ARIADNE project of Europe.

8 Conclusion

A forecast made in 2017 through a survey by International Data Corporation (IDC) suggested that, by 2022, the cloud industry would become about $400 billion industry much due to adoption of cloud in educational institutes and due to innovations in digital technologies [43]. The top four countries to have adopted cloud-based learning are, USA, India, China and South Korea. A Sloan consortium study suggests about 6 million enrollments for online education in USA itself. While for developing a country like India, the online education boom got more impetus after COVID-19 pandemic and even hinterlands of the country, with internet accessibility, joined the bandwagon of online education revolution. China is estimated to have over 70 colleges and institutes imparting education based on cloud platform [39]. The lingering issues of safety are a major concern along with initial cost of deployment [44]. The present work paid emphasis on advantages of E-learning when aided by cloud platform along with the major security issues that can be confronted [45]. The cloud computing is in itself a rapidly evolving field and its application is improving for various fields [46]. Hence, the issues of security in academic learning through E-learning can be looked upon from newer angles and emerging threats can be catered by more robust and efficient processes.

References

1. Liang Y, Wu S (2021) Applying the cloud intelligent classroom to the music curriculum design of the mental health education. Front Psychol 12:729213. Published online 16 Nov 2021. https://doi.org/10.3389/fpsyg.2021.729213
2. Sharma R (2017) Internet of things: an approach for advancement in educational institution. In: 2016 1st India international conference on information processing (IICIP). Date of Conference: 12–14 August 2016. Date Added to IEEE Xplore: 13 July 2017
3. Chen L, Ifenthaler D, Yau JY-K (2021) Online and blended entrepreneurship education: a systematic review of applied educational technologies Entrep Educ 4(2):191–232. Published online 2021 April 16. https://doi.org/10.1007/s41959-021-00047-7
4. Majeed A, Ali M (2018) How Internet-of-Things (IoT) making the university campuses smart? QA higher education (QAHE) perspective. In: 2018 IEEE 8th annual computing and communication workshop and conference (CCWC). IEEE Xplore: 26 Feb 2018
5. Nabi SA, Gurram D (2015) Mobile hybrid cloud computing for educational institutions: Mobihybrid EduCloud. In: IEEE sponsored 2nd international conference on innovations in information, embedded and communication systems (ICIIECS)
6. Thavi R, Jhaveri R, Narwane V, Gardas B, Jafari Navimipour N (2021) Role of cloud computing technology in the education sector. J Eng Des Technol. https://doi.org/10.1108/JEDT-08-2021-0417
7. Slater EV, Barwood D, Cordery Z (2022) Pre-service teachers' use of ICT to collaborate to complete assessment tasks. Austr Educ Res, 1–20. Advance online publication. https://doi.org/10.1007/s13384-022-00580-x
8. https://blog.ecosystm360.com/cloud-transforming-education-industry/
9. Asad MM, Naz A, Shaikh A, Alrizq M, Akram M, Alghamdi A (2022) Investigating the impact of IoT-Based smart laboratories on students' academic performance in higher education.

Universal access in the information society, 1–15. Advance online publication. https://doi.org/10.1007/s10209-022-00944-1
10. Upadhyaya J, Ahuja NJ (2017) Quality of service in cloud computing in higher education: a critical survey and innovative model. In: 2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC). Date of Conference: 10–11 February 2017, Date Added to IEEE Xplore: 5 October 2017
11. Almajalid R (2017) A survey on the adoption of cloud computing in education sector. ArXiv. Vol.abs/1706.01136
12. Suleiman MM, Kaur T, Sharma A, Tewari AK, Kuliya M, Aliyu AS (2021) Review of application of cloud computing in education. J Appl Sci Inf Comput 1(2)
13. Noh NHM, Amron MT (2021) Exploring cloud computing readiness and acceptance in higher education institution: a PLS-SEM approach. Asian J Univ Educ (AJUE) 17(4)
14. Alimboyong CR, Bucjan ME (2021) Cloud computing adoption among state universities and colleges in the Philippines: issues and challenges. Int J Eval Res Educ (IJERE) 10(4):1455–1461
15. IDC forecasts worldwide public cloud services spending. https://www.idc.com/getdoc.jsp?containerId=prUS44891519. 28 Feb 2019
16. Alashhab ZR, Anbar M, Singh MM, Leau YB, Al-Sai ZA, Alhayja'a SA (2020) Impact of coronavirus pandemic crisis on technologies and cloud computing applications. J Electron Sci Technol 19, Article 100059
17. Li W, Guo Y (2022) A secure private cloud storage platform for english education resources based on IoT technology. Comput Math Methods Med 2022:8453470. Published online 2022 June 14. https://doi.org/10.1155/2022/8453470
18. Zhen C, Hu K (2022) Design of edge computing online classroom based on college English teaching. Comput Intell Neurosci 2022:7068923. https://doi.org/10.1155/2022/7068923
19. Niharika K, Lavanya G (2012) Educational cloud: utilization of IaaS versus PaaS services. Int J Sci Eng Res 3(10). ISSN 2229-5518
20. Iftikhar W, Mahmood Z, Arun KC (2018) A review on cloud computing issues and benefits in higher education institutes. Int J Creative Res Thoughts (IJCRT) 6(2)
21. Bruggeman B, Garone A, Struyven K, Pynoo B, Tondeur J (2022) Exploring university teachers' online education during COVID-19: tensions between enthusiasm and stress. Comput Educ Open 3:100095. ISSN 2666-5573
22. Wu CH, Tang YM, Tsang YP, Chau KY (2021) Immersive learning design for technology education: a soft systems methodology. Front Psychol 12:745295. Published online 17 Dec 2021. https://doi.org/10.3389/fpsyg.2021.745295
23. Peramunugamage A, Ratnayake UW, Karunanayaka SP (2022) Systematic review on mobile collaborative learning for engineering education. J Comput Educ, 1–24. https://doi.org/10.1007/s40692-022-00223-1
24. Shahzad A, Golamdin AG, Ismail NA (2016) Opportunity and challenges using the cloud computing in the case of Malaysian higher education institutions. Int J Manage Sci Inf Technol (IJMSIT) 20:1–18. ISSN 1923-0273
25. Chamunorwa T, Modran HA, Ursuțiu D, Samoilă C, Hedeșiu H (2021) Reconfigurable Wireless sensor node remote laboratory platform with cloud connectivity. Sensors (Basel) 21(19):6405. https://doi.org/10.3390/s21196405
26. Xu X, Xie J, Wang H, Lin M (2022) Online education satisfaction assessment based on cloud model and fuzzy TOPSIS. Appl Intell (Dordr) 52(12):13659–13674. Published online 5 March 2022. https://doi.org/10.1007/s10489-022-032
27. Daher W, Salameh H (2022) The role of a ministry of education in addressing distance education during emergency education. Eur J Investig Health Psychol Educ 12(5):478–493. Published online 20 May 2022
28. Olaloye FJ, Adeyemo AD, Edikan E, Lawal CO (2019) Cloud computing in education sector: an extensive review. Int J Civil Eng Technol (IJCIET) 10(03):3158–3171
29. Hamutoglu NB (2020) Acceptance and use of cloud computing systems in higher education: an application of TAM 3 within the socio-cultural context of educational Institutions. Malay Online J Educ Technol 8(4)

30. Ding Y, Li Y, Cheng L (2020) Application of internet of things and virtual reality technology in college physical education. IEEE Access 8:96065–96074. Date of Publication: 11 May 2020
31. Zhang L, Yu X (2021) Intelligent retrieval method of mobile learning resources in the intelligent higher education system. Int J Syst Assur Eng Manag 17:1–13. https://doi.org/10.1007/s13198-021-01455-7
32. Baharuddin, Ampera D, Fibriasari H, Sembiring MAR, Hamid A (2021) Implementation of cloud computing system in learning system development in engineering education study program. Int J Educ Math Sci Technol (IJEMST) 9(4):728–740. https://doi.org/10.46328/ijemst.2114
33. Tariq MI, Tayyaba S, Rasheed H, Ashraf MW (2017) Factors influencing the cloud computing adoption in higher education institutions of Punjab, Pakistan. In: 2017 international conference on communication, computing and digital systems (C-CODE). Date of Conference: 8–9 March 2017. Date Added to IEEE Xplore: 04 May 2017
34. Madni SHH, Ali J, Husnain HA, Masum MH, Mustafa S, Shuja J, Maray M, Hosseini S (2022) Factors influencing the adoption of IoT for E-learning in higher educational institutes in developing countries. Front Psychol 13:915596. Published online 8 July 2022. https://doi.org/10.3389/fpsyg.2022.915596
35. Onyema EM, Eucheria NC, Nneka UA, Afriyie RK, Nwoye OU (2020) Cloud security challenges: implication on education. IJCSMC 9(2):56–73
36. Lu Y, Xing M, Song J (2022) Design and implementation of balance ability assessment training system for special children based on education cloud integration. J Environ Public Health 2022:9359367. Published online 26 Sept 2022. https://doi.org/10.1155/2022/9359367
37. Villegas-Ch W, García-Ortiz J, Román-Cañizares M, Sánchez-Viteri S (2021) Proposal of a remote education model with the integration of an ICT architecture to improve learning management. PeerJ Comput Sci 7:e781. Published online 3 Dec 2021. https://doi.org/10.7717/peerj-cs.781
38. Sophonhiranrak S (2021) Features, barriers, and influencing factors of mobile learning in higher education: a systematic review. Heliyon 7(4):e06696. Published online 8 April 2021. https://doi.org/10.1016/j.heliyon.2021.e06696
39. Ngampornchai A, Adams J (2016) Students' acceptance and readiness for E-learning in Northeastern Thailand. Int J Educ Technol High Educ 13:34. https://doi.org/10.1186/s41239-016-0034-x
40. Juma MK, Tjahyanto A (2019) Challenges of cloud computing adoption model for higher education level in Zanzibar (the Case Study of SUZA and ZU). Procedia Comput Sci 161(2019):1046–1054
41. Liu C, Song B (2021) Impact assessment of big data on higher education management based on time-varying clustering sampling algorithm. Comput Intell Neurosci 2021:7954797. Published online 14 Dec 2021. https://doi.org/10.1155/2021/7954797
42. Wang H, Chen M (2022) Application of the flipped classroom mode under few-shot learning in the teaching of health physical education in colleges and universities. Comput Intell Neurosci 2022:1465613
43. Aydin H (2021) A study of cloud computing adoption in universities as a guideline to cloud migration. SAGE Open 2021:1–14
44. Agrawal S (2021) A survey on recent applications of cloud computing in education: COVID-19 perspective. J Phys: Conf Ser 1828(2021):012076
45. Qasem YAM, Abdullah R, Jusoh YY, Atan R, Asadi S (2019) Cloud computing adoption in higher education institutions: a systematic review. IEEE Access 7:63722–63744. Date of Publication: 10 May 2019
46. Odeh M, Garcia-Perez A, Warwick K (2017) Cloud computing adoption at higher education institutions in developing countries: a qualitative investigation of main enablers and barriers. Int J Inf Educ Technol 7(12):921–927

Parkinson's Disease Detection: Comparative Study Using Different Machine Learning Algorithms

Vijaykumar Bhanuse, Ankita Chirame, and Isha Beri

Abstract Detection of Parkinson's disease is often established on clinical monitoring and evaluation of medical signs, which include an indication of different motor signs. Furthermore, the ordinary point of view encounters rationality as it is based on analysis of motions that are specifically not recognized, tough to categorize, and steers miscategorization. Since Parkinson's disease sufferers have distinct voice characteristics, vocal recordings are needed and highly regarded as a diagnostic tool. If Machine Learning techniques can truly note this disease with the help of a voice recording dataset, this would be a useful screening phase before seeing a doctor. This chapter shows comparison of some ML models like XGboost, random forest, Naive Bayes, logistic regression, SVM, decision tree, and K-NN to get the best result for detection of disease.

Keywords Healthcare · Machine Learning (ML) · Parkinson's disease · XGBoost · Support Vector Machine

1 Introduction

Parkinson's disease (PD) is a central nervous system related and neurodegenerative disease. There is an estimation that it affects somewhat 1% of the population worldwide. This mostly contains an age group of 60 years and above. The majority symptoms of PD are tremor, rigidity of limbs, slowness in movements. Also, the most dangerous and common one is the drastically dropping dopamine levels in a

V. Bhanuse (✉) · A. Chirame · I. Beri
Instrumentation and Control Engineering, Vishwakarma Institute of Technology, Pune, India
e-mail: vijaykumar.bhanuse@vit.edu

A. Chirame
e-mail: ankita.chirame18@vit.edu

I. Beri
e-mail: isha.beri18@vit.edu

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_32

person's brain which is the reason for the above symptoms. These are motor signs. Other notable changes are, reduced ability to smell, forgetfulness or confusing one memory with another, depression and lack of sleep, which are non-motor signs.

The analysis of PD is usually conducted on motor signs. Regardless of the institution of cardinal symptoms of PD in medical assessments, many of the ranking scales used in the estimation of disorder severity have no longer been completely evaluated and validated. Although non-motor signs and symptoms are existing in many people prior to the onset of PD, they lack specificity, are problematic to investigate and/or yield variability from affected person to patient. Therefore, non-motor signs do not enable the prognosis of PD independently.

Nowadays, in healthcare domains, Machine Learning is getting recognition. Basically, this computer program is programmed in such a way that it learns and extracts the needed information or data present in the dataset in a semi-automatic aspect.

In this project, to get the accurate precision of Parkinson's disease present in a patient, various machine learning models are generated. These models include Support Vector Machine, Decision Tree, Random Forest, Naive Bayes, XGBoost, Logistic Regression and K-NN. These models were implemented on a dataset, which is a collection of voice recordings documentation of PD-diagnosed patients and healthy people. These models were trained and tested using the given dataset to get the highest precision rate among mentioned models.

2 Literature Review

These years, scope of research in the subject of ML models to the identification of Parkinson's disease has enhanced. Some prior papers have assessed work of ML techniques in detection of Parkinson's disease; they were restricted to the survey of wearable sensor data, motor signs and kinematics.

(1) Ref paper [1] in this paper the author explored the use of multi-modal machine learning techniques to predict the likelihood of parkinson disease. They presented novel framework that leveraged GenoML to enhance predictions by integrating multi-omic data types.
(2) Ref paper [2] states that Dataset obtained from voice recordings of both diagnosed patients and healthy persons has been used to perform different machine learning models, such as, Convolutional Neural Network, Hidden Markov Model and Artificial Neural Network. Other models like acoustics features derived from the speaker's sound unit such as RMS, MFCC, chrome STFT, Zero crossing rate, spectral centroid and bandwidth and Roll-off are also used to gain more information. Results are stated declaring ANN as accuracy of 96% compared to other results.

(3) Ref paper [3] mentions that work of classification methods like DMneural, Neural Networks, decision tree and regression were used to detect Parkinson's disease with the help of SAS software. It states Neural Networks as the best classifier by 92.9%.
(4) Ref paper [4] studies TQWT (tunable Q-factor wavelet transform) for feature extraction on Parkinson's disease patients' high-frequency resolution voice signals compared to other wavelet transforms. This is later compared with state of the art feature extraction methods, showing TQWT has better performance compared to the former.
(5) Ref paper [5] introduces two rules to speech analysis, which are fractal scaling and recurrence. They together reproduce the 'hoarseness' diagram. After that bootstrapped classifier distinguishes disordered from normal ones by quadratic discriminant analysis.

3 Implementation

3.1 Database

The dataset used in this project was taken from UCI ML Repository: Parkinson Dataset. The data is built by Oxford University in association with The National Center for Voice and Speech, Denver; which owns the recording of the speech signals used in the dataset. The above-mentioned dataset consists of 31 cases of Parkinson's disease patients. Of these 31 cases, 23 consists of the people who are suffering from Parkinson's disease and the remaining 8 do not have Parkinson's disease. There are, in total, 195 voice recordings documented in the dataset. These 195 recordings are, in fact, 6/7 recordings of 31 cases considered. The reason for this many recordings is to eliminate the false positives and be sure about the result. And the columns dwell various voice measures such as, average, minimum, maximum vocal fundamental frequency; variation in fundamental frequency; variation in amplitude; ratio of noise to tonal component; non-linear dynamical complexities; exponent for scaling signal fractal and non-linear measures of fundamental frequency variation. There is also a column that states the 'status' of the patient i.e., one and zero which represents non-Parkinson's case and Parkinson's case, respectively.

3.2 Workflow

See Fig. 1.

i. *Loading the Dataset and Exploring*

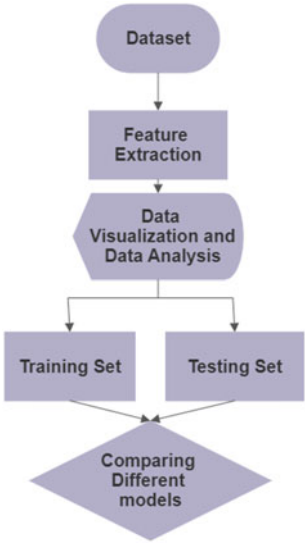


Fig. 1 Workflow of the project

In this step, the data is imported and explored. The numeric values 0 and 1 are also extracted from the dataset, where it is predicted if the patient has Parkinson's or not. Here, all the features of all the attributes with the values can be explored.

ii. *Data Processing*

In this phase of the project, the dataset is further split into two categories—training and testing. In furtherance of finding the most precise model for the given dataset, there is a necessity to train and test the given dataset with various machine learning algorithms.

iii. *Data Visualization*

Data is visualized with the assistance of the seaborn, which is a library in python used for data visualization. In this project, the created data frame is visualized via pair plot, counterplot, heatmap, box plot and bar plot.

iv. Model Building

Here, different classifiers are being trained and tested. The classifiers are logistic regression, k-NN, random forest, Naive Bayes, XGBoost, decision tree and support vector machine.

I. Support Vector Machine

It makes the best decision edge which can isolate n-dimension areas in groups, it simply puts a current data point in the right group later. It picks extreme vectors which create a hyperplane. They are support vectors, which make it a Support Vector Machine.

II. Random Forest

It is a supervised learning technique that is used in regression, classification problems. In this, decision trees are built on various samples, take major votes for average and classification in regression. It performs better outcomes for classification problems.

III. Decision Tree

It is a supervised learning algorithm, generally picked for resolving classification questions. This is a tree-shape classifier, i.e., internal nodes indicate elements of dataset, branches indicate decision rules, every leaf node says output. Decision nodes make decisions and contain many branches. Leaf nodes are output of those decisions.

IV. Naive Bayes

It is a supervised learning algorithm and is a probabilistic classifier. It creates a fast ML model for fast predictions. High dimensional datasets use this algorithm, like a text classifier.

V. XGBoost

Extreme Gradient Boosting is an open-source library. It is a scalable, distributed (GBDT) ML lib. It is the major MLlib for ranking problems and regression.

VI. Logistic Regression

It is a supervised learning technique which uses to see the probability of a chosen variable. In this, there are only two attainable categories.

VII. K-NN

It is a supervised learning algorithm. It figures the resemblance between the new data and present cases, places new samples in groups which are the same to available classes. It is used for classification and regression.

4 Results and Discussions

In this project, many models like XGboost, Naive Bayes, logistic regression and many more are used, with each having its different accuracy. The first used is logistic regression and it has a precision of 92%. Further, decision tree and K-NN model is used which acquires precisions of 89 and 91, respectively. Additionally, models like support vector machine also known as SVM, Naive Bayes are also tested and the precisions of 93 and 80 are obtained. Lastly, the XGboost and Random Forest model is applied and 95 and 96 precision is obtained, respectively (Table 1).

Table 1 Cumulative results of all models

S. No.	Model name	F1 score (in percentage)
1	Logistic Regression	92
2	Decision Tree	89
3	KNN	91
4	Support Vector Machine	93
5	Naive Bayes	80
6	XG Boost	95
7	Random Forest	96

5 Future Scope

As of now, this model estimates whether a person has Parkinson's disease or not; but, to increase the scope of the project may also include the stage of the disease, that is, either early or medium or the last stage.

6 Conclusion

Various machine learning algorithms were discussed and applied to the database for Parkinson's disease detection successfully. These models make diagnosis easier and less time-bound. As per results, Random Forest model is best with 96% precision.

References

1. Makarious MB, Leonard HL, Vitale D, Iwaki H, Sargent L, Multi-modality machine learning predicting Parkinson's disease
2. Radha N, Sachin Madhavan RM, Sameera Holy S, Parkinson's disease detection using machine learning techniques
3. Das R (2010) A comparison of multi-classification methods for diagnosis of Parkinson's disease. Expert Syst Appl 37:1568–1572
4. Sakar CO, Serbes G, Gunduz A, Tunc HC, Nizam H, Sakar BE, Tutuncu M, Aydin T, Isenkul ME, Apaydin H (2018) A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform
5. Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM (2007) Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. Biomed Eng

Implementation of Blockchain in Automotive Industry to Secure Connected Vehicle Data: Study and Analysis

Yedida Venkata Rama Subramanya Viswanadham and Kayalvizhi Jayavel

Abstract The automotive industry is undergoing rapid transformation because of the collaboration between technology companies and automakers. To support the present trend of connected cars, several stakeholders like insurance firms, car buyers and sellers and government agencies need different kinds of vehicle data. Data gathering is currently either manual or unverified, which raises trust, validity, and exactness concerns such as tampered safety checks and false or duplicate vehicle data records, among others. A robust instrument that is capable of safeguarding vehicle data is essential. It is necessary to record the changes for the purpose of auditing and ultimately establishing faith in the system. To meet these requirements, we propose utilizing blockchain technology. Healthcare, transportation, finance, cybersecurity, and supply chain management are just a few of the fields in which blockchain technology has recently gained popularity and been implemented. The determination of this survey is to deliver a comprehensive investigation of the advantages and uses of the Blockchain application in the automotive segment.

Keywords Automotive industry · Blockchain · Consensus algorithms · Autonomous vehicles · NFT's · Ethereum · Solidity · MOBI

1 Introduction

Data and privacy breaches are possible as the data-driven mobility ecosystem expands. A lot of data needs to be collected from cars and sensors and stored in a central repository for a self-driving car to work well. Cyberattacks are more likely

Y. V. R. S. Viswanadham (✉)
Department of Computer Science, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India
e-mail: yv8261@srmist.edu.in

K. Jayavel
Department of Information Technology, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_33

to target data that is stored in a centralized system. As more vehicles are introduced, it will also be challenging to manage and expand the system. Functions like Lock and unlock, temperature controls, and other features of many modern smart vehicles can be controlled remotely from smartphones or computers. As a result, hackers can gain access to these remotely operated features, making automobiles vulnerable [1].

The data gathered by vehicles can be misused in a variety of ways if it gets into the wrong hands. Information about a person's car use, including in-vehicle communications, driving directions, and infotainment services, could reveal individual information and lead to serious issues like identity theft. In addition, the data that is collected can be used for targeted advertising, which can be displayed on specific screens, on mobile devices in the vehicle, through the speakers in the vehicle, or by routing the vehicle to show passengers commercial activities that might not be of interest to them [2]. Automakers ought to guarantee a high level of safety because sensitive personal information like locations, addresses, and driving patterns are collected.

2 Importance of Data for Autonomous and Smart Vehicles

Data about driving patterns, locations, traffic, and other variables are generated by autonomous vehicles. They are increasingly connected to roadside infrastructure and other smart vehicles in proximity. The collected data can be analyzed by smart vehicles to predict the condition of the roads and identify traffic conditions to make decisions about the best routes, parking spots, and so on. In addition, smart vehicles could quickly create an instant city map, identify nearby accidents, and identify services like charging stations on the road by utilizing these data resulting in improved passenger experiences [3]. Passengers and drivers, automobile manufacturers, support service providers, and authorities in charge of city and road management will all be major stakeholders in the automobile data that is generated. Service providers can utilize the data to comprehend user behaviors and enhance service quality. In smart cities, the data also aids authorities in effectively managing traffic [4].

3 Adoption of Blockchain in Automotive Industry

Blockchain has the potential to completely alter the automotive industry. Blockchain technology is advantageous for every process in the automotive value chain, including purchasing, manufacturing, distribution, and service functions due to its distributed, immutable, and transparent nature. Smart automobiles are the transportation of the future [5]. Soon, we will see driverless/autonomous vehicles that use data on passenger behavior and traffic to make decisions and move around on the road. These intelligent vehicles can communicate with one another and other facilities on the road thanks to wireless networks. However, new challenges come with technological advancement. Cyberattacks can target the data collected by autonomous

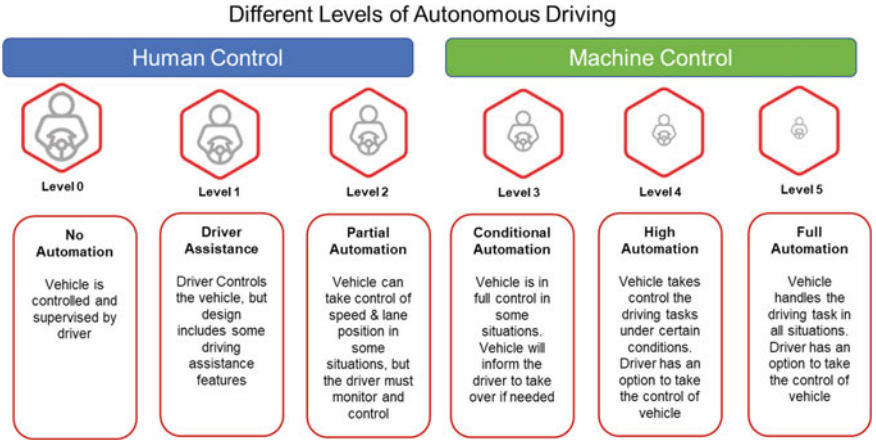


Fig. 1 Levels of autonomous driving

vehicles, and there are many ways to misuse it. As a result, automobile manufacturers are experimenting with and developing new solutions to ensure the privacy and security of their data using Blockchain technology [4].

Different Levels of Autonomy
See Fig. 1.

4 Overview of Blockchain Technology

Blockchain is a data structure that simultaneously ensures decentralization, transparency, and security by storing transactional records. A blockchain is a distributed ledger that can be accessed by everybody on the network. Information that has already been stored on a blockchain can't be changed easily. A blockchain protects each transaction with a digital signature that verifies its authentication. Due to the usage of encryption and digital signatures, the data that is inserted into the blockchain cannot be altered or tampered. Blockchain technology makes consensus, or agreement among all network participants, possible. A blockchain digitally stores all data and provides a common history that all network participants can access [6].

4.1 Features and Advantages of Blockchain

- **Reduction in Time**: Blockchain considers a rapid goal of exchanges with high unwavering quality and security finance. Verification, settlement, and clearance are quick and easy.

- **Unalterable Transactions**: This technology is used to verify the inalterability of all operations by sequentially registering transactions, indicating that all transactions are immutable. As a result, data integrity is maintained, and modification is limited.
- **Reliability**: By certifying and confirming the uniqueness of each chain party, duplicate records are eliminated, transactions are accelerated, and rates are reduced.
- **Security**: Cryptography in Blockchain employs the SHA-256 hashing algorithm to guarantee the security of its data. The technology is distributed laser. To ensure that the system continues to function despite the separation of any node, each party will keep a copy of the initial chain.
- **Collaboration**: Without the need for a third party to act as an intermediary, Blockchain platform enables each party to conduct business directly with the other.
- **Decentralized**: This is because the system is not controlled by a single authority. Every node uses specific paradigms and rules when interacting with Blockchain data.

4.2 Ethereum Blockchain Platform

Ethereum is an open source, distributed, and decentralized computing platform. Ethereum is a public or private blockchain with a built-in turning-complete language to develop smart contract. Additionally, it provides a platform for the creation and operation of smart contracts and decentralized applications, or DApps. Ether is a cryptocurrency in the Ethereum network [7–9].

4.3 Smart Contract

A smart contract is a business logic code that is stored on a blockchain-based platform and automatically executes all or part of an agreement. Many platforms and applications built with blockchain or distributed ledger technology rely heavily on smart contracts. Anyone can create and deploy a smart contract to a blockchain. Any interested party can see exactly a smart contract logic when it receives digital assets because the code is visible and verifiable. Although many other cryptocurrency blockchains, such as EOS, Neo, Tezos, Tron, etc. can run smart contracts, Ethereum is the most widely used platform [7, 8].

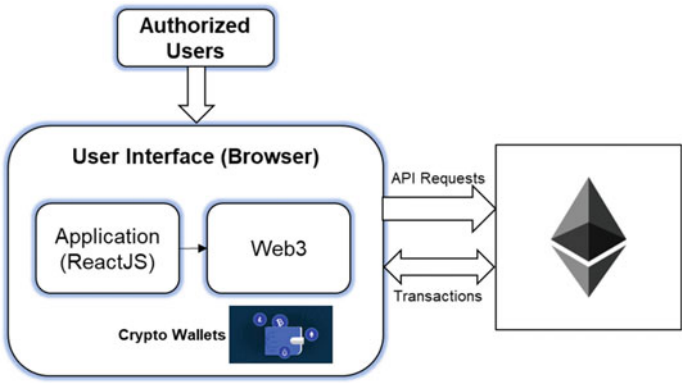


Fig. 2 Frontend flow diagram

4.4 Solidity

Solidity is an objected oriented, contract-oriented, and high-level language to develop and deploy smart contract. Solidity is statically types, support inheritance, libraries, and user-defined types among other features.

4.5 Ethereum Blockchain Architecture—Frontend

The technology used to create the front-end and Web Interface for the Ethereum application is React JS. Server oversees all data interactions between the User Interface and the backend database in traditional web services. The role of the server in Ethereum architecture is minimal. Web UI development is made simpler and easier by the extensive libraries offered by React JS [4].

To communicate with the Ethereum network, JavaScript code was used to build the application. It focuses on a variety of functionalities, beginning with the contract's status before calling backend functions from the contract ABI, like car registration, rating update, and data retrieval from Blockchain (Fig. 2).

4.6 Ethereum Blockchain Architecture—Backend

The creation and writing of smart contracts are essential components of the backend design. Smart contract can deploy to any Ethereum network, including the main network, test networks (Ropsten, Kovan, and Rinkeby), local-host 8245, and custom RPC, after writing the contract logic. The Ethereum Test Network choice is Rinkeby Testnet. On this network, test applications are hosted. Every network transaction,

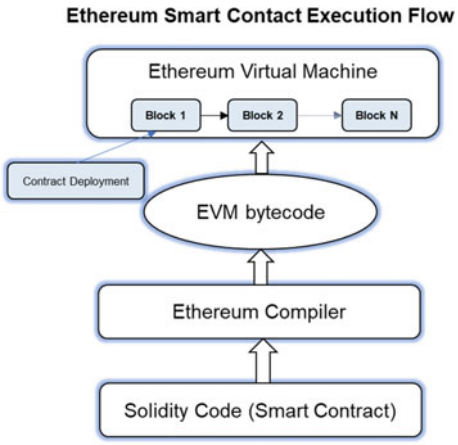


Fig. 3 Ethereum smart contract process

such as inserting records and extracting data, requires gas and ethers to be spent [4]. Backend process shows in Fig. 3.

4.7 Consensus Algorithms in Blockchain

A method for ensuring the authenticity of all blockchain transactions is consensus. It is frequently accomplished through the contributions of other chain network participants. There are typical kinds of consensus algorithms [9]:

Proof of Work
PoW is the consensus algorithm used by Bitcoin and some other familiar cryptocurrency networks. Bitcoin and a few other well-known cryptocurrency networks employ the PoW consensus mechanism. To gain permission to insert new transactions to the blockchain, participant nodes must answer a computationally intensive problem. Nodes can obtain Bitcoins by validating the blocks, participants who attempt to solve the issue are commonly denoted as "miners". The complete Bitcoin mining procedure consumes a lot of energy and takes a long time to complete the process [9, 10].

Proof of Stake
PoS is an energy-efficient alternative to the PoW algorithm. The node with more cryptocurrencies has a greater opportunity to verify transactions, and its algorithm is straightforward. This mechanism is based on the idea that if a node has more property on the chain; they will be more motivated to keep the system true. PoS can significantly decrease computing power compared to PoW because it does not include resolving compute-intensive problems. However, PoS does not improve the scalability of transaction verification [9, 10].

Proof of Authority
Gavin Wood, cofounder of Ethereum and former CTO, proposed the PoA consensus algorithm in 2017. A real solution for scaling up blockchain networks is presented in it. Although its concept is like that of PoS, PoA determines the right to verify transactions based on the reputation of nodes rather than the number of cryptocurrencies. To put it another way, transactions are checked by nodes that have been chosen to be reliable. PoA is a highly scalable system because of their partial number of block validators. Companies can utilize the advantages of blockchain technology while maintaining their privacy with the PoA model. The PoA is being used, for example, in Microsoft Azure [9, 10].

Proof of Burn
A consensus algorithm for state agreement and Blockchain network validation is known as Proof of Burn. It is regarded as an alternative to PoW that aims to stop people from spending cryptocurrency coins twice. Coins must be paid for to become a validator block node. Validated coins, on the other hand, are destroyed or burned. PoB avoids wasting resources and money by not performing the costly validation process [8].

5 SWOT Analysis of Blockchain in Automotive

SWOT analysis provides a quick overview of all the important considerations that must be made illustrated in Fig. 4.

Strengths:	***Weakness***:
• *Decentralization & Availability*	• *Immature, early stage of development*
• *Efficiency in operations*	• *Scalability Issues*
• *Cyber resilience*	• *High Energy Consumption*
• *Integrity & Tamperproof*	• *Low Performance*
• *Human Error Reduction*	• *Lack of Interoperability*
• *Modern Cryptography and Safety*	• *Lack of Skilled Human Resources*
• *Asset Traceability and Provenance*	• *Complexity*
• *No Data Loss & Modifications*	• *Lack of Trust in New Technology*
Opportunities:	***Threats***:
• *Fraud reduction*	• *Institutional adoption barriers*
• *Reduced systemic risk*	• *Medium or long-term investment*
• *Network effect*	• *Technological vulnerabilities*
• *Open-source code*	• *Perception of insecurity*
• *Ease in cross-border trade*	• *Lack of Awareness*
• *Reduction of verification procedures*	• *Low adoption from external actors*
• *Digital twin enabler*	• *Unfavourable government policies*
• *Circular economy enabler*	

Fig. 4 Blockchain SWOT analysis in automotive

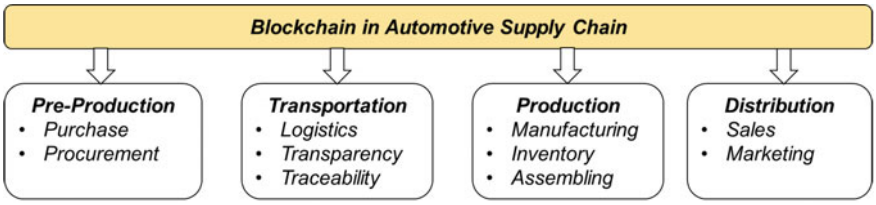


Fig. 5 Scope of blockchain in automotive

- **Strengths**: The primary advantages are efficiency and operational resilience. By eliminating intermediaries, transaction fees can be reduced, and data can be shared directly among parties without the use of a third party.
- **Weakness**: The technology's early stage, high implementation costs, high energy consumption, low performance, and slow process are major flaws.
- **Opportunities**: Integration of data for supply chain management, a safe Internet of Things, and cybersecurity.
- **Treats**: Ability to scale up, laws, and policies from the government to put the technology into use.

6 Role of Blockchain in Automotive Supply Chain Industry

In the automotive sector, a wide range of initiatives and partners can make use of blockchain. With the growth of industry 4.0, its output value is promising. Numerous global sectors are significantly impacted by blockchain technology. In the auto industry, buying and other data can be stored digitally in a blockchain [11, 12] (Fig. 5).

Integration of Blockchain in Automotive Supply Chain and different phases:

- Preproduction
- Transportation
- Production
- Distribution

7 Distributed Vehicle Ledger

Decentralized car ledger maintenance is made possible via blockchain. Manufacturers and suppliers are the main parties who keep track of component identities. The authenticity of the car's auto parts can be controlled by all stakeholders. Additionally, the vehicle ledger keeps track of every accident, occasion, and condition information. The vehicle's conditions and maintenance data, including replaced components, repaired components, and maintenance records, should be updated on the blockchain

Fig. 6 Categories of vehicle data in blockchain

Vehicle Ledger Data Summary	
Type of Data	*Overview*
Customer Identity	• *Customer Profile* • *Ownership* • *Customer Credits* • *Driving Preferences*
Parts & Components Identity	• *Component Details* • *Part Details* • *Vendor Details* • *Production Date* • *Batch & Serial Number*
Vehicle History	• *Service Records* • *Accidents & Claims Details* • *Maintenance History* • *Wears & Tears etc*

platform by all allowed parties and communicated to others. Other stakeholders can regularly validate the most recent data, and networked sensors can keep an eye on the health of the vehicles in real time [13] (Fig. 6).

8 Non-Fungible Tokens in Automotive

NFT (Non-Fungible Token) is a digital identifier that cannot be replicated, substituted for, or subdivided. It is recorded in a blockchain and is used to confirm identity and authenticity. An NFT's ownership is recorded in the blockchain and can be transferred by its owner, making it possible to trade and sell. NFT's could replace a vehicle's paper deed or electronic title stored on a centralized database because they digitally record asset ownership [14]. An Italian automaker made the announcement earlier this year that it will use NFTs to record and store maintenance records for its new SUV on a blockchain. It would be the first automaker to use NFTs in this way with this implementation. In a car market that frequently relies on third parties to track vehicle records, this could increase transparency and efficiency. One example is the automotive industry, which is just getting started. There will be an increasing number of NFT applications in the automotive industry [15].

9 Blockchain Usecases in Automotive

The automotive sector makes use of blockchain in the following ways [16] (Fig. 7):

- **Insurance**: A vehicle's mileage can be thoroughly verified and secured using the blockchain, allowing drivers who don't drive often to save money on insurance.

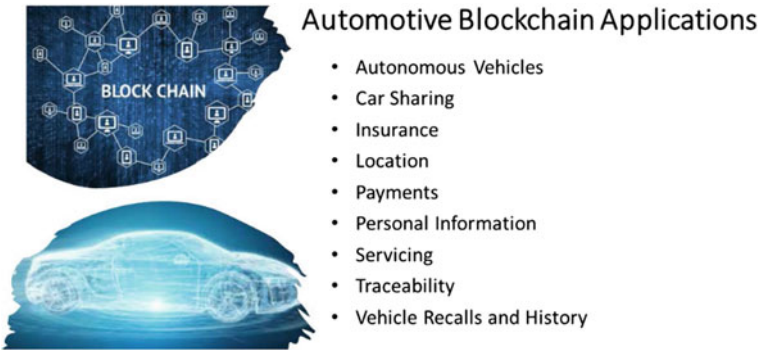


Fig. 7 Automotive applications in blockchain

- **Carsharing**: An original equipment manufacturer (OEM) provides timeshare vehicles. A single-source, usage-based payment system instantly settles transactions between titleholders, operatives, and third-party facility providers, and tours are tracked on the blockchain.
- **Payments**: E-contracts are great assets that blockchain helps to control because they allow buyers and sellers to send money directly and safely without using a bank or other intermediary.
- **Individual information**: On the blockchain, driver info can be kept safe and secure, allowing a vehicle to be immediately tailored to a user's priorities, such as setting the temperature, mirror, seat, and voice.
- **Location**: Fleet Owners can locate all their vehicles and employees with RFID tags and information stored in the blockchain. This could result in effort and travel time savings, particularly in the business of parcel delivery.
- **Traceability**: Blockchain can be used in the supply chain domain to check base, and status of materials used to make car came from, right down to the raw material when it comes out of the mine.
- **Vehicle Recall and History**: The fact that a vehicle's VIN can be stored in the blockchain means that, in the event of a recall, owners of the vehicle with a problem can be intimated, resulting in a significant reduction in manufacturing costs and inconvenience for drivers.
- **Autonomous Vehicles**: For autonomous vehicles to be safe, vast amounts of information need to be handled, analysed, and distributed promptly and safely. This can be done in a reliable and secure manner using the blockchain.
- **Servicing**: Car service history can be stored on blockchain by simply scanning a QR code. The technology can also be used to make sure that only OEM-supplied original parts are used for maintenance and replacements.

10 Mobile Open Blockchain Initiative (MOBI)

To provide a plan for overcoming obstacles to Ledger System adoption, MOBI is replacing isolated and secretive research and development (R&D) with collaboration among technology developers, automobile manufacturers, transportation authorities, and other stakeholders. The primary objective of MOBI's first project is to create a vehicle-specific identity system known as a "car passport" that tracks mileage and all relevant data on the blockchain [17]. Insurance applications that run on the blockchain are also a target for development now that vehicle identity has been determined. Insurance companies can use an immutable data feed to determine a vehicle's history and apply a good pricing model based on the perceived riskiness of drivers or vehicles based on age, gender, make, or model. Use-based mobility pricing is the name given to this model. Additionally, blockchain could automate the processing of insurance claims, reduce information asymmetries in the used car market, and improve accuracy and verifiability [18].

11 Emerging Technologies in Automotive and Role of Blockchain

The growth of smart mobility is aided in multiple ways by several emerging technologies or clusters of emerging technologies. Smart sensors, Connected Mobility, Blockchain, Digitalization, Big data, AI (Artificial Intelligence), and the Internet of Things (IoT) are the most important for the period up to 2030. Major shifts in mobility are being driven by the availability of new technologies now. Examples of these emerging technologies include C-V2X and 5G connectivity technologies, environmental observation systems (smart sensors), AI (Artificial Intelligence), Big Data, and Blockchain [19, 20]. The combination of Technologies Internet of Things (IoT), artificial intelligence (AI), and big data, blockchain has significantly more potential to disrupt the status quo than it does on its own [21, 22].

Innovative Smart Mobility applications are based on data collection, storage, processing, and analysis. For future mobility options, emerging technologies that facilitate these data-driven processes are therefore essential. Technologies Integration and data flow are shown in Fig. 8.

12 Data Security in Automotive Using Blockchain

Blockchain has the potential to completely alter the automotive industry. Blockchain technology is advantageous for every process in the automotive value chain, including purchasing, manufacturing, distribution, and service functions due to its distributed,

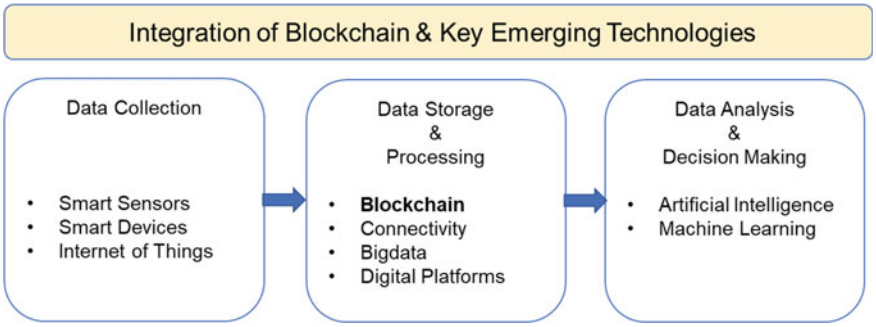


Fig. 8 Key emerging technology integration with blockchain

immutable, and transparent nature. Users' data is protected by blockchain technology, and self-driving cars can access this data to improve road navigation. This data will be able to be stored in a standard, decentralized accounting format. With efficient data sharing, connected car technology will function effectively. Because of the decentralized ledger of blockchains, every node on the network, in this case, every car, and data point—has access to all data concurrently and precisely. Blockchain innovation offers automakers a genuine chance to abandon their ongoing bumbling strategic policies and embrace new and more effective approaches to carrying on with work. The automotive sector has already seen some intriguing blockchain developments. Blockchain grants users and component suppliers extraordinary levels of control. Blockchain-based systems promise immutability and built-in verification for all organizations, which is why automakers are making strategic investments in this disruptive technology.

13 Conclusion

With the assistance of blockchain technology, automakers stand a real chance of implementing novel business strategies that are more efficient. There have already been some intriguing blockchain developments in the automotive industry. Blockchain provides unprecedented levels of control to users and component suppliers. Automakers are strategically investing in this disruptive technology because blockchain-based systems promise immutability and built-in verification for all organizations. In defiance of the non-collaborative organizational structures that are currently in place, blockchain technology has the potential to provide the automotive industry with a platform that can allocate information that is trustworthy and cyber-resilient. It is essential to carry out an objective assessment when deciding whether to capitalize in blockchain from a cybersecurity and business standpoint, despite the hype that has been reported by several organizations. This review

covered a wide range of topics regarding the emergence of disruptive technologies like blockchain. In addition, we provide a comprehensive perspective on the blockchain-based advanced automotive industry and go over the various use cases.

References

1. Subedi P, Yang B, Hong X (2022) Synchronizing tasks for distributed learning in connected and autonomous vehicles. J Commun Netw 24(4):393–407. https://doi.org/10.23919/JCN.2022.000028
2. Taherifard N, Simsek M, Lascelles C, Kantarci B (2020) Attention-based event characterization for scarce vehicular sensing data. IEEE Open J Veh Technol 1:317–330. https://doi.org/10.1109/OJVT.2020.3024755
3. Mohamed SAE, AlShalfan KA (2021) Intelligent traffic management system based on the internet of vehicles (IoV). J Adv Transp 2021(4037533):1–23. https://doi.org/10.1155/2021/4037533
4. Fraga-Lamas P, Fernández-Caramés TM (2019) A review on blockchain technologies for an advanced and cyber-resilient automotive industry. IEEE Access 7:17578–17598. https://doi.org/10.1109/ACCESS.2019.2895302
5. Khelifi A, Aziz O, Farooq MS, Abid A, Bukhari F (2021) Social and economic contribution of 5G and blockchain with green computing: taxonomy, challenges, and opportunities. IEEE Access 9:69082–69099. https://doi.org/10.1109/ACCESS.2021.3075642
6. Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B (2019) A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access 7:82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045
7. Ramesh VKC, Kim Y, Jo J-Y (2020) Secure IoT data management in a private ethereum blockchain. In: 2020 IEEE 44th annual computers, software, and applications conference (COMPSAC). IEEE, pp 369–375. https://doi.org/10.1109/COMPSAC48688.2020.0-219
8. Jabbar R et al (2022) Blockchain technology for intelligent transportation systems: a systematic literature review. IEEE Access 10:20995–21031. https://doi.org/10.1109/ACCESS.2022.3149958
9. Jiang Y-T, Sun H-M (2021) A blockchain-based vehicle condition recording system for second-hand vehicle market. Wireless Commun Mob Comput 2021(6623251):1–10. https://doi.org/10.1155/2021/6623251
10. Verma S, Yadav D, Chandra G (2022) Introduction of formal methods in blockchain consensus mechanism and its associated protocols. IEEE Access 10:66611–66624. https://doi.org/10.1109/ACCESS.2022.3184799
11. Reddy KRK, Gunasekaran A, Kalpana P, Sreedharan VR, Kumar SA (2021) Developing a blockchain framework for the automotive supply chain: a systematic review. Comput Ind Eng 157:107334. https://doi.org/10.1016/j.cie.2021.107334. ISSN 0360-8352
12. Song JM, Sung J, Park T (2019) Applications of blockchain to improve supply chain traceability. Procedia Comput Sci 162:119–122. https://doi.org/10.1016/j.procs.2019.11.266. ISSN 1877–0509
13. Wang X, Wang Y, Liu A (2020) Trust-driven vehicle product-service system: a blockchain approach. Procedia CIRP 93:593–598. https://doi.org/10.1016/j.procir.2020.04.149. ISSN 2212–8271
14. Nadini M, Alessandretti L, Di Giacinto F, Martino M, Aiello LM, Baronchelli A (2021) Mapping the NFT revolution: market trends, trade networks and visual features. Sci Rep 11(20902):1–11. https://doi.org/10.1038/s41598-021-00053-8
15. Cornelius K (2021) Betraying blockchain: accountability, transparency and document standards for non-fungible tokens (NFTs). Information 12(358):1–17. https://doi.org/10.3390/info12090358

16. Al-Jaroodi J, Mohamed N (2019) Blockchain in industries: a survey. IEEE Access 7:36500–36515. https://doi.org/10.1109/ACCESS.2019.2903554
17. Powell LM, Schwartz J, Hendon M (2021) The mobility open blockchain initiative: identity, members, technologies, and future trends. In: Revolutionary applications of blockchain-enabled privacy and access control, p 20. https://doi.org/10.4018/978-1-7998-7589-5.ch005
18. Gerger A (2021) Blockchain technology in the automotive industry: use cases and statistical evaluation. In: Industry use cases on blockchain technology applications in IoT and the financial sector, p 32. https://doi.org/10.4018/978-1-7998-6650-3.ch012
19. Retrieved from https://www.mgmotor.co.in/content/dam/mgmotor/documents/mg-dc-pdf-0261.pdf
20. Kaur M, Khan MZ, Gupta S, Alsaeedi A (2022) Adoption of blockchain with 5G networks for industrial IoT: recent advances, challenges, and potential solutions. IEEE Access 10:981–997. https://doi.org/10.1109/ACCESS.2021.3138754
21. Salah K, Rehman MHU, Nizamuddin N, Al-Fuqaha A (2019) Blockchain for AI: review and open research challenges. IEEE Access 7:10127–10149. https://doi.org/10.1109/ACCESS.2018.2890507
22. Sunny FA et al (2022) A systematic review of blockchain applications. IEEE Access 10:59155–59177. https://doi.org/10.1109/ACCESS.2022.3179690

A Short Survey on Fake News Detection in Pandemic Situation Towards Future Directions

Rathinapriya Vasu and J. Kalaivani

Abstract In December 2019, numerous news posts regarding the situation of COVID-19 in electronic media, traditional print, and social media emerged. These media sources are getting information from both non-trusted and trusted medical healthcare systems. The fake news regarding the pandemic situation is spread rapidly in multimedia. The wide spread of small deceptive information can cause unwanted exposure and anxiety for taking medical remedies and gives tricks for digital marketing. These unwanted medical remedies may lead to deadly factors. Therefore, a model to detect fake content and misinformation from the multimedia news pool during the pandemic is essential. The purpose of this systematic literature review is to review the significant studies about misinformation and fake news during COVID-19 on social media. In this survey, the currently used deep structured architectures to detect FND are utilized in the post and pre-pandemic situation. This review also explores the algorithmic categorization, dataset details, performance metrics, and tools utilized for implementation. Finally, this review gives the research gaps towards the future direction that has been aroused in the FND approaches, and it provides the appropriate pathway for researchers to implement several improvements in the future for getting better outcomes with a limited number of input data.

Keywords Fake news detection · Pandemic situation · COVID-19 · Machine learning techniques · Deep learning techniques · Algorithmic categorization · Performance metrics · Dataset details · Implementation tools

R. Vasu (✉) · J. Kalaivani
Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India
e-mail: rathinapriyavasu7@gmail.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_34

1 Introduction

People can easily share information with the extensive usage of social platforms and mobile phones anytime and anywhere. False news is "articles that are verifiably and intentionally false, and that should mislead readers". Moreover, fake news is stated as "the false news that has been spread under the guise of authentic news, and that is spread throughout the internet or outlets with an intention for gaining the data financially and politically based on increased readership and biased public opinion". The dense distribution and the continuous distribution of contents may cause significant impacts on society. On February 8, 2021, there were 106 million people confirmed with COVID-19 around the whole world, and there are 2 million people who died due to COVID-19 in the entire world [1]. Moreover, COVID-19 Hoaxes can be highly outspread through the multimedia such as Twitter, Facebook, etc.

Fake news can be spread through a social network based on images and content [2]. The manual FND over multimedia needs expert knowledge and an excessive workload to detect false information. Therefore, several tools are implemented to identify false content on Twitter, Facebook, and other social networks. But, this manual detection is not suitable for sizeable dimensional news articles [3]. To rectify these disadvantages, several semantic-based and artificial intelligence-based FND approaches have been developed. This artificial intelligence system creates a bottleneck effect when the optimization and tuning of a large number of parameters. Another difficulty of the FND over multimedia is the nature of long texts because long texts contain more entities based on the involvement of places, persons, and particular occasions.

Thus, the advancement of artificial intelligence approaches, such as supervised learning and machine learning-based FND schemes, has been utilized to provide better outcomes [4]. The aspects used to detect false news identification systems in the machine learning-based techniques are multimedia information, propagation information, social context information, and text content of news. Several architectures have been implemented for detecting the context information and the text content over the social platform. The supervised learning models are initially trained for obtaining high classification results about fake articles or real articles with the usage of "Convolutional Neural Network (CNN), Neural Network (NN), and Long Short Term Memory (LSTM)". But, these detection models take more time over FND, and the robustness of the system is low. Hence, several unsupervised learning-based false news identification approaches have been implemented. The commonly used unsupervised learning-based approaches for the FND over social media are the Gibbs method, Random forest, and logistic regression [5]. These methods effectively identify the hidden patterns in the news articles and the detection results. But several improvements are required to enhance the system's scalability and reduce the time and computational complexity with a limited number of samples. Hence, this survey helps to identify the features and challenges of recently used FND models towards its future directions in the pandemic situation.

The essential contributions of this survey paper are explained below:

- To provide a deep survey about the current advancements in the false news detection approach over multimedia in a pandemic situation with several categorizations of works.
- In this survey, recently utilized datasets, the implementation tools, classifiers, and the performance measures used for the FND are deeply analyzed.
- To summarize an effective research gap based on the existing FND approaches to make several improvements for resolving the challenges aroused in the FND models.

The remaining sections utilized for describing the survey are explained as follows: Sect. 2 gives the current developments in the false news detection methods over social networks during the pandemic and describes datasets and the implementation tools used for these currently developed techniques. Section 3 explains the algorithms used for the classification stage and also the performance measures that were taken into consideration for the existing FND. Section 4 summarizes the research gaps toward future directions, and Sect. 5 concludes this survey.

2 Literature Review on Fake News Detection in a Pandemic Situation

2.1 Related Works

In 2021, Li et al. [6] have employed a new unsupervised learning approach for FND during the pandemic situation. At first, some information was obtained from social networks to integrate user information, images, propagation, and text content. The confidential information between features was retrieved using an autoencoder with the addition of a gated recurrent layer and the self-attention layer. Finally, two real-world datasets were considered for the performance evaluation. In 2021, Shishah [7] has proposed an FND approach based on the export system model. This framework has been developed by combining relational and entity-based features to detect fake news. Further, two real-world datasets were considered for comparative analysis over various FND approaches.

In 2021, Cherifi et al. [8] have demonstrated a machine learning-based scheme for the prediction of false content on social media. The binary classification has been performed on the information to measure the complex network and identify the user profile features independently. Extensive performance validation has been carried out various export systems in terms of different evaluation measures. In 2022, Agarwal et al. [9] have recommended an FND technique by extracting the spatial and temporal features from the Meta-data of the news articles. The spatial features from the news contents were determined through NER tagging, and the temporal features were identified through the supervised learning model. Hence, the false information was effectively predicted using these spatial and temporal features.

In 2021, Song et al. [10] have introduced an FND framework using a graph-based network. The redundant information from the original news, like temporal information, content semantics, and structures, has been fused using the dynamic diffusion network. Thus, the large-scale real-world datasets have been utilized for the computation of the performance of the developed graph-based model. In 2022, Sedik et al. [11] have offered a deeply structured related false news prediction approach over multimedia platforms. The most commonly used GLOVE embedding scheme was utilized to perform text encoding. Then the encoded words were embedded into a specific word length using the concatenation of CNN, GRU, and LSTM. The Kaggle dataset was compared with different baseline techniques to validate the efficacy.

In 2021, Vishwakarma and Meel [12] have presented a self-ensembling concept related to the semi-supervised CNN approach for FND in social media. The stylometric and linguistic features from the given annotated news articles are being explored to find hidden patterns. After, neural networks were utilized to predict the unknown proxy labels. The performance of the developed framework has been evaluated through the current baseline approaches while taking comparative measures like precision and accuracy. In 2021, Liu et al. [13] have framed a deep triple network for FND by leveraging the knowledge graphs from the news articles. In this, the extracted graphs, background knowledge graphs, and open knowledge graphs were taken to extract high-level and low-level information from the news articles. The results indicated that the DTN-based approach had a better outcome than the conventional FND models.

In 2021, Nasir et al. [14] have initiated to development of a hybrid deep structure-related approach to detect fake information over multimedia. The hidden patterns in the hybrid network have been determined to provide better detection accuracy over fake news. The effectiveness of the designed FND method has been compared over different benchmark datasets, and the implementation results demonstrated that this method gets higher generalization ability than others. In 2020, Kaliyar et al. [15] have implemented a DCNN for the automatic identification of misinformation instead of using the handcraft-related information from news articles. The discriminatory features in the news article have been determined by integrating multiple hidden layers. Thus, distinct benchmark datasets were utilized to compare the developed method with the conventional false information prediction methods during the pandemic.

In 2021, Goldani et al. [16] have designed a capsule NN to detect fake news. A static word embedding technique has been introduced to update and uptrain the long-length and medium-length news. Thus, the different features have been extracted using various levels of n-grams. Finally, this model has been evaluated with two different FND methods, and the simulation outcome showcased that this accomplished high sensitivity than existing FND methods. In 2021, Kaliyar et al. [17] have aimed to develop a tensor factorization approach for the detection of false news, which relates to the deep structures for FND. The latent representation of social context and the information were obtained using a coupled-matrix-based tensor factorization model. After, the classification between social context and the information was obtained using the deep neural network by tuning optimal hyperparameters within

the system. From the classification results, the designed model outperformed well than the other models.

In 2022, Li et al. [18] have designed a web of things categorization framework by using fuzzy logic approach over social networks. Initially, the features from the news articles were extracted through pre-trained CNN. Then the high dimensionality indexing method was used to retrieve the redundant characteristics from the features. Secondly, the image and the data have been categorized through the ranking method, and thirdly, the fraudulent web contents have been determined using a fuzzy approach. Finally, parametric analysis has been done to validate the efficiency. In 2021, Kaliyar et al. [19] have suggested a BERT-based FND approach by mapping various parallel blocks of the single-layer DCNN with different filters and kernel sizes. This has produced high ambiguity in the natural language processing for classifying false news. The developed model performed well than the other baseline FND approaches.

In 2022, Malhotra and Malik [20] have recommended a fast text-modified embedding-based fake news identification approach via A-square CNN (ASCNN). The features from the news articles have been analyzed through the n-grams and B-GRU model. Thus, the classification of false news has been performed using AMSGradAdamNC-based CNN. When comparing the effectiveness of various classifiers, the developed approach has been found to be highly robust, fast, and more accurate than others. In 2020, Reddy et al. [21] have employed a text mining-based false news identification model with the usage of ensemble methods. The actual contents of the news articles have been identified through a text mining approach then the classification process has been carried out over these news articles. The accuracy of the newly designed approach has been high when compared to other conventional fake news identification approaches.

In 2021, Asghar et al. [22] have explored a rumor detection approach using deep learning methods. Contextual information has been emphasized in both backward and forward directions. Then the rumor news was predicted through Bi-directional LSTM, and the results were classified as non-rumors and rumors. Further, statistical analysis has been performed to find the effectiveness of the developed FND model. In 2019, Jain et al. [23] have demonstrated a false news prediction method with the help of a machine learning approach. The decisions about real and fake news were made by the Support Vector Machine (SVM) classifier. The evaluation outcome demonstrated this model produced higher classification results than the other approaches.

In 2019, Kaliyar et al. [24] have suggested a tree-based machine learning method for the classification of FND in pandemic situations. Here, the context-level features were combined by using the gradient boosting algorithm. Furthermore, the effectiveness of the machine structure-related FND approach has been verified through various benchmark datasets and this model provided higher detection accuracy and scalability than other models. In 2021, Khanday et al. [25] have proposed a machine learning-related technique for the classification of COVID-19 propaganda information. The relevant data from Twitter were extracted using Application Program Interface (API) on social content. In addition, the manual attenuation has been performed

individually. The decision tree algorithm has been demonstrated for the final classification of false news. The overall efficacy of the designed model has been analyzed through different classification approaches, and the results proved this model had achieved greater performance than the existing methods.

2.2 Dataset Analysis

Researchers utilize several datasets for the detection of fake news over multimedia. A real-time collection of information is also present in this dataset. Various types of datasets utilized for FND in the pandemic situation are given in Table 1. Most of the researcher uses Kaggle, FakeNewsNet, and Politifact dataset to detect false information. In this survey, various dataset was investigated to detect fake news in social media. Here, the various datasets from the current research work are analyzed. Moreover, an effective investigation is made to identify fake news during the pandemic. Moreover, the dataset helps to examine real-time fake news processing. Hence, the data are collected from the website and social media for the practical analysis of fake news in the pandemic situation.

2.3 Chronological Review

The year-wise implementations of the FND approach are depicted in Fig. 1. In this survey, the review of FND approaches is taken in 2019–2022 to know the recent advancements made in detection approaches. Most of the FND approaches will be taken in the year 2021. The current research on the FND method over social platform is taken for this review.

2.4 Study on Implementation Tools

The FND methods in the pandemic situation are carried out through various implementation tools like Linux, python, BiGCN, NVIDIA, TESLA, Tensor flow, and Keras. The percentage analysis of implementation tools on recently developed FND method is showcased in Fig. 2. From this analysis, the NVIDIA tool is mainly used for conducting experiments over false news detection models.

Table 1 Datasets used for the find in a pandemic situation over multimedia

Citation	Datasets
Li et al. [6]	MediaEval Weibo dataset
Shishah [7]	Politifact and pymedia
Cherifi et al. [8]	Poynter politifact and lead stories
Agarwal et al. [9]	FactCheck, observador and snopes
Song et al. [10]	Weibo, FakeNewsNet, and twitter
Sedik et al. [11]	Kaggle
Vishwakarma and Meel [12]	Kaggle dataset
Liu et al. [13]	2016 US election and Brexit
Nasir et al. [14]	ISO and FA-KES
Kaliyar et al. [15]	Kaggle, FakeNewsNet
Goldani et al. [16]	LIAR, ISOT
Kaliyar et al. [17]	BuzzFeed, PolitiFact, FakeNewsNet
Li et al. [18]	kaggle, FakeNewsNet
Kaliyar et al. [19]	The real-world dataset collected by U.S. general presidential election-2016
Malhotra and Malik [20]	Kaggle
Reddy et al. [21]	FakeNewsNet, McIntire, Politifact
Asghar et al. [22]	COAE2014 and IMDB
Jain et al. [23]	Media news dataset
Kaliyar et al. [24]	Kaggle
Khanday et al. [25]	Twitter API

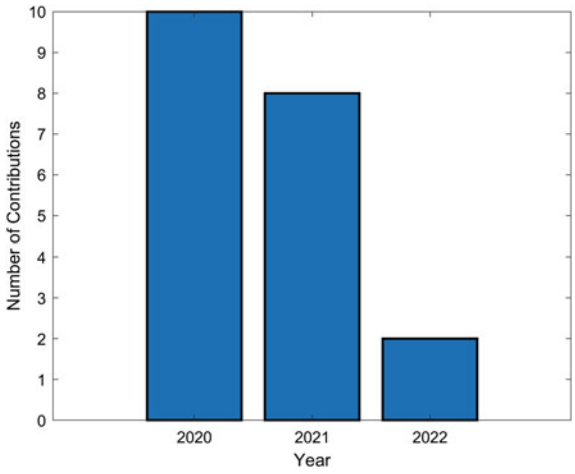


Fig. 1 A chronological review of FND techniques over social media

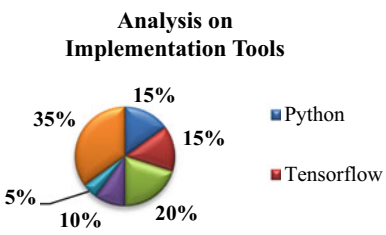


Fig. 2 Analysis of implementation tools used for fake news detection

3 Deep Learning Algorithms Used for FND in Pandemic Situation

3.1 Algorithmic Classification

The final detections of accurate and fake information are identified in the final classification stage. There are several machine learning and deep structure-related classifiers utilized for this detection of false information in the pandemic situation. These classifiers are effectively found in the hidden patterns in the original news articles. These redundant features are mainly helpful for the FND over multimedia. The most generally used deep learning-based classifiers to detect fake news are CNN, RNN, DCNN, autoencoder, LSTM, and BERT. The most commonly utilized machine learning-related classification algorithms [26, 27] for identifying false news are SVM, LR, fuzzy, and decision tree.

CNN-Based FND Models CNN-based detection schemes broadly classify accurate and false information. CNN gives several advantages like low computational complexity, fewer costs, and higher detection accuracy. Moreover, it effectively finds hidden features from the original information. Hence, misclassification is greatly minimized by using this CNN classifier. In this survey, the CNN-based FND approaches are present in CNN-LSTM-GRU [11], CNN-RNN [14], DCNN [15], and A-Square CNN [20].

Neural Networks Based FND Models Neural network based FND schemes produce high sensitivity and high detection accuracy over real news and false news, and this is highly suitable for classifying with large amounts of input data. The complex structure has been effectively analyzed over these neural networks based FND approach. The neural network based classifiers utilized for this survey are RNN [13], capsule neural network [16], DNN [17], and DNN [22].

Other deep learning-based approaches used for taking a review of this survey are Autoencoder [6], BERT [7], LSTM [9], GCL [10], BERT [19], and EL [21].

Machine Learning-Based FND Techniques These approaches are also helpful for the FND in the pandemic situation. These techniques provide high generalization ability, reducing the overfitting and the underfitting problems. The machine learning structure-related classification algorithms to be considered for the detection of fake

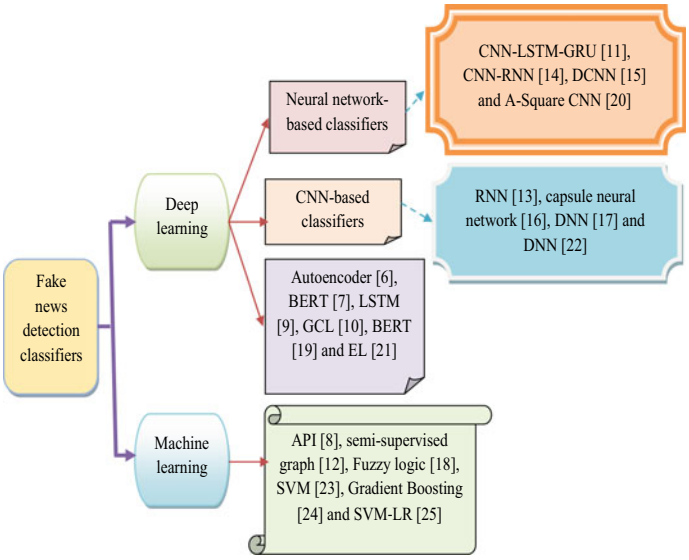


Fig. 3 Algorithmic categorization of recently used FND approaches

information are API [8], semi-supervised graph [12], Fuzzy logic [18], SVM [23], Gradient Boosting [24], and SVM-LR [25]. The algorithmic classification of recently used FND models in the pandemic situation is diagrammatically represented in Fig. 3.

3.2 Evaluation Metrics Utilized in FND

The performance metrics used for evaluating the effectiveness of the FND models. The performance validation is also carried out through these performance metrics. The most widely used performance metrics for the FND methods are "accuracy, precision, F1-score, recall" and some negative measures like "True Positive Rate, False Positive Rate, and False Negative Rate". The currently used performance measures for FND are illustrated in Table 2.

4 Research Gaps and Challenges

In COVID-19 related pandemic situation, the wrongdoers spread smear campaigns, satire, clickbait, astroturf memes, conspiracies, hoaxes, fake news, false articles, misinformation, rumors, false content, and other forms of deception on social media. This gives tremendous strain on society in the form of damaging justice, truth, democracy, journalism, freedom of expression, public trust, and reputation. Hence,

Table 2 Performance metrics used for performance validation of conventional FND systems

Citation	Accuracy	Precision	F1-score	Recall	Miscellaneous
[6]	–	✓	–	–	AUC, MACRO-F1, and MICRO-F1
[7]	✓	✓	✓	✓	TPR and FPR
[8]	✓	–	–	–	–
[9]	✓	✓	✓	✓	–
[10]	✓	✓	✓	✓	–
[11]	✓	✓	✓	✓	–
[12]	✓	✓	✓	✓	–
[13]	✓	✓	✓	✓	–
[14]	✓	✓	✓	✓	–
[15]	✓	✓	✓	✓	–
[16]	✓	✓	✓	✓	TPR and FPR
[17]	✓	✓	✓	✓	–
[18]	✓	✓	✓	✓	–
[19]	✓	–	–	–	Cross-entropy loss, FPR, FNR, and confusion matrix
[20]	✓	✓	✓	✓	–
[21]	✓	✓	✓	✓	–
[22]	✓	✓	✓	✓	–
[23]	✓	–	–	–	–
[24]	✓	–	✓	–	Detection time and Training loss
[25]	✓	✓	✓	✓	–

paramount importance is given to detecting false news on social media during the pandemic. Many of these techniques have been developed for FND on multimedia based on the profile of new generators, propagation of the news on users, and the tweets content. The implementation of network-based methods relies on the complete graph, which takes time for FND on the social platform. Developing user profile-based methods is only suitable for fake accounts and bot detection. It cannot detect false information and real information. Moreover, the human content-related detection approaches allow only deceiving content-based techniques. In addition, false news is creating significant problems in today's world. The early FND faces several challenges because of the enormous input data. Another significant challenge that arises in the detection of false news is the shortage of labeled training models and shortage of training data. If similar patterns are present in the input data means, that leads to several difficulties in the classification of real versus false news. Further developments are needed to improve the model accuracy with a limited amount of input data. In addition, the model reliability and robustness of the system have been further improved in the future. Moreover, the time and computational complexity of the false news detection approaches are also improved in the future.

5 Conclusion

This paper reviewed the previously used FND technologies with the exploration of different deep structure-related techniques in the pre-and post-pandemic situations. The recent developments in the deep structure and supervised and unsupervised FND models have been deeply explained in this survey. It mainly focused on the type of datasets utilized for the recent developments in FND approaches and also the implementation tools being used for conducting experiments over the false news detection approach. In addition, the classifiers used for the identification of false information and the performance metrics used to validate the effectiveness of the model have been analyzed. Finally, the research gaps toward future direction were explained to encourage the researchers to further develop the FND models over multimedia networks.

References

1. Olaleye T, Abayomi-Alli A, Adesemowo K, Arogundade OT, Misra S, Kose U (2023) SCLAVOEM: hyper parameter optimization approach to predictive modeling of COVID-19 infodemic tweets using smote and classifier vote ensemble. Soft Comput 27:3531–3550
2. Batzdorfer V, Steinmetz H, Biella M, Alizadeh M (2022) Conspiracy theories on twitter: emerging motifs and temporal dynamics during the COVID-19 pandemic. Int J Data Sci Anal 13:315–333
3. Mittal R, Mittal A, Aggarwal I (2021) Identification of affective valence of twitter generated sentiments during the COVID-19 outbreak. Soc Netw Anal Min 11(108):1–12
4. Lanius C, Weber R, MacKenzie Jr WI (2021) Use of bot and content flags to limit the spread of misinformation among social networks: a behavior and attitude survey. Soc Netw Anal Min 11(32):1–15
5. Verma PK, Agrawal P, Madaan V, Prodan R (2023) MCred: multi-modal message credibility for fake news detection using BERT and CNN. J Ambient Intell Humanized Comput 14:10617–10629
6. Li D, Guo H, Wang Z, Zheng Z (2021) Unsupervised fake news detection based on autoencoder. IEEE Access 9:29356–29365
7. Shishah W (2021) Fake news detection using BERT model with joint learning. Arab J Sci Eng 46:9115–9127
8. Cherifi H, Qureshi KA, Malick RAS, Sabih M (2021) Complex network and source inspired COVID-19 fake news classification on twitter. IEEE Access 9:139636–139656
9. Agarwal IY, Rana DP, Shaikh M, Poudel S (2022) Spatio-temporal approach for classification of COVID-19 pandemic fake news. Soc Netw Anal Min 12(68):1–9
10. Song C, Shu K, Wu B (2021) Temporally evolving graph neural network for fake news detection. Inf Process Manage 58(6):102712
11. Sedik A, Abohany AA, Sallam KM, Munasinghe K, Medhat T (2022) Deep fake news detection system based on concatenated and recurrent modalities. Expert Syst Appl 208:117953
12. Vishwakarma DK, Meel P (2021) A temporal ensembling based semi-supervised ConvNet for detecting fake news articles. Expert Syst Appl 177:115002
13. Liu J, Wang C, Li C, Li N, Deng J, Pan JZ (2021) DTN: deep triple network for topic-specific fake news detection. J Web Semant 70:100646
14. Nasir JA, Khan OS, Varlamis I (2021) Fake news detection: a hybrid CNN-RNN based deep learning approach. Int J Inf Manage Data Insights 1(1):100007

15. Kaliyar RK, Goswami A, Narang P, Sinha S (2020) FNDNet—a deep convolutional neural network for fake news detection. Cogn Syst Res 61:32–44
16. Goldani MH, Momtazi S, Safabakhsh R (2021) Detecting fake news with capsule neural networks. Appl Soft Comput 101:106991
17. Kaliyar RK, Goswami A, Narang P (2021) EchoFakeD: improving fake news detection in social media with an efficient deep neural network. Neural Comput Appl 33:8597–8613
18. Li T, Yu J, Zhang H (2022) Web of things based social media fake news classification with feature extraction using a pre-trained convoluted recurrent network with deep fuzzy learning. Theoret Comput Sci 931:65–77
19. Kaliyar RK, Goswami A, Narang P (2021) FakeBERT: fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools Appl 80:11765–11788
20. Malhotra P, Malik SK (2022) An efficient fake news identification system using A-SQUARE CNN algorithm. Wireless Pers Commun 125:2075–2100
21. Reddy H, Raj N, Gala M, Basava A (2020) Text-mining-based fake news detection using ensemble methods. Int J Autom Comput 17:210–221
22. Asghar MZ, Habib A, Habib A, Khan A, Ali R, Khattak A (2021) Exploring deep neural networks for rumor detection. J Ambient Intell Humanized Comput 12:4315-4333
23. Jain A, Shakya A, Khatter H, Gupta AK (2019) A smart system for fake news detection using machine learning. In: 2019 international conference on issues and challenges in intelligent computing techniques (ICICT). IEEE, pp 1–4
24. Kaliyar RK, Goswami A, Narang P (2019) Multiclass fake news detection using ensemble machine learning. In: 2019 IEEE 9th international conference on advanced computing (IACC). IEEE, pp 103–107
25. Khanday AMUD, Khan QR, Rabani ST (2021) Identifying propaganda from online social networks during COVID-19 using machine learning techniques. Int J Inf Technol 13:115–122
26. Tabjula JL, Kanakambaran S, Kalyani S, Rajagopal P, Srinivasan B (2021) Outlier analysis for defect detection using sparse sampling in guided wave structural health monitoring. Struct Control Health Monit 28(3):e2690
27. Jagadeeshwar TL, Kalyani S, Rajagopal P, Srinivasan B (2022) Statistics-based baseline-free approach for rapid inspection of delamination in composite structures using ultrasonic guided waves. Struct Health Monit 21(6):2719–2731

Periocular Biometrics and Its Applications: A Review

Aishwarya Kumar and K. R. Seeja

Abstract The periocular region is the area under the proximity of the eyes. The periocular region provides a significant number of features that can be used in situations of the occluded face. Due to face occlusion, a very large portion of the face does not contribute to the features, which may lead to unsatisfactory performance of the system. Focusing only on the periocular region will enhance the performance of the system as only these features will be considered for the development of the application. This paper provides an extensive survey of the different applications that use the periocular region for biometrics and soft biometrics. The paper provides an overview of the different preprocessing steps the researchers are using over recent years. Along with this, the paper also presents possible research opportunities in the domain.

Keywords Periocular region · Facial landmark detection algorithms · Feature extraction · Person identification

1 Introduction

The biometric system is a process that is used to identify a person by using his traits. The different biometric attributes which can be used to identify an individual are the face, fingerprints, iris, ear, voice, footprints, etc. There are some forensic methods (like DNA through blood, hair, or saliva) as well which are used in critical applications. Biometric systems can be used to authenticate a person in different scenarios like criminal investigations, surveillance systems, access-control systems, airport security systems, etc.

A. Kumar (✉) · K. R. Seeja
Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, Kashmere Gate, Delhi 110006, India
e-mail: aishwarya001mtcse21@igdtuw.ac.in

K. R. Seeja
e-mail: seeja@igdtuw.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_35

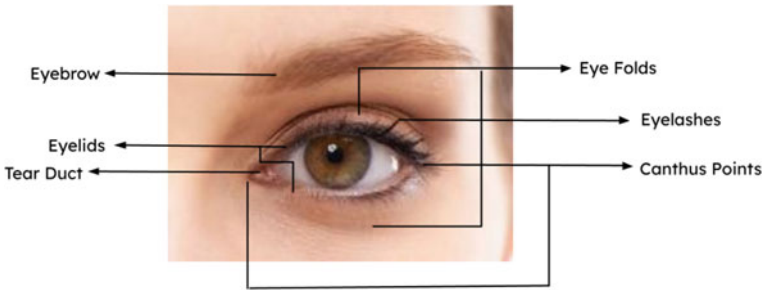


Fig. 1 Periocular portion of the face

Many a time, it is seen that the face of the person is covered with a veil or mask, making it difficult for facial image-based identification systems to identify him. Periocular Region can be used in such situations. The periocular region includes features around the eyes, like eyebrows, eye folds, eyelids, tear ducts, canthus points, etc. A detailed description of the characteristics of the periocular portion is displayed in Fig. 1.

The primary objective of this paper is to put forward an illustrative survey about the existent techniques used in literature considering the periocular area and its combination with the iris features. The periocular biometric system has been discussed in Sect. 2. Section 3 is an elaborative explanation of the various benchmark datasets which are available in the literature. The facial landmark extraction techniques which can be used as a preprocessing step are examined in Sect. 4, followed by Sects. 5, 6, and 7 which discuss the region of interest extraction methods, widely used feature descriptors, and the different classification techniques opted by the researchers. Sections 8, 9, 10, and 11 discuss the various applications of the periocular region. The possible research opportunities are mentioned in Sect. 12. The paper has been concluded in Sect. 13.

2 Periocular Biometrics System

The Periocular Biometric System can be developed by following the five-step approach shown in Fig. 2. The first module is the data acquisition step. The data can be gathered either by using the benchmark datasets which are publicly available or by capturing the periocular images. There are some periocular datasets available, discussed in Sect. 3. Another way can be generating the periocular datasets using facial images. The next step would be the preprocessing of the gathered images. The preprocessing can include various steps like dataset augmentation, image equalization, image filtering, image segmentation, etc. These steps can vary according to the application under consideration. The next phase would be to find the optimal region of size. The different procedures present in the literature for optimal ROI extraction

Fig. 2 Periocular biometrics system

have been reviewed in Sect. 5. After having the optimum region, features are derived from them. Features are useful information that will help the system to recognize the subject. The last step would be training the model and classifying the image of the person. Recognition of the person and evaluation of the results are also performed in this step.

3 Periocular Databases

To develop any application based on periocular region, a significant number of images are necessary for training and evaluation. The researcher can create their image database by capturing the images through a camera or a sensor. Datasets can also be formed by crawling the internet sources. There are some benchmark datasets available that researchers can use for their research. The benchmark datasets are those which are created by different organizations and institutions, and available freely or upon request through a licensing agreement. Some of the benchmark periocular image databases are discussed in Table 1 and a few sample of periocular images from these benchmark datasets are shown in Fig. 3.

4 Facial Landmark Extraction Techniques

The first and foremost step for developing a periocular region-based system is to have a large set of periocular images. The researchers [15–17] are using the periocular datasets which have been discussed in Sect. 3 for developing their respective applications. The other way could be to extract the periocular area from the facial images like the work done in [18]. To extract the eye vicinity portion from the face, the coordinates or landmarks of the face are required. Different algorithms can be used to perform this task. These algorithms work by first detecting the face and then extracting the coordinates from it. Among these algorithms, the Dlib algorithm has been widely used in many pieces of research like Reddy and Derakhshani [19].

Table 1 Benchmark periocular image datasets

Name of the dataset	No. of subjects	No. of images	Link to download/license agreement
Ethnic ocular [1]	1,034	85,394	https://www.dropbox.com/sh/vgg709to25o01or/AAB4-20q0nXYmgDPTYdBejg0a?dl=0
SBVPI (sclera blood vessels, periocular, and iris) [2, 3]	55	1,858	https://sclera.fri.uni-lj.si/datasets.html
MOBIUS(mobile ocular biometrics in unconstrained settings) [4]	100	16,717	https://sclera.fri.uni-lj.si/datasets.html
I–SOCIAL–DB [5]	400	3,286	https://iebil.di.unimi.it/ISocialDB/index.html
FOCS (face and ocular challenge series) [6]	136	9,581	https://cvrl.nd.edu/media/django-summernote/2018-09-19/a160e284-751e-4587-a292-7ce77e8b387b.pdf
Cross–eyed-cross-spectral iris/periocular recognition database [7]	120	3,840	https://sites.google.com/site/crossspectrumcompetition/home/how-obtain-and-cite-the-database
CSIP (cross sensor iris and periocular dataset) [8]	50	2,004	http://csip.di.ubi.pt/
IIITD multispectral periocular dataset [9]	62	1,240	http://www.iab-rubric.org/index.php/multi-spectral
UBIPr [10]	261	10,950	http://iris.di.ubi.pt/ubipr.html
UFPR–periocular dataset [11]	1,122	33,660	https://web.inf.ufpr.br/vri/databases/ufpr-periocular/
UFPR–eyeglasses [12]	83	2,270	https://web.inf.ufpr.br/vri/databases/ufpr-eyeglasses/
VISOB 2.0 (visible light mobile ocular biometrics) [13]	150 (training) 100 (testing)	75,428	https://sce.umkc.edu/research-sites/cibit/dataset.html/
CASIA–IrisV4–distance [14]	142	2,567	http://biometrics.idealtest.org/dbDetailForUser.do?id=4#/

4.1 Dlib Facial Landmark Detection Algorithm

The Dlib library which was created by King [20] is a C++ and Python-based toolkit for face localization and facial landmark identification. The algorithm works in two phases: initializing the detector to locate and detect the faces existing in the image and initializing the predictor using a pre-trained model for detecting the key points or the landmark points. The pre-existing model used for landmark localization is trained on the iBUG-300W dataset and is based on ensemble regression trees. It is a regression problem as there is a continuous series of points that needs to be predicted. The Dlib

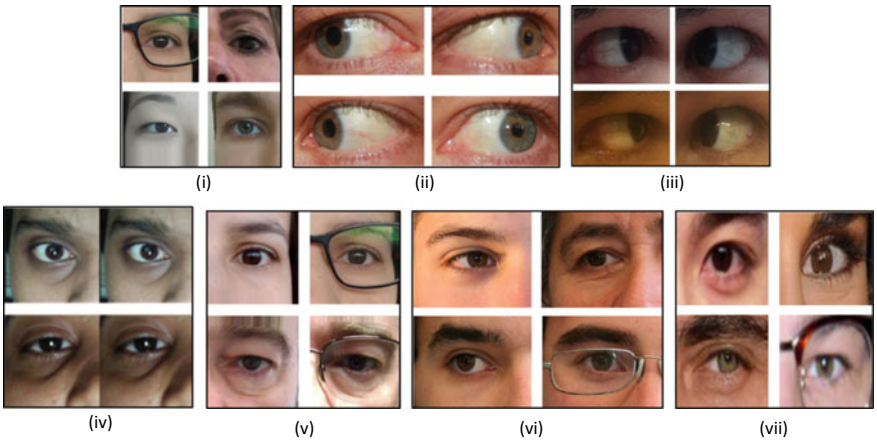


Fig. 3 Sample of periocular images from different datasets, (i) UFRP Dataset [11], (ii) SBVPI Dataset [2, 3], (iii) MOBIUS Dataset [4], (iv) VISOB 2.0 Dataset [13], (v) UFPR Eyeglasses Dataset [12], (vi) UBIPr Dataset [10], (vii) Ethnic Ocular Dataset [1]

algorithm is an implementation of the model proposed by [21]. Face localization is based on HOG, and linear SVM. This facial landmark detector algorithm detects 68 coordinates, including eyes, eyebrows, nose, lips, chin, and jawline. Out of the 68 landmark points, only 22 points are related to the periocular region. These landmark points are the coordinates of the eye and the eyebrows.

4.2 FacemarkLBF Model

The FacemarkLBF model is an implementation of the research proposed by Ren et al. [22]. This face landmark detection algorithm is supported by OpenCV but does not have Python support. To detect the key points, the algorithm has to be coded in C++ programming language. A pre-existing model is available for coordinate detection. The faces in the images are detected through the Haar Cascade classifier and then the LBF model landmark detector is initiated. Just like the Dlib algorithm discussed in Sect. 4.1, the FacemarkLBF model also returns 68 landmark points in the form of a vector.

4.3 MediaPipe Face MeshLibrary

The MediaPipe library is developed by Google and provides a set of solutions [23] for face detection, facial landmark points localization, iris detection, hand coordinates localization, pose detection, object detection, 3D object detection, etc. These

solutions are open source and most of them are operatable on both Android and iOS operating systems. These solutions provide support for C++, Python, and JavaScript languages. The MediaPipe library for landmark detection [24] works in two phases: detecting the faces from the image and then locating the landmark points. It uses a pre-trained model BlazeFace for the execution of these phases. The output of this library is 468 3D facial key points for the image passed as input. These landmarks include the positions of eyebrows, eyes, nose, lips, forehead, cheeks, chin, and jawline. Along with this basic mesh model, the library also provides an Attention-based mesh model to provide attention to the meaningful facial features.

4.4 MTCNN Model

Multi-Task Cascaded Convolution Networks (MTCNN) is an architecture proposed by Zhang et al. [25] which uses multi-task learning for both facial detection and landmarks localization. MTCNN is a Python-based algorithm, considered as a two-stage model. Before passing a facial image to the model, the image needs to be resized to form a pyramid-like structure. The model uses three CNN for extracting the coordinates of the face. The first network is the P-Net (Proposed Network) which takes the resized image as an input, performs the convolution operations, and produces the candidate bounding box coordinates for different facial regions as an output. The P-Net model has no dense layer. The next network is known as the R-Net (Refine Network) which improves the output of the P-Net and tries to detect the correct bounding box for the face available in the picture. The output of this second network is detected face along with the face bounding box coordinates. The R-Net model also produces a vector with ten elements that help to localize the facial key points. The third and last model in the MTCNN architecture is the O-Net (Output Network) which refines the output of the R-Net model and produces five facial positions as the final output. The only downside of this detection algorithm is that it takes high processing time as compared to other algorithms discussed in Sects. 4.1, 4.2, and 4.3.

5 Region of Interest (ROI) Extraction

After extracting the periocular portion, the next stage is to get the optimal region of interest through which the best results can be achieved. There is no conventional procedure that can be used to find out the optimal region of interest. Many researchers had proposed different methodologies to extract the ideal area from the periocular region. Kumari and Seeja [26] proposed an unprecedented method to extract rectangular and polygon-shaped ROI. The rectangular ROI extraction algorithm used the corner points of the eyes (known as canthus points), corner points of the eyebrows, and the midpoint of the eyebrow as the reference points. Halfway

point of the line joining the corner points was calculated. Then the distance between the midpoint of the eyebrow and the midpoint of the corner points was calculated and used as a reference to calculate the coordinates of the rectangular bounding box. The polygon-shaped ROI extraction algorithm also worked similarly.

Merkow et al. [27] used frontal and near-frontal facial images to extract the optimal periocular region from them. The near-frontal images are rotated to modify them as frontal images. To get the appropriate angle of rotation, the centroids of the eyes were taken into consideration. The proposed algorithm used the inner and outer canthus points, the peak of the nose, the midway of the upper lip, and the center of the chin as the reference points. The line joining the corner points was named as the eye line. The ROI was found by keeping the eye line as the center and considering half of the distance between the nose and the eyes, upwards and downwards the eye line.

Manyala et al. [28] used the Viola-Jones algorithm to identify the face and the eyes. The optimum size of the ROI was determined by the following coordinate equation:

$$\left(\mathrm{w} + \frac{\mathrm{w}}{a}, h + \frac{h}{b}\right) \tag{1}$$

where w and h are the width and height of the bounding box containing the eye pair detected by Viola-Jones. a and b are the hyperparameters that can be set accordingly to consider the whole periocular region (including the eyebrows). After the generation of this eye-paired ROI, the left and the right periocular region was separated.

Kumar and Pavani [29] normalized the near–frontal facial images. The center of the eyes was then located and considering the eye center as the reference point, a square-shaped bounding box was formed around each eye. Reddy and Derakhshani [19] detected the facial landmark points through the Dlib library. The inner and outer corner points and the bottom tip of the nose were considered as reference points. Eye centers were produced such that they are at three–fourth of the height from the top and one–fourth of the width from the sides. The extracted periocular ROI had both eyes.

6 Feature Extraction Techniques

After extracting the optimal periocular region, the next step which remains constant for all the applications is feature extraction. Feature extraction is the process that focuses on those useful pixel values of an image which helps to identify the image. Feature extraction plays a significant role in problems such as object detection, person identification, image reconstruction, and many more. Feature extraction techniques are broadly classified into two: handcrafted and non-handcrafted feature extraction. Handcrafted feature extraction methods (LBP, HOG, SIFT, Gabor, GLCM, BSIF, GIST, etc.) are the ones where the features are manually extracted using specially

engineered extraction techniques. Non-handcrafted feature extraction includes automatic detection and extraction of features using deep learning-based algorithms like CNN models.

6.1 Local Binary Pattern (LBP)

LBP is a global feature descriptor that analyzes the whole image as a single vector and the same rule has been applied to all the pixels of the images. To calculate the LBP code for an image, a reference pixel is considered. The reference pixel is surrounded by eight neighboring pixels. These neighboring pixels are used for generating the LBP code for the reference pixel. The generated binary values (by comparison of reference and neighboring points) are combined either clockwise or anti-clockwise and then the binary values are converted to decimal. The decimal value is the LBP code for the reference pixel. The researchers [17, 18, 30–32] used the Local Binary Pattern as an extraction method to draw out features from the periocular region.

6.2 Histogram of Oriented Gradients (HOG)

HOG is used to convert an image to a feature vector of a length selected by the user. The dimension of the input image (width: height) for the HOG feature descriptor should be in the ratio 1:2. Histogram of Gradients is calculated using 9 bins for each 8×8 pixel. HOG is calculated after block normalization of the bins is calculated. HOG is a popularly used technique for feature extraction. Hernandez-Diaz et al. [17], Kumar and Rao [30], Oishi et al. [31] and Kumari and Seeja [33] used Histogram of Oriented Gradients along with other feature extraction methods to identify the person using the periocular region.

6.3 Scale Invariant Feature Transform (SIFT)

SIFT is a local feature descriptor that starts by detecting and locating the features from an image. After locating the landmarks, orientation is assigned to them and at the end, a feature vector is generated. As the name suggests, the features generated by SIFT are invariant by changing the scale and rotation. It is a handcrafted feature extraction method used by Hernandez-Diaz et al. [17], Oishi et al. [31], Nakamura et al. [32] and Kamarajugadda and Movva [34] in their research work. Table 2 shows a comparison between these popular handcrafted feature extraction techniques.

Apart from the popularly used LBP, HOG, and SIFT feature descriptors, the researchers also used some other feature descriptors. Alonso-Fernandez et al. [18] and Sarmah et al. [35] used Gabor filters for periocular feature extraction. Gabor

Table 2 Comparison between LBP, HOG, and SIFT

	Local or global descriptor	Binary or real descriptor	Scale dependent or independent	Rotation invariant	Key outlines	Primary uses
LBP	Local feature descriptor	Real (floating point)	Dependent	Yes	Robust to monotonic grayscale changes and uses all 8 directions for each pixel to capture details only at a fixed scale	Capturing local patterns and extremely fine-grained details for face recognition and texture classification
HOG	Global feature descriptor	Real (floating point)	Dependent	No	Considers orientation histogram of edge intensity, gradient, and orientation to calculate first-order gradients of each cell in both the horizontal and vertical directions	Detecting objects by capturing edges and corners in images
SIFT	Local feature descriptor	Real (floating point)	Independent	Yes	Extract the local extrema and feature points with the difference-of-gaussian (dog) function	Detecting the interest points/key points in the facial images

filter is a modulation of Gaussian and sine waves which allow only a certain band of frequencies to pass through it. It is a texture-based feature extraction that produces the highest value at the change of texture in an image. Gray-Level Co-Occurrence Matrix (GLCM) is also a texture-based global feature descriptor that analyzes how frequently a particular pair of pixels occur together in an image, i.e., it tries to find out the co-occurrence of the different pixel values. Alonso-Fernandez et al. [18] compared the efficiency of the GLCM feature extraction technique along with other feature descriptors.

7 Classification Techniques

After preprocessing and feature extraction, the images are then put into different classes using a classification algorithm. If gender classification is the problem under consideration, then the images will be classified into two classes (Male and Female). If we are building an age classifier, the output classes will be different age groups, like Young, Middle, and Old. The output will be the name of the person in case of person identification. So, most of these problems are multi-class classification problems.

Most of the research works [15, 18, 19, 33, 36, 37] used SVM as a classification algorithm. SVM works by finding the hyperplane between the classes by minimizing the gap between the hyperplane and the support vectors. Deep learning architectures like Convolution Neural Networks (CNN) can also be used as a classifier. CNN can work as both a feature extractor and classifier. Zhou [16] and Kumari and Seeja [33] trained a CNN model from scratch while Hernandez-Diaz et al. [17] used different pre-trained models using transfer learning as a classifier. Some researchers [30, 34] have also used City Block Distance as the classifier.

8 Applications of Periocular Region

Due to Covid-19, around 70% of the subject's face is covered with a mask. The only visible and clear region is the periocular area. The periocular portion provides a large quantity of identifying features. The periocular region is being used in different aspects with the fusion of the periocular portion along with the iris. The periocular region as an individual feature or periocular region along with the iris features are being used in many applications, some of them being person recognition, emotion recognition, and age and gender classification. The research work in these areas has been discussed in Sects. 9, 10, and 11.

9 Person Identification Through Iris and Periocular Region

Person Identification is a very prominent research topic. Many researchers have worked in this area and achieved a remarkable recognition rate. In the recent past, several works have been conducted by using periocular and iris as the key identifying trait. Ripon et al. [38] used the CASIA dataset to examine the proposed model. The research started by extracting the Left, and Right ROI regions from the face. A CNN architecture is used as a feature extractor, and the images were classified using the KELM classifier. Umer et al. [39] proposed a person identification system through a combination of the iris and periocular area. The iris portion from the images of the four benchmark datasets was segmented and normalized before passing to the pre-trained VGG16, ResNet50, and Inception v3 models. These models are also separately trained using the periocular region. The final class of the person was decided on the basis of the combination of decisions of the iris-based models, and the periocular region-based models. Oishi et al. [31] also used a similar approach by integrating the iris, and periocular region features using AdaBoost. Kumari and Seeja [33] also used a HOG to extract the actionable key points from the periocular area. Along with this, the research also used non-handcrafted periocular features obtained through pre-trained CNN models. The gender of the subject was also extracted by training a raw CNN model. The fusion of these features was passed to the multi-class SVM model for person identification.

Ipe and Thomas [37] used SVM for multi-class classification. The periocular region was segmented using the Viola-Jones algorithm. The features were extracted through a transfer learning approach using the AlexNet Model. Abhinav et al. [40] used the Viola-Jones algorithm to locate the face and eyes from the live videos. Pre-trained VGG16 model was used as a feature extractor and a classifier. On the other hand, Nakamura et al. [32] opted for SIFT and LBP for feature extraction from the left and right periocular portions. XgBoost algorithm was used to fuse the recognition scores of the two models. Ipe and Thomas [15] used the Viola-Jones algorithm for periocular segmentation. High-resolution images were created through the Very Deep Super Resolution (VDSR) algorithm. Feature extraction was done using the pre-trained AlexNet Model through transfer learning and multi-class SVM was used to classify the images of the person. Zhou [16] proposed a Siamese structure for person identification through the periocular region. The Siamese structure is a fusion of two CNN models which are having different base models (backbone models). The researcher compared the efficiency of the pre-trained MobileNet model with a raw CNN through the Siamese Structure. Also, the flipped images were provided in the input to enhance the performance of the proposed methodology.

Hernandez-Diaz et al. [17] used pre-trained CNN models for classification. Feature extraction was done using handcrafted methods like LBP, HOG, and SIFT on contrast-enhanced periocular images. Kumar et al. [41] used the Haar Cascade classifier to locate the face and eyes, then a pre-trained ResNet18 model was used to classify the person through the periocular region. Kamarajugadda and Movva [34] suggested a periocular area-based biometric system using SIFT and SURF as the feature descriptors. The periocular region was extracted by taking the eye center as the point of reference. The classification was done through the City Block Distance classifier. A comparison was done between the accuracy obtained by using the left and right periocular areas with the complete facial features on the benchmark datasets. Mon et al. [42] also proposed a GUI-based periocular biometric system where the periocular portion was captured using a webcam and stored in the database for a new user. On the captured periocular images, feature extraction and score-level matching were performed. The proposed system proved to be successful in all the unit-level and integration tests.

Zhao and Kumar [43] suggested an improvement in the existing periocular identification system by providing attention to the crucial features which were eyebrows and eyes. The proposed AttNet Architecture used four convolutional units and a dense layer. Another network (fully convolutional; no dense layer) having four convolutional blocks was used to provide attention to the eyes and the eyebrows by segmenting them. Along with the architecture, the researchers proposed a verification-oriented loss function as well. The model was evaluated on six benchmark datasets. Kayande et al. [44] used pre-existing VGG16 network as a feature extractor and a combination of two neural networks for classification. One ANN was fine-tuned to classify the left periocular images while the other was used for the right periocular images. This hybrid model approach outperformed the single model (ANN and SVM) for classification. Kumar and Rao [30] evaluated the accuracy of

four different feature extraction methods on the FRGC dataset. City Block distance metrics were used as a classifier.

10 Soft Biometrics Using Periocular Region

Soft Biometrics are those characteristics of a person which may not be unique but can help with recognition and verification. These additional features when included in a biometrics identification system can enhance the results of the system. Identification of two soft biometrics traits, namely gender and age, through the iris and the periocular area has been discussed in Sects. 10.1 and 10.2.

10.1 Gender Identification Using Iris and Periocular Region

Gender classification is a binary classification problem which means that the image will either fit a Male class or a Female class (two output classes). The researchers have used different methodologies to identify the gender of a person using the iris or periocular area. Tapia et al. [45] focused on the texture of the iris to categorize the iris regions based on the gender of the person. Various experiments were performed by using different implementations of LBP, and HOG as feature extractors. The iris regions were divided into different regions and windows. The size of the window and the overlap between the windows were the hyperparameters, and their optimal value was found by conducting different experiments. The research concluded that using Uniform LBP with histogram on an overlapping window of 50% achieved the maximum accuracy. Merkow et al. [27] suggested a novel algorithm to crop the periocular area from the images of the frontal faces, and near-frontal faces. The periocular region was extracted through seven key points. The research proceeded by validating the experiment on four different regions, Region 1 which consisted of the facial region from eyebrows to chin, Region 2 which had only eyes, and nose, Region 3 which had the periocular region only, and Region 4 which considered the full facial region. Local Binary Pattern (LBP), and Pixel-based features were used as a Feature Extraction method. The gender of the subject was classified by using different ML algorithms. The research concluded by stating that only Periocular Region could be used for Gender Classification with an accuracy of at least 85% accuracy.

Viedma et al. [46] tried to find the predominant and critical features of the periocular area for gender classification. To extract the features, the SVM classifier (Gaussian kernel) along with 9 ensemble classifiers were used. The results showed that the relevant features were intensity, texture, and shape of the eye, which are not located in the iris. After extracting the 400 best features, the XgBoost classifier is used to predict gender. The research also tried to compare the performance of the classifier for full periocular iris images and iris-excluded images. Tapia and Aravena [47] presented a CNN-based methodology to predict the gender of the subject from

the Left and Right Periocular Region. LeNet5 Architecture was modified to develop two CNN models. The first CNN was trained only for the left eye images, while the other CNN was for the right eye images. The primary objective of the research was to evaluate the performance of the fusion of these CNN models. A third CNN model was also implemented which was trained on both left and right eye. The results obtained from the research supported the fact that the Fusion Model (CNN–1 for Left Eyes + CNN–2 for Right Eyes) proved to be the most efficient model with an accuracy of 87.26% in 300 epochs in the training time of 2 hr.

Rasheed et al. [48] also presented a transfer learning-based approach to predict the gender from the masked faces. The approach involves using different Pre–Trained Models namely, DenseNet121, DenseNet169, ResNet50, ResNet101, Xception, Inception V3, MobileNet V2, EfficientNet B0, and VGG16. These models are fine-tuned by adding a SoftMax prediction layer to the base model. The research proceeded in two parts by training the models with both masked, and non-masked faces, and by training the models with the masked faces only. For the first experiment, Inception V3 outperformed the others, while for the other experiment, EfficientNet B0 produced the best accuracy. The models were also tested on the unseen images for which DenseNet121, Xception, EfficientNet B0, and Inception V3 performed the best.

10.2 Age Identification Using Periocular Region

Age Identification is a multi-class classification problem and can be perceived in two ways: identifying the age group (for example, age group 25 to 35, age group 35 to 45, etc.) and identifying the age category (for example, young, middle, old), to which a person belongs. Kamarajugadda and Polipalli [49] focused on identifying the age group for a subject from the periocular region (containing both eyes). A handcrafted feature descriptor, SURF was used to locate and draw out features from the periocular area. The researcher proposed a hybrid classification model based on SVM and KNN. The extracted features from SURF were passed as input to the SVM model that was used to extract the optimized dataset. The images of that were then classified using the KNN classifier. The accuracy of the proposed model was compared with individual models (SVM and KNN). The proposed model achieved the lowest error rate and highest accuracy in the least training time.

11 Emotion Recognition Using Periocular Region

Emotion Recognition is a process that is used to infer and recognize the different mental states, i.e., emotions of the person. The emotions can be categorized as happiness, sadness, anger, disgust, fear, and surprise [50]. The eyes of a person strongly reflect emotions. So, focusing only on the periocular area to recognize the emotions

of a subject can prove to improve the results of the existing systems. There are a few researchers in recent years (after 2018) who have used the periocular area for classifying the emotions of a subject. Reddy and Derakhshani [19] proposed a cross-dataset analysis where the FACES dataset had been utilized for training the model, and the CFD dataset had been utilized to test and examine the performance of the model. These databases contain facial images. The first step of the research was to get the facial landmark coordinates through the Dlib algorithm. Using the eye landmark coordinates, the periocular region was extracted such that both the eyes lie at 75% from the top, and the center of the eyes was at 25% from the slides. The approach was validated using the proposed transfer learning-based MobileNet-V2 model, and the traditional model which used HOG for feature extraction and SVM for identification. The fine-tuned MobileNet-based architecture outperformed the traditional model with a significant accuracy rate.

Alonso-Fernandez et al. [18] on the other hand analyzed the performance of different handcrafted feature descriptors, namely LBP, HOG, Gabor, GLCM, and GIST for emotion identification through the periocular portion. The researcher used CK+ to extract the small (without forehead with some part of eyebrows) and the large (including some part of the forehead) periocular region. Each periocular region was fragmented into non-overlapping blocks for feature extraction. The dataset has video frames that were resized using bicubic interpolation and contrast amplification was limited using the CLAHE algorithm. For each video, the first and the last three frames were considered. Leave-one-subject-out technique was used to create the training and testing datasets. To perform data augmentation, the images in the training datasets were mirrored. SVM was used to classify the emotions and it was concluded that the maximum accuracy was achieved using the combination of all five feature descriptors. Agrawal and Christopher [51] considered the periocular region which itself consists of 12 features out of the 30 facial features for recognizing the expressions of the subject through the video frames of the CK+ Dataset. The dataset has frames with eight different expressions. The actionable units (features) which were available in the corresponding picture along with the intensity of the feature present were stored. Out of the twelve detected periocular features, five features were not contributing to the emotions shown by subjects, so these features were removed. The selected feature set contained only seven identifying periocular features. The research then proceeded by using five different classification algorithms out of which the Random Forest classifier outperformed others. All the images which were expressing happy emotions were classified correctly using the Random Forest classifier.

12 Future Scope

In recent times, there are several pieces of research which were already being conducted using the periocular region, but there are still some possibilities for future researchers, as mentioned below:

- Identifying emotions through the periocular region is still a topic of research as not much work is being done in recent history.
- Using the attention mechanism on the extracted periocular region to enhance the focus on the critical regions.
- Defining a standard procedure for finding the optimal region of interest.

13 Conclusion

The paper provides an extensive review of the systems built using periocular region in recent times. The paper delivers an illustrative view of the literature about the different benchmark datasets, widely used facial landmark extraction algorithms, region of interest extraction along with feature extraction techniques, and classification methods. Using periocular region as the primary trait for biometrics and soft biometrics achieved significantly higher accuracy. Emotion recognition through the periocular region is an emerging field. Periocular region-based research does not demand much user cooperation and can attain the best results without being impacted by unusual constraints.

References

1. Tiong LCO, Lee Y, Teoh ABJ (2019) Periocular recognition in the wild: implementation of RGB-OCLBCP dual-stream CNN. Appl Sci 9(2709):1–17. https://doi.org/10.3390/app9132709
2. Vitek M, Rot P, Štruc V, Peer P (2020) A comprehensive investigation into sclera biometrics: a novel dataset and performance study. Neural Comput Appl 32:17941–17955. https://doi.org/10.1007/s00521-020-04782-1
3. Rot P, Vitek M, Grm K, Emeršič Ž, Peer P, Štruc V (2020) Deep sclera segmentation and recognition. In: Handbook of vascular biometrics. Advances in computer vision and pattern recognition. Springer, Cham, pp 395–432. https://doi.org/10.1007/978-3-030-27731-4_13
4. Vitek M et al (2020) SSBC 2020: sclera segmentation benchmarking competition in the mobile environment. In: 2020 IEEE international joint conference on biometrics (IJCB). IEEE, pp 1–10. https://doi.org/10.1109/IJCB48548.2020.9304881
5. Labati RD, Genovese A, Piuri V, Scotti F, Vishwakarma S (2021) I-SOCIAL-DB: a labeled database of images collected from websites and social media for Iris recognition. Image Vis Comput 105:104058. https://doi.org/10.1016/j.imavis.2020.104058
6. NIST, Face and Ocular Challenge Series Database. Retrieved from http://www.nist.gov/itl/iad/ig/mbgc.cfm
7. Sequeira A et al (2016) Cross-eyed–cross-spectral iris/periocular recognition database and competition. In: 2016 international conference of the biometrics special interest group (BIOSIG). IEEE, pp 1–5. https://doi.org/10.1109/BIOSIG.2016.7736915
8. Santos G, Grancho E, Bernardo MV, Fiadeiro PT (2015) Fusing iris and periocular information for cross-sensor recognition. Pattern Recogn Lett 57:52–59. https://doi.org/10.1016/j.patrec.2014.09.012
9. Sharma A, Verma S, Vatsa M, Singh R (2014) On cross spectral periocular recognition. In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 5007–5011. https://doi.org/10.1109/ICIP.2014.7026014

10. Padole CN, Proenca H (2012) Periocular recognition: analysis of performance degradation factors. In: 2012 5th IAPR international conference on biometrics (ICB). IEEE, pp 439–445. https://doi.org/10.1109/ICB.2012.6199790
11. Zanlorensi LA, Laroca R, Lucio DR, Santos LR, Britto AS, Menotti D (2022) A new periocular dataset collected by mobile devices in unconstrained scenarios. Sci Rep 12(17989):1–18. https://doi.org/10.1038/s41598-022-22811-y
12. Zanlorensi LA, Proença H, Menotti D (2020) Unconstrained periocular recognition: using generative deep learning frameworks for attribute normalization. In: 2020 IEEE international conference on image processing (ICIP). IEEE, pp 1361–1365. https://doi.org/10.1109/ICIP40778.2020.9191251
13. Nguyen H (Mark), Reddy N, Rattani A, Derakhshani R (2021) VISOB 2.0–the second international competition on mobile ocular biometric recognition. In: Pattern recognition. ICPR international workshops and challenges. ICPR 2021. Lecture notes in computer science, vol 12668. Springer, Cham, pp 200–208. https://doi.org/10.1007/978-3-030-68793-9_14
14. Chinese Academy of Sciences Institute of Automation (CASIA) (2010) CASIA iris image database, vol 5. Retrieved from http://biometrics.idealtest.org/
15. Ipe VM, Thomas T (2020) Periocular recognition under unconstrained conditions using CNN-based super-resolution. In: Applied soft computing and communication networks. ACN 2019. Lecture notes in networks and systems, vol 125. Springer, Singapore, pp 235–246. https://doi.org/10.1007/978-981-15-3852-0_15
16. Zhou Y (2021) Evaluation of biometric recognition in the COVID-19 period. In: 2021 2nd international conference on computing and data science (CDS). IEEE, pp 243–248. https://doi.org/10.1109/CDS52072.2021.00049
17. Hernandez-Diaz K, Alonso-Fernandez F, Bigun J (2018) Periocular recognition using CNN features off-the-shelf. In: 2018 international conference of the biometrics special interest group (BIOSIG). IEEE, pp 1–5. https://doi.org/10.23919/BIOSIG.2018.8553348
18. Alonso-Fernandez F, Bigun J, Englund C (2018) Expression recognition using the periocular region: a feasibility study. In: 2018 14th international conference on signal-image technology and internet-based systems (SITIS). IEEE, pp 536–541. https://doi.org/10.1109/SITIS.2018.00087
19. Reddy N, Derakhshani R (2020) Emotion detection using periocular region: a cross-dataset study. In: 2020 international joint conference on neural networks (IJCNN). IEEE, pp 1–6. https://doi.org/10.1109/IJCNN48605.2020.9207542
20. King DE (2009) Dlib-ml: a machine learning toolkit. J Mach Learn Res 10:1755–1758
21. Kazemi V, Sullivan J (2014) One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1867–1874
22. Ren S, Cao X, Wei Y, Sun J (2014) Face alignment at 3000 FPS via regressing local binary features. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1685–1692
23. Solutions–mediapipe. Retrieved from https://google.github.io/mediapipe/solutions/solutions.html
24. Kartynnik Y, Ablavatski A, Grishchenko I, Grundmann M (2019) Real-time facial surface geometry from monocular video on mobile GPUs. arXiv:1907.06724 [cs.CV], arXiv:1907.06724v1 [cs.CV], https://doi.org/10.48550/arXiv.1907.06724
25. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503. https://doi.org/10.1109/LSP.2016.2603342
26. Kumari P, Seeja KR (2021) An optimal feature enriched region of interest (ROI) extraction for periocular biometric system. Multimed Tools Appl 80:33573–33591. https://doi.org/10.1007/S11042-021-11402-0/FIGURES/16
27. Merkow J, Jou B, Savvides M (2010) An exploration of gender identification using only the periocular region. In: 2010 Fourth IEEE international conference on biometrics: theory, applications and systems (BTAS). IEEE, pp 1–5. https://doi.org/10.1109/BTAS.2010.5634509

28. Manyala A, Cholakkal H, Anand V, Kanhangad V, Rajan D (2019) CNN-based gender classification in near-infrared periocular images. Pattern Anal Appl 22:1493–1504. https://doi.org/10.1007/s10044-018-0722-3
29. Kumar KK, Pavani M (2019) Periocular region-based biometric identification using local binary pattern and its variants. In: Information and communication technology for competitive strategies. Lecture notes in networks and systems, vol 40. Springer, Singapore, pp 581–590. https://doi.org/10.1007/978-981-13-0586-3_57
30. Kumar KK, Rao PT (2018) Periocular region based biometric identification using the local descriptors. In: Intelligent computing and information and communication. Advances in intelligent systems and computing, vol 673. Springer, Singapore, pp 341–351. https://doi.org/10.1007/978-981-10-7245-1_34
31. Oishi S, Shirakawa Y, Ichino M, Yoshiura H (2018) Personal identification by integrating a number of features from iris and periocular region using AdaBoost. J Inf Process 26:518–528. https://doi.org/10.2197/ipsjjip.26.518
32. Nakamura T, Yoshiura H, Ichino M (2021) Periocular recognition based on features from thermal and visible-light images. In: 2021 IEEE 10th global conference on consumer electronics (GCCE). IEEE, pp 980–981. https://doi.org/10.1109/GCCE53005.2021.9621934
33. Kumari P, Seeja KR (2021) A novel periocular biometrics solution for authentication during Covid-19 pandemic situation. J Ambient Intell Human Comput 12:10321–10337. https://doi.org/10.1007/S12652-020-02814-1/TABLES/10
34. Kamarajugadda KK, Movva P (2019) Periocular region based biometric identification using SIFT and SURF key point descriptors. In: 2019 IEEE 10th annual information technology, electronics and mobile communication conference (IEMCON). IEEE, pp 968–972. https://doi.org/10.1109/IEMCON.2019.8936271
35. Sarmah AJ, Dutta GN, Lahkar D, Talukdar N, Lahkar B (2022) Biometric authentication-person identification using iris recognition. Int J Innovative Sci Res Technol 7(5):1100–1104. Retrieved from www.ijisrt.com
36. Ismail S, Ali FHM, Aljunid SA (2022) The performance of deep neural networks in deformed iris recognition system. J Positive Sch Psychol 6(3):8517–8529. Retrieved from http://journalppw.com
37. Ipe VM, Thomas T (2020) CNN based periocular recognition using multispectral images. In: Advances in signal processing and intelligent recognition systems. SIRS 2019. Communications in computer and information science, vol 1209. Springer, Singapore, pp 94–105. https://doi.org/10.1007/978-981-15-4828-4_9
38. Ripon KSN, Ali LE, Siddique N, Ma J (2019) Convolutional neural network based eye recognition from distantly acquired face images for human identification. In: 2019 international joint conference on neural networks (IJCNN). IEEE, pp 1–8. https://doi.org/10.1109/IJCNN.2019.8852190
39. Umer S, Sardar A, Dhara BC, Rout RK, Pandey HM (2020) Person identification using fusion of iris and periocular deep features. Neural Netw 122:407–419. https://doi.org/10.1016/j.neunet.2019.11.009
40. Abhinav K, Aneesh R, James P, Varghese A (2021) Attendance marking system using periocular recognition with temperature monitoring (ASPR). In: 2021 9th international conference on reliability, infocom technologies and optimization (trends and future directions) (ICRITO). IEEE, pp 1–5. https://doi.org/10.1109/ICRITO51393.2021.9596524
41. Kumar G, Zaveri MA, Bakshi S, Sa PK (2022) Who is behind the mask: periocular biometrics when face recognition fails. In: 2022 second international conference on power, control and computing technologies (ICPC2T). IEEE, pp 1–6. https://doi.org/10.1109/ICPC2T53885.2022.9777027
42. Mon CS, Hussin AAA, Sin TK (2020) Analyzing the periocular biometric-based access control systems. J Phys Conf Ser 1529(032024):1–8. https://doi.org/10.1088/1742-6596/1529/3/032024
43. Zhao Z, Kumar A (2018) Improving periocular recognition by explicit attention to critical regions in deep neural network. IEEE Trans Inf Forensics Secur 13(12):2937–2952. https://doi.org/10.1109/TIFS.2018.2833018

44. Kayande V, Kaware P, Khambete A, Kulkarni S (2021) Periocular recognition using CNN based feature extraction and classification. In: 2021 10th IEEE international conference on communication systems and network technologies (CSNT). IEEE, pp 393–400. https://doi.org/10.1109/CSNT51715.2021.9509734
45. Tapia JE, Perez CA, Bowyer KW (2015) Gender classification from iris images using fusion of uniform local binary patterns. In: Computer vision–ECCV 2014 workshops. ECCV 2014. Lecture notes in computer science, vol 8926. Springer, Cham, pp 751–763. https://doi.org/10.1007/978-3-319-16181-5_57/COVER
46. Viedma I, Tapia J, Iturriaga A, Busch C (2019) Relevant features for gender classification in NIR periocular images. IET Biometrics 8(5):340–350. https://doi.org/10.1049/iet-bmt.2018.5233
47. Tapia J, Aravena CC (2018) Gender classification from periocular NIR images using fusion of CNNs models. In: 2018 IEEE 4th international conference on identity, security, and behavior analysis (ISBA). IEEE, pp 1–6. https://doi.org/10.1109/ISBA.2018.8311465
48. Rasheed J, Waziry S, Alsubai S, Abu-Mahfouz AM (2022) An intelligent gender classification system in the era of pandemic chaos with veiled faces. Processes 10(1427):1–15. https://doi.org/10.3390/pr10071427
49. Kamarajugadda KK, Polipalli TR (2019) Extract features from periocular region to identify the age using machine learning algorithms. J Med Syst 43(196):1–15. https://doi.org/10.1007/s10916-019-1335-0
50. Ekman P, Friesen WV (1971) Constants across cultures in the face and emotion. J Pers Soc Psychol 17(2):124–129. https://doi.org/10.1037/h0030377
51. Agrawal E, Christopher J (2020) Emotion recognition from periocular features. In: Machine learning, image processing, network security and data sciences. MIND 2020. Communications in computer and information science, vol 1240. Springer, Singapore, pp 194–208. https://doi.org/10.1007/978-981-15-6315-7_16/COVER

Salt Segment Identification in Seismic Images Using UNet with ResNet

P. Venkata Uday Kiran, G. Anuradha, L. Sai Manohar, and D. Kirthan

Abstract Salt segmentation is the process of identifying whether a subsurface target is a salt or not. There are several places on Earth where there are significant amounts of salt as well as oil and gas. For businesses engaged in oil and gas development, finding the exact locations of significant salt deposits is crucial. Also, lands that have been impacted by salt are not useful for farming. The absorption capacity of the plant reduces due to the presence of salt in the soil solution, which results in a reduction in growth rate. So, to identify the land that contains salt, salt segmentation is being done. The seismic image of a particular pixel is analyzed to classify it either as salt or sediment. TGS Salt Identification Challenge dataset is used which consists of 4,000 seismic image patches of size ($101 \times 101 \times 3$) and corresponding segmentation masks of size ($101 \times 101 \times 1$) in the training set. 18,000 seismic image patches are present in the test set which are used for the evaluation of the model. The model used is a combination of UNet with ResNet-18 and ResNet-34. Using this architecture, the salt region can be determined from the seismic data and display the value of the entire salt region. IoU is used as a performance metric to evaluate the model. The outcomes demonstrate that the ensemble model outperforms individual network models and achieves better segmentation results.

Keywords Semantic segmentation · Seismic images · UNet · ResNet · IoU · Deep learning

P. V. U. Kiran (✉) · G. Anuradha · L. S. Manohar · D. Kirthan
V R Siddhartha Engineering College, Vijayawada, India
e-mail: pvudaykiran6197@gmail.com

G. Anuradha
e-mail: ganuradha@vrsiddhartha.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_36

1 Introduction

The depiction of subsurface structures is made possible by seismic imaging. The seismic imaging process uses sound waves that are emitted and reflected off subsurface materials that are detected on the surface by receivers known as geophones. The sound signal that is being reflected is collected to create a 3D image of the subsurface rock structure in the later stages of processing. The borders of several rock species are visible in seismic pictures. According to theory, the strength of the rejected signal is thought to be significantly correlated with the physical differences between the rocks at the area of contact. This effectively implies that, while seismic pictures provide nothing about the actual rocks, they do provide information about the borders between rocky deposits. Identification of salt deposits is crucial because seismic pictures are utilized in the search for hydrocarbon fuel sources by assisting in the discovery of possible reservoir rocks.

Agriculture and oil drilling requires the automated and precise determination of whether a subsurface target is a salt. Unfortunately, it may be quite challenging to determine the exact location of significant salt deposits. Expert interpretation of salt bodies is still required for professional seismic imaging. So, identifying the exact location of salt deposits is necessary for semantic segmentation.

In this paper, two encoders are used for identifying salt deposits. One of them is a ResNet 18 and the other is a ResNet 34. The images from the dataset are taken and preprocessed using the data augmentation technique. To enhance the quality of the data used to train artificial neural networks, a deep learning technique called data augmentation is used. By introducing variations to current data samples, the training dataset is artificially enlarged. The preprocessed images are then given to a deep learning model called U-Net which uses ResNet-18 and ResNet-34 as encoders. The dice loss function is then used to improve the IoU score. It is also used to figure out how closely the predicted image resembles the original image. A publicly available dataset called as TGS dataset is used to train these two models. At last, IoU is calculated to analyze the performance of the two models.

The remainder of the paper is organized as follows. Related works for this are given in Sect. 2. In Sect. 3, we introduce the workflow and the methodology of the proposed UNet with ResNet18 and ResNet34 Architecture for relevant to identify salt segments in seismic images. In Sect. 4, the IoU percentage is presented and detection results are obtained to assess the performance of the proposed approach and also we have shown how our model identifies the salt in the given seismic image. Conclusions and policy implications are given in Sect. 5. Lastly, Sect. 6 contains the future scope.

2 Related Works

In this paper, an enhanced UNet deep network is used for the identification of faults and salt domes. Further, the encoders of VGG19 and ResNet34 are used for transfer learning to improve the performance of UNet and before employing the fused UNet on seismic data, the networks are initially trained on real-world images (ImageNet). At last, ResNet skip connections are used to create a model which is unaffected by the similarity between seismic data noise discontinuities. Netherlands offshore F3 block, LANDMASS, and TGS datasets are used for the evaluation of the model performance by using the performance metrics like Precision, Recall, F1-score, and IoU. The pre-trained ResNet34 fails to detect salt boundaries in zoomed regions. The disadvantage of this model is the pre-trained ResNet34 fails to detect salt boundaries in zoomed regions [1].

A UNet-based deep convolutional neural network is used for the semantic segmentation of satellite images. There are two subnetworks in the proposed TL-DenseUNet. To obtain multilevel semantic information, the encoder subnetwork employs a transferring 4 DenseNet while the decoder subnetwork uses dense connections to combine the information in each layer. Various metrics like F1 score, recall, precision, overall accuracy (OA), IoU, MIoU, and kappa coefficient and are used to assess the performance of TL-DenseUNet. TL-DenseUNet kappa coefficient is enhanced by more than 0.0752 when compared with several other cutting-edge models. The model lacks in the identification of ground features with comparable spectra that were liable to classification. The drawback of this model is that it lacks the identification of ground features with comparable spectra that were liable to classification errors [2]

U-Net plus Se-ResNet based deep learning method is used for the identification of salt segments in seismic images. After that, the model is trained using Lovasz Loss after initially employing a combination of dice loss and binary cross entropy (BCE). TGS dataset is used for the evaluation of the model which contains images of 101×101 pixels. k-fold cross-validation was used for the evaluation of the model. IoU was used as the evaluation metric for this segmentation issue. An average IoU metric score of 0.9037, 0.84201, 0.866771, and 0.819871 is achieved on Lovasz-Train, Lovasz-Valid, BCE-Train, and BCE-Valid respectively. The disadvantages are it needs to have directly assisted the process of seismic data interpretation without the compulsion of providing seismic features [3].

A deep-supervised network architecture called UNet with an encoder and decoder that combine feature maps with associated resolutions via jump connections is used for salt 5 segmentation. The training is done in two stages in which the binary cross-entropy loss is used in the first 10 epochs to improve the learning rate and the Lovasz loss function for the next 100 epochs to optimize the IoU metric. The proposed method has achieved 87.39. Due to the lack of data, the encoder uses ImageNet pre-trained weights to achieve better results and to reduce training time. The limitation of this project is an insufficient amount of data [4].

A deep CNN-based model that is combined with human interactions is proposed for the interactive salt segmentation of seismic images. Negative and positive points

are converted into two Euclidean distances maps (EDMs) which are then paired with the seismic images to train the CNN model and included the interaction points in the proposed technique. A combination of U-net and PPM make up the model which was developed using the data from the TGS dataset. The likelihood maps generated by the CNN model are then improved using a graph cut technique, which causes to update of the salt boundaries. The saltISCG technique provides two mean average precisions (mAPs) of 91.59. The disadvantage of this method is that it is not a fully automatic method, and requires adding some interaction points to the complex seismic images [5].

In this paper, the author develops a highly generalized fully convolutional Dense Net with the help of self—an attention mechanism for automatic salt segmentation. This framework's robustness is measured by testing the proposed framework with the data set. The author uses TGS salt segmentation data set and a 3-D SEAM dataset. The author suggested 6 network architectures and used architectures of convolutional blocks, denser blocks, squeeze and excitation blocks, and transition-down blocks. To evaluate the performance of the model performance metrics like Accuracy, Precision, F1-score, recall, and IoU are taken into consideration. The limitation of this system is the suggested DL framework's potential to produce noticeably larger errors towards the edges/pinch-outs of salt bodies is one of its disadvantages [4].

To describe the intricate distinctions between the true and modelled multiples in a nonlinear relationship, the author of this study introduces U-net, a well-known deep learning technique. This framework's robustness is measured by testing the proposed framework with the Sigsbee2B data set. The architecture of the U-Net is described and the traditional LR method is reviewed. To compare the results of various adaptive subtraction strategies objectively, they define the signal-to-noise ratio (SNR). The proposed U-net method's primary drawback is its expensive computational requirements [6].

In this, the author offers a thorough analysis of the recent literature on visual attention models, recurrent networks, and generative models in conflictual circumstances. The architecture of the DEEP NEURAL NETWORK and DL-BASED IMAGE SEGMENTATION MODELS are described. The data sets are divided into 3 types for DL-based picture segmentation: 2D (pixel) images, 2.5D RGB-D (colour + depth) images, and 3D (voxel) 7 images. The datasets KITTI and Cam Vid are among the most widely used in computer vision, together with PASCAL Visual Object Classes (VOC). Pixel accuracy, Mean Pixel Accuracy (MPA), Intersection over Union (IoU), and Dice coefficient are the measures for the image segmentation models. The limitation of this model is it performs only on a subset of object classes [7].

3 Methodology

See Fig. 1.

In our project, we are considering the problem of salt identification as a semantic segmentation problem in which input images are classified pixel-by-pixel into salt and non-salt regions. In the corresponding mask against an image, salt regions are represented by white pixels while non-salt regions are in black as shown in Fig. 2.

According to our proposed system in Fig. 1, seismic images from the dataset were augmented and batched using a data loader, these batched images are fed into UNet with the ResNet model for training, and finally predicted mask is obtained using the trained model.

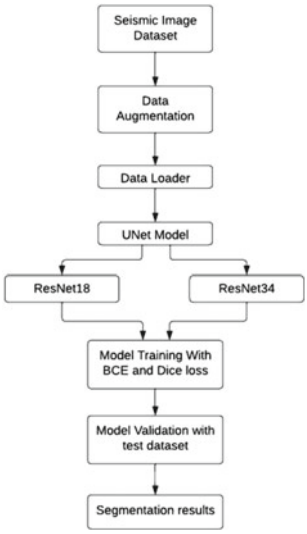


Fig. 1 Proposed system diagram

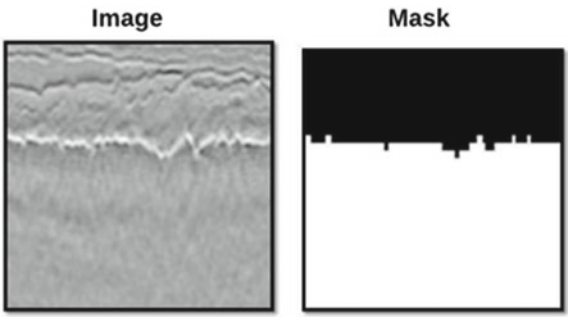


Fig. 2 Sample image and its corresponding mask

3.1 Dataset Description

TGS salt identification challenge dataset is used for model evaluation which consists of 4000 labelled images in the training dataset [8]. The training dataset also consists of the corresponding masks of the seismic image. The training dataset consists of seismic images which are of the size $101 \times 101 \times 3$ and mask images of the size $101 \times 101 \times 1$. There are also some anomalies in the dataset where there are some images that are black in colour and don't contain any corresponding mask. There is a need for data augmentation. The dataset also contains the depth information of each image. The dataset is split into training, testing, and validation with 3420, 200, and 380 images respectively.

3.2 Data Augmentation

The data present in the training dataset is very less. So, there is a need for augmentation. Data augmentation is the process used to add more amount of data to the dataset by using the images that are already available in the current dataset. This can be done by making small changes to the image using different techniques like flipping, geometric transformations [9], cropping, rotating, noise injection, etc. The geometric transformations technique is used in the project to increase the dataset size. But, there is no built-in image and mask data augmentation in PyTorch. Therefore, if you transform the image at random, the image and mask cannot match. So, data augmentation is aided by a python library called albumentations which is used for image augmentation [10]. After importing albumentations library transformations have been done which include Optical Distortion, Transpose, Horizontal Flip, Vertical Flip (change X and Y axis), and Random Rotate 90. After augmentation, the resulting images are shown in Fig. 3.

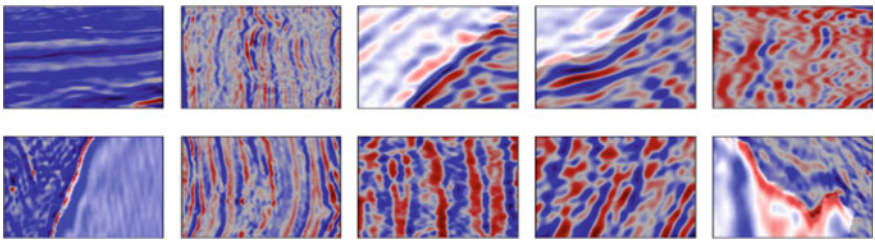


Fig. 3 Augmented data

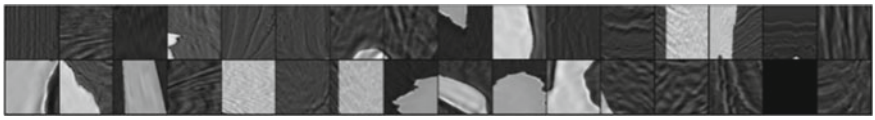


Fig. 4 Data loader

3.3 Data Loader

PyTorch provides a data primitive called torch.utils.data.DataLoader. The pre-loaded datasets can be used by using this data loader [11]. Data loader offers the options such as batch sampler, batch size, drop last, and collate function to automatically combine individual fetched sequence data into batches. The data loader feeds the training data in batches to the model. A batch size of 32 is used in this project and after shuffling the augmented data it looks as shown in Fig. 4.

3.4 UNet with ResNet-18 and ResNet-34

In our project, salt segmentation is done by using UNet in combination with ResNet18 and ResNet34. UNet with ResNet18 and ResNet34 involve using ResNet18 and ResNet34 as encoders. A convolutional neural network architecture called a U-Net is employed for segmentation tasks. U-Nets are particularly beneficial for applications that need high spatial accuracy and to be the same size as the input. Because of this, they excel at producing segmentation masks and performing image processing.

When CNNs are used to classify images, the image is typically captured and down sampled using a series of stride two convolutions into one or more categories, each time shrinking the grid size. To produce an image that has the size same as the input or may be greater, there is a need for an up sampling method that expands grid size. As a result, the network topology seems to be in the form of a U-like structure. The up sampling/decoder path constitutes the right side of the UNet architecture and the down sampling/encoder path constitutes the left side of the UNet (see Fig. 5).

The prediction results lack in detail when we use only U-Net architecture. To overcome this problem, skip connections are added between network blocks. The original pixels are merged with the resulting Res block through a skip link, allowing the final computation to take place while being conscious of the initial pixels inputted into the model. As a result, the minuscule characteristics of the input image are all placed at the top of the U-Net which maps to the output from the input almost instantly. Since the outputs are combined, dense blocks are more similar than Res

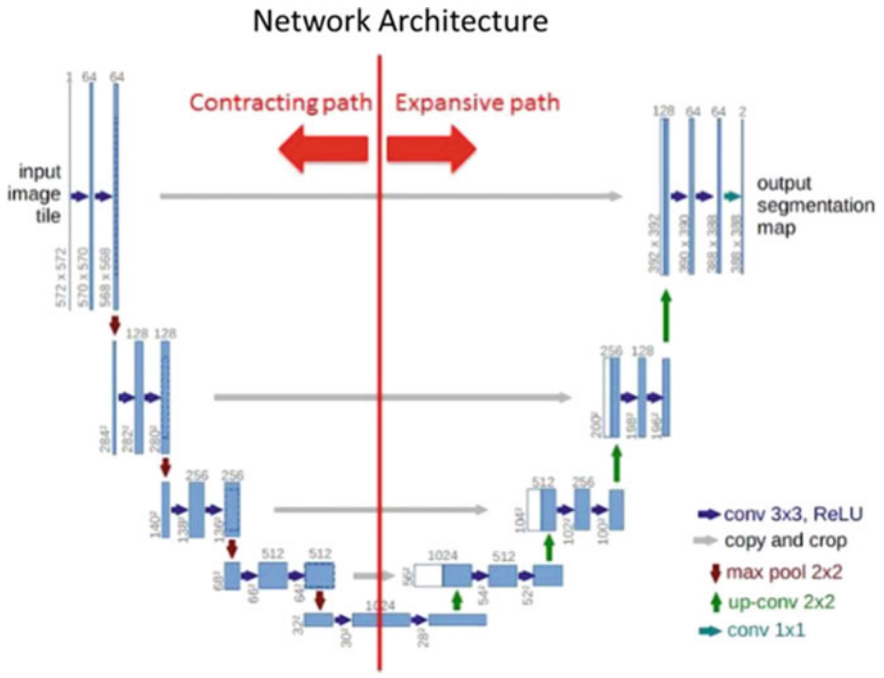


Fig. 5 UNet architecture

blocks to U-Net blocks. However, two more convolutions reduce the grid size once more, which also helps avoid consuming too much memory.

ResNet Encoder

The encoder/down sampling portion of the U-Net can utilize a ResNet (see Fig. 6.) (the left half of the U). In my models, I have used ResNet-18 and ResNet-34 encoders. ResNet-18 has 18-layer architecture and ResNet-34 has 34-layer architecture. An 18-layer and 34-layer architecture of ResNet is faster to train the model and efficiently utilizes memory.

Decoder

The U-Net will automatically build the decoder portion of the architecture when an encoder is provided. A feature vector of high dimension created by the encoder is used by the decoder to create a semantic segmentation mask [12]. A transpose convolution of size 2×2 is used by the decoder at the beginning. Then the relevant feature map with necessary skip connections in the encoder is concatenated with it. Due to the network's depth, the skip connections offer functionality from prior levels that are occasionally lost. Then, two 3×3 convolutions are employed, with a ReLU activation function coming after each convolution. The final decoder's output is subjected to a 1×1 convolution by using sigmoid activation. This function provides the segmentation mask for each pixel classification.

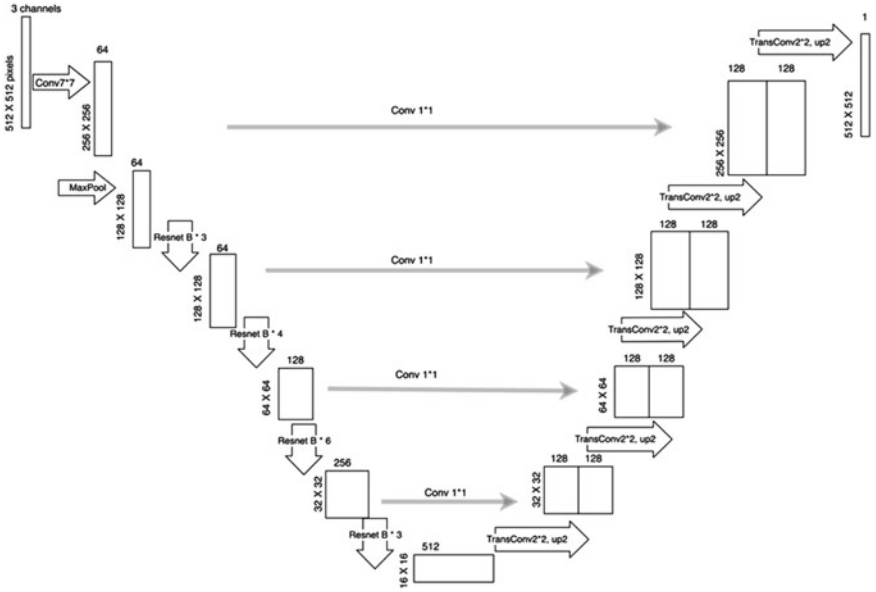


Fig. 6 UNet with ResNet

3.5 Model Training

Both models UNet with ResNet 18 and ResNet 34 were trained for 100 epochs each, where the first 50 epochs used BCE loss and the next 50 epochs used Dice loss functions with a learning rate of 0.0001.

3.6 Loss Functions

In the model, BCE and dice loss functions are used. As we need to categorize each image pixel-wise to determine whether it contains salt or not, it is considered a binary classification issue [13]. BCE is employed for binary classification problems. When employing the BCE loss, just a single output node is required to divide the information into two categories. But BCE loss does not take image level forecasting into account since it only microscopically analyzes loss rather than considering it globally. So, we have used another loss function called Dice loss. The dice coefficient, which gauges the degree of data instance overlap, serves as the foundation for dice loss functions. Dice loss is typically used to determine how similar two samples are. The main purpose of segmentation is to improve IoU or Dice coefficient metric [14]. This is the reason why these metrics are directly employed as the loss function.

$$\text{BCE loss} = -\left[\text{target} * \log(\text{our\_ value}) + (1 - \text{target}) * \log(1 - \text{our\_ value})\right]$$

$$\text{Dice loss} = \frac{2|A \cap B|}{|A| + |B|}$$

3.7 Model Evaluation Criteria

Depending on the issue and the design of the model, CNN models are assessed differently. Most often, accuracy, precision, and recall are utilized as measurement criteria. We utilized IoU as the evaluation measure for the segmentation challenge. IoU is the primary assessment criterion for assessing the excellence of the produced masks. For binary segmentation, the average IoU is calculated by considering each class IoU For this process, both the actual shape features from the training set and the ground truth features from the training model are needed. It only divides the combined cumulative area as given in the equation by the overlap area between these two bounding boxes. A reasonable forecast is often regarded as having an IoU value of more than 0.5.

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}}$$

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

4 Results

Each image in the dataset is preprocessed by normalizing the image and then the image is cropped to a size of 128 × 128. Then, we train two DCNN architectures on the input data. In the model, the first UNet network uses a ResNet18 encoder, second uses ResNet34. Each network in the model is trained for 50 epochs using the BCE loss function and Dice loss is used for the next 50 epochs. The training and validation loss and IoU for ResNet34 are in Figs. 7 and 8. At last, the weighted average ensemble technique is used to improve the prediction IoU value. The IoU values at the 50th epoch are listed as shown in Table 1.

After ensembling all models the segmentation can be done by using the images taken from the test set. The yellow region in the Fig. 9 shows the salt region.

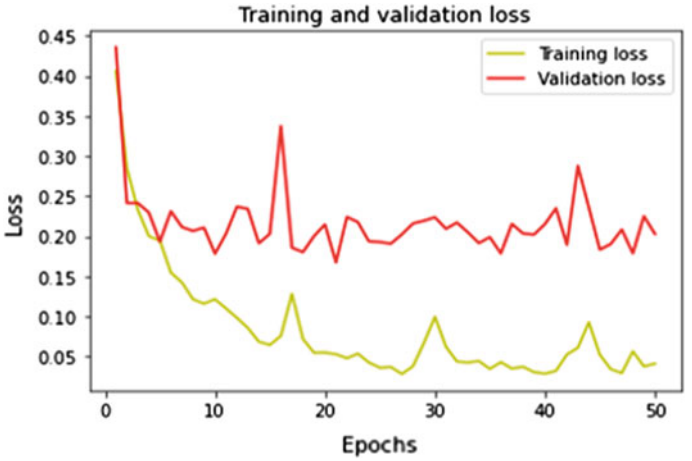


Fig. 7 Training and validation loss for UNet + ResNet34 model

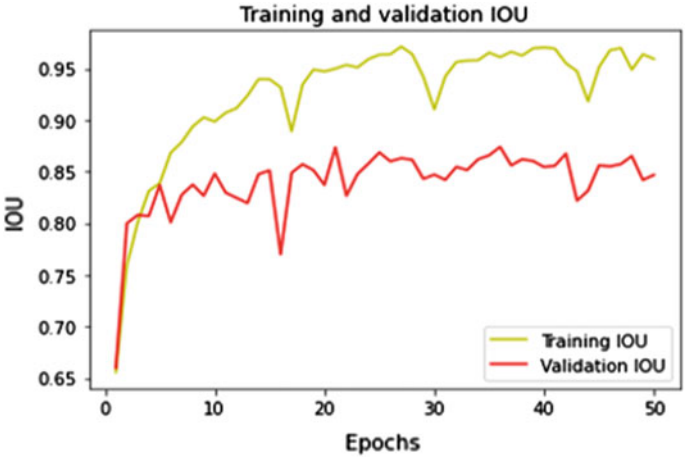


Fig. 8 Training and validation IoU for UNet + ResNet34 model

Table 1 IoU for model networks

Models	Loss function	Train IoU	Validation IoU
UNet + ResNet18	BCE Loss	0.87	0.79
	Dice loss	0.89	0.817
UNet + ResNet34	BCE loss	0.87	0.816
	Dice loss	0.898	0.827

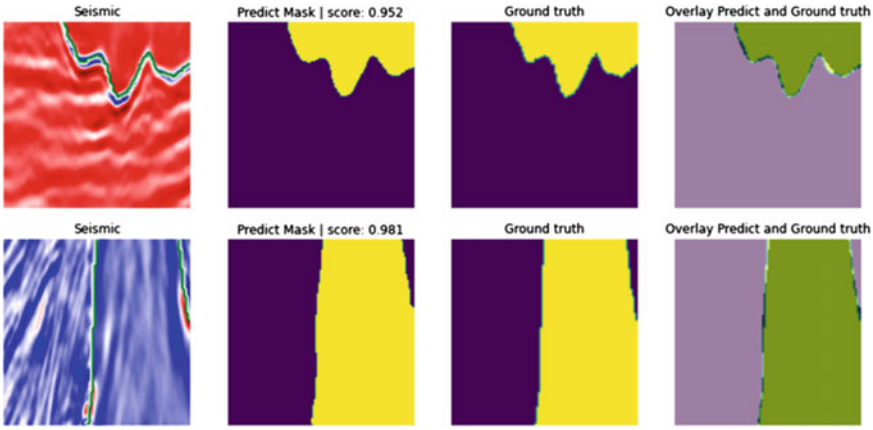


Fig. 9 Prediction on a test image

5 Conclusion

In this paper, models have been suggested for the identification of a salt region in seismic images. We have used UNet architecture by making ResNet-18 and ResNet-34 as encoders in the model. These networks are then trained and the BCE loss function is applied for the first 50 epochs and then the Dice loss function is employed for the next 50 epochs. The UNet-ResNet34 network achieves higher IoU when compared to the UNet-ResNet18 network. Then the results are combined by using the weighted average ensemble technique. The proposed model is evaluated by using the TGS salt identification challenge dataset.

6 Future Scope

Several directions can be looked at for future work. The first involves replacing the existing encoder with a pre-trained network and only keeping the decoder portion of the proposed architecture to apply transfer learning to it. The second approach would be to train additional classification models that would determine whether or not a patch includes salt, using the segmentation model conditionally. This would help to enhance salt detection metrics. The third direction would be to evaluate the design performance on different datasets. Since the training dataset is small, pseudo-labelling can be used in future work.

References

1. Alfarhan M, Deriche M, Maalej A (2022) Robust concurrent detection of salt domes and faults in seismic surveys using an improved UNet architecture. IEEE Access 10:39424–39435
2. Zhang H, Zhu P, Liao Z, Li Z (2022) SaltISCG: interactive salt segmentation method based on CNN and graph cut. IEEE Trans Geosci Remote Sens 60:1–14
3. Li Z, Sun N, Gao H, Qin N, Li Z (2021) Adaptive subtraction based on U-Net for removing seismic multiples. IEEE Trans Geosci Remote Sens 59(11):9796–9812
4. Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N, Terzopoulos D (2022) Image segmentation using deep learning: a survey. IEEE Trans Pattern Anal Mach Intell 44(7):3523–3542
5. ul Islam MS (2020) Using deep learning based methods to classify salt bodies in seismic images. J Appl Geophys 178:104054
6. Milosavljevi´c A (2020) Identification of salt deposits on seismic images using deep learning method for semantic segmentation. ISPRS Int J Geo-Inf 9(24):1–16
7. Wang D, Chen G (2021) Seismic stratum segmentation using an encoder-decoder convolutional neural network. Math Geosci 53:1355–1374
8. An Y, Guo J, Ye Q, Childs C, Walsh J, Dong R (2021) Deep convolutional neural network for automatic fault recognition from 3D seismic datasets. Comput Geosci 153:104776
9. Ahmad P, Jin H, Alroobaea R, Qamar S, Zheng R, Alnajjar F, Aboudi F (2021) MH UNet: a multi-scale hierarchical-based architecture for medical image segmentation. IEEE Access 9:148384–148408
10. Abdollahi A, Pradhan B, Alamri AM (2022) An ensemble architecture of deep convolutional Segnet and UNet networks for building semantic segmentation from high-resolution aerial images. Geocarto Int 37(12):3355–3370
11. Cao Y, Liu S, Peng Y, Li J (2020) DenseUNet: densely connected UNet for electron microscopy image segmentation. IET Image Process 14(12):2682–2689
12. Diakogiannis FI, Waldner F, Caccetta P, Wu C (2020) ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data. ISPRS J Photogrammetry Remote Sens 162:94–114
13. Hu G, Hu Z, Liu J, Cheng F, Peng D (2022) Seismic fault interpretation using deep learning-based semantic segmentation method. IEEE Geosci Remote Sens Lett 19:1–5, Art no. 7500905
14. Tran S-T, Nguyen M-H, Dang H-P, Nguyen T-T (2022) Automatic polyp segmentation using modified recurrent residual UNet network. IEEE Access 10:65951–65961

A Study of Comparison Between YOLOv5 and YOLOv7 for Detection of Cracks in Concrete Structures

Ajay Anoop and Jeetu S Babu

Abstract Mechanical failures of man-made structures pose risk to human life and property. Location and mitigation of cracks before it propagates through critical regions of a mechanical/civil structure prevents the occurrence of accidents and loss of life. Crack propagation, also known as sub-critical crack propagation or stress corrosion, frequently happens under low stress and is characterized by gradual propagation. By releasing the elastic strain energy brought on by an external load creates the formation of new surfaces. Cracks spread to reduce the energy of the system. Surface cracks can be found using various non-destructive testing methods: Visual Optical Testing, Eddy Current Testing, Liquid Penetrant Testing, and Magnetic Particle Testing. This study is currently limited to the use of visual testing using computer vision, feature extraction from captured data using multiple image processing algorithms to identify cracks using an object detector model build using data points collected from a user data set. In a systematic manner, we tried to develop object detector models separately using YOLOv5 and YOLOv7 and performed a study on different standard evaluation metrics obtained from the two frameworks. YOLOv5 and YOLOv7 are the two recent additions to the YOLO family. YOLO-based object detector uses deep learning to train models for object detection using pre-trained weights from COCO data set. Results indicate further application of the detector model to be assertive on any physical structures pertaining to risk of surface cracks.

Keywords YOLOv5 · YOLOv7 · Object detection · Computer vision · Deep learning · COCO data set

A. Anoop (✉) · J. S. Babu
Department of Mechanical Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, Vallikavu, India
e-mail: ajayanoop629@gmail.com

J. S. Babu
e-mail: jeetusbabu@am.amrita.edu

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_37

1 Introduction

Technology is evolving in an exponential trend. Modular methods have replaced conventional routes in most industrial plants. It has directly impacted work environments significantly. The enormity of area it serves are diverse, it affects trade and world economy the same way it affects the daily life of common man, such that its significance often goes unnoticed. This paper proposes several soft computing techniques to extract information from data collected using an external hardware such as a camera mounted on a drone. Drones fall into a diverse category of unmanned vehicles and use robotics principles. Robotics is an engineering discipline that amalgamates different fields of engineering to build and develop solutions for problems faced by mankind.

The services provided to mankind are the key aspect for the existence of technology. It supports various work systems, improving and enhancing safety, thereby reducing risk of failure. Early detection of failure in engineered structures helps prevent fatal accidents, often involving human casualty. Risk mitigation in such structures requires precise calculation and planned reinforcement in a time to time basis. To deploy humans in missions, involving risk is a risk on its own due to the human life involved. Monitoring large engineering structures like dams, buildings, and bridges is often a tedious and time-consuming task. Moving across these structures also increases the duration of such missions. Drones, though in this case falls as a far scope of the paper, we plan it in a way such that a drone can be easily installed into the system without having much change in system design.

Visual detection of cracks using photogrammetry is a time proven technique due to high degree of reliability, though the complexity involved in incorporating advanced instruments into a drone architecture is still under development. Cracks or propagation of cracks in civil structures can cause partial or complete structural loss with damage to property and environment [1]. Therefore, it's important to assess the quality and integrity of these structures in a periodic basis using an automated system or a human–robot interaction (HRI) replacing a human–human environment.

A drone-based camera with a micro-controller that is programmed to detect cracks can relay the information back to a base station using telemetric feedback. This will substantially decrease the delay in transmission of useful data to the one that involves human–human operating environment. This decreases the cost thus involved significantly.

However, a drone might obtain false information due to improper calibration of the equipment or malfunction of detection algorithm, i.e. bogus identification due to terrain distance calculation error affecting detection parameters. This has to be dealt with modern powerful photogrammetry devices. There's always a trade-off between economics and equipment. Improved semiconductor-based micro-electronics have made it possible to create small, extremely sensitive, high-resolution cameras that can be installed on drones and be used to capture images with even the smallest level of detail [2]. Machine learning algorithms can be trained to capture images while simultaneously detecting the design parameter [3].

2 Related Works

Using Keras and the VGG-16 model, Gopalakrishnan et al. explore several deep learning techniques for locating cracks and triggering mechanisms to view a crack from big databases like ImageNet [4]. The ImageNet database, which is based on the hierarchical structure of WordNet, which is a well-defined collection of image data set. As a result, the database is extremely well dispersed across categories, and the results are based on extremely exact inputs.

In their discussion and review of several image processing methods for object detection, particularly fracture detection, Arun et al. claim that the majority of crack detection techniques uses camera type images for analysis with superior segmentation techniques like the threshold technique and reconstructable feature extraction methodology for a comprehensive analysis of damage [5].

Pei et al. have proposed a modified ultrasonic method which aimed at enhancing the use of ultrasound technology for fracture depth identification. The outcomes collected have made it possible to offer important information regarding the extent of the damage [6].

Shirahata et al. have put out an approach that uses a non-destructive ultrasonic test for distinguishing fatigue-related cracks. They have created tandem array ultrasonic testing equipment that is capable of identifying insufficient penetration. The study also developed image reconstruction system employing the multi-synthetic aperture focusing technique allowed for the observation of lengthier cracks, its tip closure, and opening [7].

With the help of the sensitivity of the embedded ultrasonic sensors, Wolf et al. have developed a detection method to locate cracks that are spreading inside concrete structures before they become visible on the concrete's surface. Due to the sensor's constant coupling to the medium, they have employed sensitivity as a consideration [8]. Through the use of non-destructive testing techniques like acoustic emission and digital image correlation, the accuracy of the crack's beginning detection was assessed.

A crack detection system with good 3D laser scanning measurement capabilities and high spatial resolution imaging has been proposed by Rabah et al. Three steps—shading correction, fracture identification, and crack mapping—were taken to complete the crack detection and mapping. In a pixel coordinate system, the fracture has been identified. After the definition was finished, the crack was remapped using reverse engineering using the referred coordinate system. This resulted from a conversion from the pixel coordinate system to the terrestrial laser scanner or global positioning system point clouds and the matching camera image [9].

Yao et al. discuss one of the most important issues affecting the performance and lifespan of civil infrastructures like bridges, pipelines, and other structures as the incidence and propagation of cracks [10]. As a result, several fracture detection and characterization approaches have been investigated and created over the past few decades in the fields of non-destructive evaluation and Structural Health Monitoring (SHM) [10].

Dinh et al. propose an automatic peak detection approach for image segmentation that can be used for concrete fracture detection [11]. Scanned images with a crack potential are firstly processed using the line emphasis filter. The processed image's greyscale histogram is then smoothed using moving average filter then examined using the offset distance and crossover index, two dynamic characteristics, to find notable peaks. The effectiveness of this algorithm is evaluated by the authors through testing the data, from high contrast to very noisy background images.

Further, studies were done on various image processing and analysis techniques for object detection. A study suggests the usage of optimized computer vision for the detection of unmarked road edges [12]. Another research focuses on the potential of crack detection in railway tracks using Unmanned Aerial Systems [13]. YOLO was used for the detection of potholes in a work done by Dharneeshkar et al. [14]. An IOT deployed early detection of building collapse was implemented and studied using sensor-based feedback [15]. An image-based study for detection of cracks using transfer learning-based deep convolutional neural network studies different network architectures such as Google net, Alexnet, and Resnet and concluded that Google net is superior to all the above systems in detecting cracks [16]. Various studies on the ensemble training model with multiple detectors of YOLO algorithms yielded impressive evaluation metrics [17].

3 Methodology

This paper consists of acquiring, testing, and validating an input image and classifying it based on the learned and trained model to identify cracks from an image using an object detection algorithm. There's no one specific solution to execute this. In this approach, we follow the route as follows:

– Object detection and image processing
– Using YOLO for object detection.

For sake of simplicity, only wall cracks shot from direct perspective of the photographer were included as the amount of varying classification based on a given parameter would increase the time required in various learning cycles. However, further studies can be extended on such a platform for improved accuracy and precision so that a much more robust and reliable output could be predicted using the same principle.

3.1 Object Detection and Image Processing

Here, we make use of images from the data set created under the name 'cracks' to start the process of smoothing the image using bilateral filter. A bilateral filter is a nonlinear, edge-preserving, and noise-reducing smoothing filter for images. It swaps

out each pixel's intensity for a weighted average of intensity values from adjacent pixels. This weight may have a Gaussian distribution as its foundation.

Once smoothing process is done, we have used canny edge detector algorithm to identify uneven edges, which are characterized by sharp edges. Canny edge detection algorithm works in multi-stages. The image is smoothed in first stage using Gaussian convolution. The image is then smoothed to emphasize areas with high first spatial derivatives using a straightforward 2D first derivative operator. In the image of the gradient magnitude, edges give rise to formation of ridges.

Non-maximal suppression is the algorithm's approach of tracking along top of these ridges and setting zero to all pixels that are not actually on the ridge top in order to produce a thin line in the output. T1 and T2 thresholds govern the hysteresis of the tracking process, with T1 > T2. Tracking only begins at a point on the ridge that is higher than T1. From there, tracking continues in both directions until the height of the ridge drops below T2. Hysteresis like this prevents noisy edges from becoming redundant edge pieces.

Images thus obtained are annotated with the help of a Python-based labelling software called LabelImg [18]. Cracks are labelled and stored as values of the data set.

3.2 Using YOLO for Object Detection

YOLOv5 was released in 2020 and has a large user network. It has various versions updated from time to time like YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Also different unofficial versions were also released like YOLO-R and YOLOv6. YOLO-R was created to carry out several tasks with a single visual representation. YOLO-R object detection uses implicit information from shallow layers and explicit information from the deep layer.

YOLOv7 is an official release and a direct update to the YOLO family from YOLOv5. YOLOv7 is the fastest and most precise real-time object detector to date, according to the Wang et al. [19]. The speed for various models ranges from 5 to 160 FPS. YOLOv7 has a 40% reduction in the number of parameters and a 50% reduction in computation when compared to the base models. The architecture is derived from YOLOv4, Scaled YOLOv4, and YOLO-R.

Based on the Common Objects in Context (COCO) data set, the YOLO family of compound-scaled object identification models was used to train the model in our context. It has basic capability for model ensembling, hyperparameter evolution, Test Time Augmentation (TTA) and export to ONNX, TFLite, and CoreML.

It is developed using Pytorch framework which is an open-source machine learning platform based on torch library. It is extensively used in applications relevant to computer vision. When it comes to performance, in terms of detecting speed and accuracy, YOLO surpasses all other object detectors. In real time, it can process images at a rate of around 155 frames per second (fps), achieving almost twice

the mAP (Mean Average Precision—an evaluation parameter for object detectors) compared to other object detectors like Faster R-CNN.

In the current study, requisite YOLO repository was cloned from GitHub. Here, we've made use of Google Colab which is an easy to use, powerful Python-based IDE. Once the repository is cloned, required dependency packages are installed as per user demand. We've made use of torch library, which is an open-source machine learning library comprising algorithms for deep learning applications. After installation of dependencies, we set path to the directories of training and testing files by mounting the data set on any online cloud storage.

4 Training-Based Learning Using YOLO

For training our model, a data set is created by capturing images using a drone-based camera. A set of 618 images were captured from various locations. Pictures were filtered and sorted for training and validation. Training data set consists of 495 images, and validation data set is of 123 images. It is a small data set which can further be extended by adding more images to the data set. For quick prototyping, a larger data set could prove time-consuming. A typical YOLOv5n arrangement as in our case won't exceed five minutes to train subjected to the extend of data set and computational speed available at Colab environment. There are mainly five training models available in YOLOv5. The least time-consuming and easiest to train is called YOLOv5n. The extend of accuracy increases in a typical hierarchical fashion. The best training model comes with YOLOv5x. We limit the training of our data set till YOLOv5l [20] (Fig. 1).

Annotations and labelling is done using LabelImg application available with python tools. User can identify a characteristic and plot boxes and classify annotations as per investigation requirement. YOLO can identify the position of bounding boxes from a text file so created in YOLO complaint format using LabelImg. Training data is divided into eight batches and 40 epochs. We use pre-trained weights to start training. The batch size represents the number of samples processed prior updating the model. The number of epochs represents the total number of trips over the training data set. A YAML file is created with the path details of training data set. YAML is a human-readable markup language that can be used with any computer language to write configuration files.

In order to train our data set with YOLOv7, we clone the GitHub repository to our active project and then instal torch library and other dependencies so that we have required packages for our model.

In Fig. 2, the graph dictates the trend in detection of parameter from YOLOv5n to YOLOv5l. YOLOv5l retains a metrics/mAP of 0.576. YOLOv7 surpasses all the models with a metrics/mAP of 0.662. This implies that YOLOv7 model has a confidence of 66.2% to categorize a valid input to cracks category by using this model. A more reliable model can be trained as per user discretion upon training with large data sets falling in the same class (Figs. 3 and 4, Table 1).

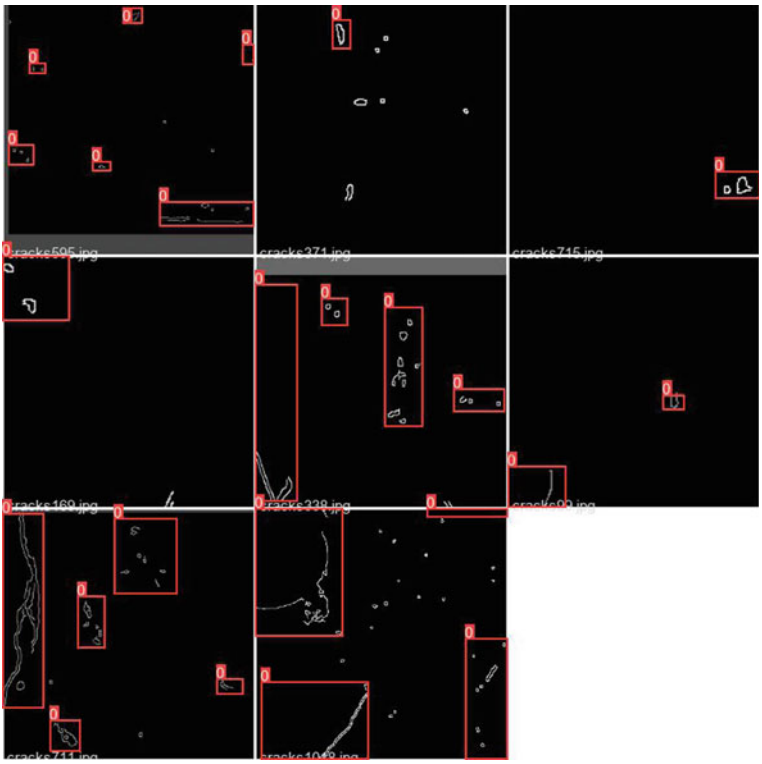


Fig. 1 Feature annotation for cracks

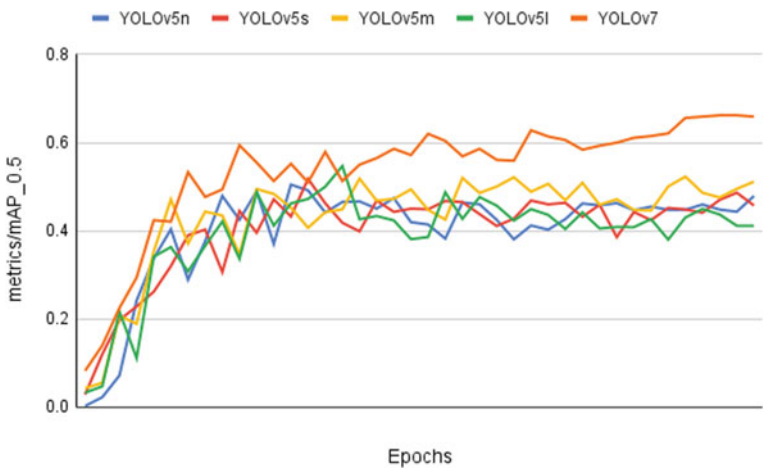


Fig. 2 mAP trend per epoch

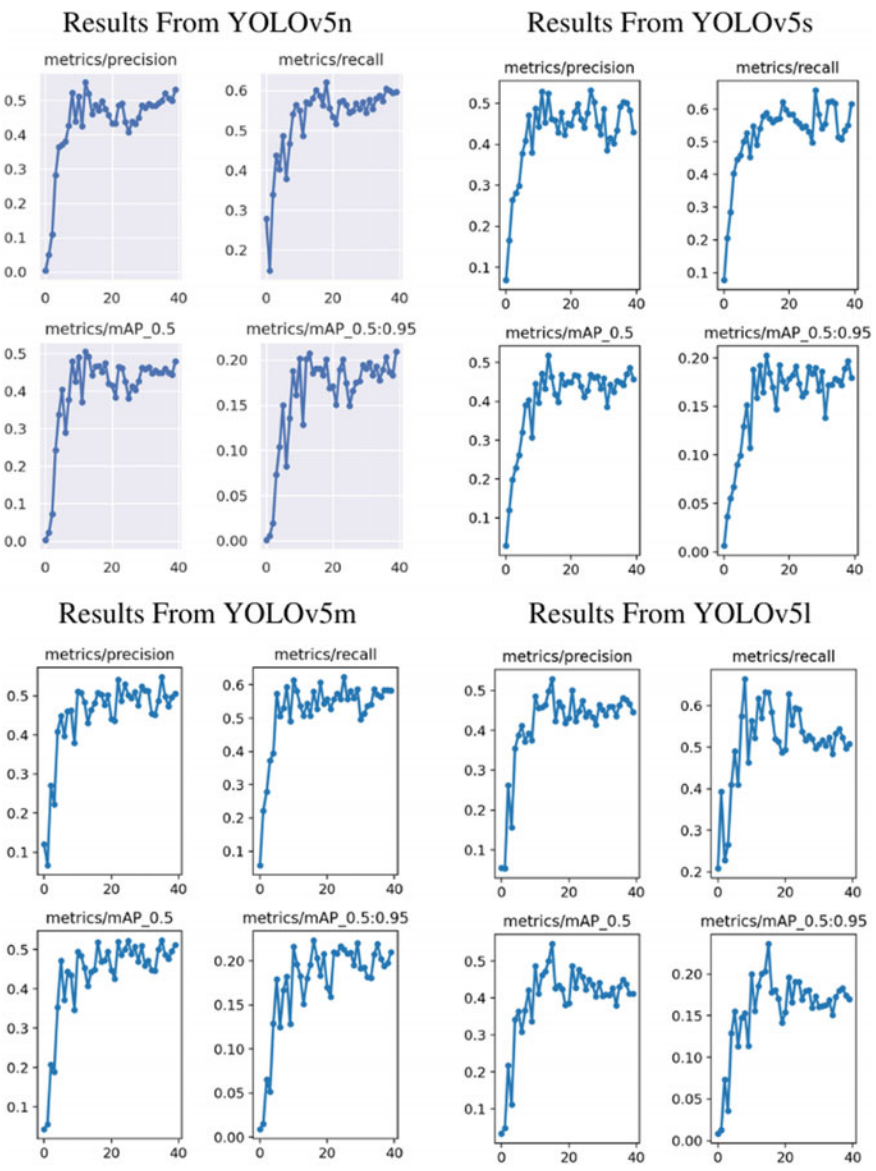


Fig. 3 Results after training with YOLOv5 models

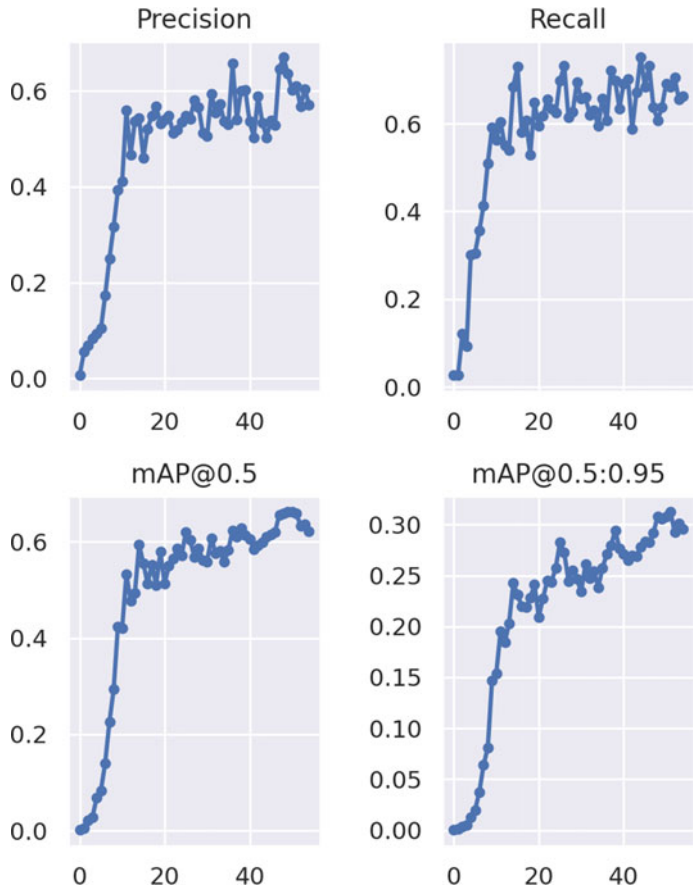


Fig. 4 Results after training with YOLOv7

Table 1 Implementation and results obtained from various YOLO models

Model	Metrics/Precision		Metrics/Recall		mAP(Mean average precision)
	Best	Least	Best	Least	
YOLOv5n	0.5529	0.0034716	0.62081	0.14765	0.47912
YOLOv5s	0.53215	0.068856	0.62416	0.028757	0.19221
YOLOv5m	0.54912	0.065842	0.61409	0.057047	0.52306
YOLOv5l	0.577	0.00499	0.635	0.279	0.576
YOLOv7	0.72	0.00652	0.751	0.0268	0.662

5 Results and Discussion

YOLOv5 is a powerful platform for object detection. The best metrics/precision retained from YOLOv5 model was 57.7% as compared to 75.1% of YOLOv7. Precision is an evaluation metric reflecting the number of positive prediction that were available. Recall is a measure that assesses the amount of correct positive predictions produced out of all possible positive predictions. The best metrics/recall was 63.5% for YOLOv5l, whereas YOLOv7 yielded a significantly higher value of 75.1%.

YOLOv7 shows a significant rise in trend in precision, accuracy, and recall. mAP value from YOLOv7 stood at 66.2% as compared to 57.6% obtained from YOLOv5l. From the run time and execution point of view, it is evident that YOLOv7 is faster by 120% compared to YOLOv5 [21]. From a similar environment for detection of potholes, mAP for YOLOv5 was found to be 58.9% as against 45.4% for Faster R-CNN [22].

It is perceived from results that finding compatible detail from images depends largely on the duration of training and amount of correctly defined data sets. A smaller data set could not assert predictions clearly, where accuracy is necessary. Time involved in acquiring large number of images for the training data set have slightly impacted the quality of detection and degree of accuracy. The usage of more critical image processing algorithms could also improve the output statistics and give more reliability in assessing details.

6 Future Work

Object detection using an image-based platform has high degree of accuracy. It can be trained based on data sets of different image qualities, texturing, grading, colour combinations, etc., to make crack detection more accurate. Shadows and angles are important as there are grey areas during detection. Based on the parametrization that the user defines for the algorithm determines the detection efficiency.

Scope of this project and its application is not limited to cracks. A multi-object detection module can be modelled using YOLO framework to detect two or more classes of objects. Further, research on this subject can be extended to larger civil structures such as dams, bridges, and buildings, where human interface can be seen as a ground station and drone as an aerial station. Multiple image processing techniques can be implemented to extract specific details from an input data set and variables can be parameterized into subsets for more powerful and accurate object detection.

7 Conclusion

The YOLOv5 and YOLOv7 algorithms were studied for detection of cracks on concrete walls as discussed in the paper. The YOLO framework was found to be convenient and the computational specification required to run the application was minimal. The target similarity matching of the detection algorithm was found to be in the range of 50% to 75% for validation data set. Real-time detection was studied through a pre-recorded video file, and it was in terms with test results already obtained.

YOLOv7 derives the detection speed and accuracy from model scaling techniques which demands varying degrees of accuracy and inference rates, while object detection models are often supplied as a series of models that scale up and down in size. The creators allow the model to scale the network depth and breadth simultaneously while concatenating layers together.

Another characteristic associated with the model is that it follows re-parametrization planning. It involves averaging a collection of model weights to produce a model that is more resistant to the broad patterns that it is attempting to create a model. Algorithm model employs gradient flow propagation channels to determine which network modules could use re-parameterization procedures.

An auxiliary head working coarse to fine acts as a node between network heads. The YOLO network head estimates the final network predictions. It has many layers in the network; an auxiliary head is added in the middle. The detection heads will make predictions while training. Creators of YOLOv7 have experiment with different levels of supervision for this head, settling on a coarse-to-fine definition where supervision is passed back from the lead head at different granularities [23].

Overall results indicate that the YOLOv7 model surpasses metrics evaluations obtained from other YOLOv5 models. With more training data set made to undergo learning, more accurate detection is expected from the algorithm.

Acknowledgements The authors bestow their sincere gratitude to Shri Mata Amritanandamayi Devi, Founder and Chancellor, Amrita Vishwa Vidyapeetham for her unending support. All the goodwill and timely guidance by Dr. Balakrishnan Shankar, Dean, Amrita School of Engineering, has proved instrumental in materializing the project to its completion. Further, we extend our indebtedness to god almighty who walked us through the emotionally straining period while doing this work.

References

1. Kim J-W, Kim S-B, Park J-C, Nam J-W (2015) Development of crack detection system with unmanned aerial vehicles and digital image processing. Advan Struct Eng Mech (ASEM15) 33(3):25–29
2. Yamaguchi T, Nakamura S, Saegusa R, Hashimoto S (2007) Image-based crack detection for real concrete surfaces. IEEJ Trans Electr Electron Eng 3:128–135

3. Agrawal A, Choudhary A (2016) Perspective: materials informatics and big data: realization of the fourth paradigm of science in materials science. APL Mater 4:053208
4. Gopalakrishnan K, Vidyadharan A, Gholami H, Choudhary AL (2018) Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model. Int J Traffic Transp Eng 8:1–14
5. Mohan A, Poobal S (2018) Crack detection using image processing: a critical review and analysis. Alexandria Eng J 57:787–798
6. Pei C, Qiu J, Liu H, Chen Z (2016) Simulation of surface cracks measurement in first walls by laser spot array thermography. Fusion Eng Des 109–111:1237–1241
7. Shirahata H, Miki C, Yamaguchi R, Kinoshita K, Yaginuma Y (2014) Fatigue crack detection by the use of ultrasonic echo height change with crack tip opening. Weld World 58:681–690
8. Wolf J, Pirskawetz S, Zang A (2015) Detection of crack propagation in concrete with embedded ultrasonic sensors. Eng Fract Mech 146:161–171
9. Rabah M, Elhattab A, Fayad A (2013) Automatic concrete cracks detection and mapping of terrestrial laser scan data. NRIAG J Astron Geophys 2:250–255
10. Yao Y, Tung S-TE, Glisic B (2014) Crack detection and characterization techniques-an overview. Struct Control Health Monit 21:1387–1413
11. Dinh TH, Ha QP, La HM (2016) Computer vision-based method for concrete crack detection. 2016 14th international conference on control, automation, robotics and vision (ICARCV)
12. Annamalai J, Lakshmikanthan C (2019) An optimized computer vision and image processing algorithm for unmarked road edge detection. Advan Intell Syst Comput 429–437
13. Sushant S, Anand S, James T, Aravind V, Narayanan G (2017) Localization of an unmanned aerial vehicle for crack detection in railway tracks. 2017 international conference on advances in computing, communications and informatics (ICACCI)
14. Dharneeshkar J, Aniruthan SA, Karthika R, Parameswaran L (2020) Deep learning based detection of potholes in Indian roads using Yolo. 2020 international conference on inventive computation technologies (ICICT)
15. Niranjan DK, Rakesh N (2020) Early detection of building collapse using IOT. 2020 second international conference on inventive research in computing applications (ICIRCA)
16. Jana S, Thangam S, Kishore A, Kumar VS, Vandana S (2022) Transfer learning based deep convolutional neural network model for pavement crack detection from images. Int J Nonlinear Anal Appl 13:1209–1223
17. Allaparthi SRR, Jeyakumar G (2022) An investigational study on ensemble learning approaches to solve object detection problems in computer vision. Math Stat Eng Appl 71(3s):399–412
18. Tzutalin (2015) LabelImg. Git code. https://github.com/tzutalin/labelImg
19. Wang C-Y, Bochkovskiy A, Liao H-YM (2022) YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696
20. Ultralytics (2020) Ultralytics/yolov5: Yolov5 in PyTorch & ONNX & CoreML & TFLite, https://github.com/ultralytics/yolov5
21. Kukil SR, Kukil RS (2022) Yolov7 paper explanation: Object detection and yolov7 pose. https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference
22. Ahmed KR (2021) Smart pothole detection using deep learning based on dilated convolution. Sensors 21:8406
23. Solawetz J (2022) Yolov7—a breakdown of how it works. https://blog.roboflow.com/yolov7-breakdown

Machine Learning-Based Identification as Well as Classification of Functional and Non-functional Requirements

R. D. Budake, S. D. Bhoite, and K. G. Kharade

Abstract In software engineering, it has become necessary to divide needs into functional and non-functional categories. This article identifies and categorizes functional (F) and non-functional (NF) needs and subclasses of NF requirements. Additionally, the subclasses of NF requirements will be discussed. A chunk of the data was gathered from various resources found on the internet. In this investigation, we began by cleaning the data by employing the normalization processes, and we then moved on to the succeeding steps, which included text preparation and vectorization. The topics covered included confusion matrix, Bag of Words, Term Frequency-Inverse Document, Featurization and Machine Learning Models, ROC and AUC curves, Bi-Grams and n-Grams in Python, and Word2Vec. The early discovery of NFRs allows us to make preliminary design choices. We used machine learning algorithms to detect and categorize both functional and non-functional needs for application software development, based on the user requirements provided to us. This article aims to assist in the application development process to various software professionals, including software developers, software designers, software testers, and so on. This paper is also beneficial for developing software more quickly and delivering it to customers. Taking this step makes it easier to create an SRS document and helps the requirement analysis phase proceed more smoothly by reducing the amount of unneeded labor and complexity.

Keywords Bag of words · ML · NLP · TF-IDF

R. D. Budake (✉)
Department of Computer Science, K.H. College, Gargoti, Maharashtra, India
e-mail: rajendrabudake@gmail.com

S. D. Bhoite
CSIBER, Kolhapur, Maharashtra, India

K. G. Kharade
Department of Computer Science, Shivaji University, Kolhapur, Maharashtra, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_38

1 Introduction

In practice, requirements are typically divided into functional and non-functional. The functional requirements of a system define the system's functionality or the process that the system must do. In contrast, the non-functional requirements of a system describe the operational features or properties of the associated system and the limitations. We teach the machine to process and evaluate vast volumes of natural language datasets as part of Natural Language Processing. Various phases involve natural language processing (NLP), including row text, tokenization, text cleaning, preprocessing, and vectorization. Text cleaning and preparation are activities that vary widely depending on the application and dataset being utilized [1]. The identification of spam as well as the organization of news can be accomplished by text categorization. When carrying out the vectorization process, a token is utilized as the input, and a feature vector is produced as the final result. Bag of Words (BOW) or Term Frequency-Inverse Document Frequency are also names used to refer to this concept. In order to categorize software requirements, the results of vectorization methods can be employed in a supervised classification algorithm such as a Support Vector Machine (SVM), K-Nearest Neighbor, or Naive Bayes [2].

2 Developed System

The effects of text normalization, feature extraction approaches, and the performance of three machine learning algorithms, SVM, k-NN, and Naive Bayes, combined with BoW and TF-IDF are reviewed in this work. BoW and TF-IDF are both used to analyze the data. In particular, attention is being paid to the performance of the SVM method because it is the most frequently utilized for text classification. Because SVM, KNN, and Naive Bayes are the algorithms that are used the vast majority of the time, this study focuses on how well these three perform [3]. To accomplish the task of needs classification, a method of feature extraction and an algorithm with the highest possible degree of precision will be determined as part of the process. This will allow for the task to be carried out successfully. Finally, the model is built and deployed, which identifies and classes the FRs and NFRs, in addition to their subclasses, in an unbalanced dataset. This model also classifies the subclasses of the FRs and NFRs. A dataset that has been trained will be given to the training classification algorithm, which will then produce a dataset that has been categorized [4].

See Fig. 1.

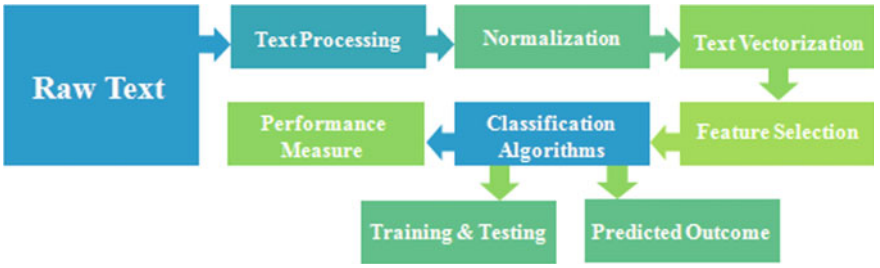


Fig. 1 Requirements classification model

2.1 Machine Learning (ML) Algorithms

Learning by machine refers to the process by which artificial intelligence can learn new code. It indicates that the program needs to be run once, regardless of how often it experiences difficulty; after that, it should not be programmed again. It adapts its code to the different scenarios it finds as it learns more about them. It is self-learning, so whatever it is that has to be learned from it, we will offer it with situations, and with those scenarios, it will provide it with past experiences. We will feed the values to it, and after learning from those previous experiences, it will come up with new answers [5]. In this study, supervised learning methods are put to use:

(a) **K-Nearest Neighbor (KNN)**: The KNN algorithm is a non-parametric, supervised learning classifier that uses proximity to make classifications or predictions about the grouping of an individual data point. It does this by comparing the location of the individual data point to other points in the dataset. In some circles, it is also referred to as KNN or k-NN. Although it can be applied to problems involving regression or classification, it is more commonly used as a classification algorithm because it assumes that points with similar characteristics can be found close to one another. Although it can be applied to regression or classification problems, it is more commonly used as a classification algorithm. The k-Nearest Neighbor algorithm categorizes data according to how similar it is to other data. KNN solves regression problems by determining the average or mode of the k instances most similar [6].

(b) **Support Vector Machine (SVM)**: The Support Vector Machine, sometimes called SVM, is an algorithm that may effectively solve classification and regression issues. An SVM will generate a linear hyperplane based on the largest margin between two classes. The marginal distance is between the nearest positive and negative points generated by parallel lines. It does not include any other information. Linearly and non-linearly, the points are simple to distinguish from one another. An unlimited number of points can traverse the marginal plane due to the absence of any limiting conditions. Finally, we can develop a generalized model with the help of SVM [7].

(c) **Naive Bayes Algorithm**: Text classification is the issue statement that can be discovered in many blogs or projects classified by Naive Bayes classifiers. It doesn't matter if it's good or poor; it just is. This is another approach believed to be a baseline for text classification. Naive Bayes is a straightforward supervised machine learning algorithm that derives its results by applying Bayes' theorem, assuming a high degree of independence between the various features. This is done in the context of assuming a high degree of independence between the various features. This only indicates that the algorithm works under the presumption that the variables that are fed into it are unrelated to one another in any way. It demonstrates an exceedingly ignorant attitude to make such an assumption based on evidence from the real world [8].

2.2 Performance Measures

In machine learning, this is the final stage of the classification process that is utilized for solving classification problems. Regarding the success of software systems, accuracy, security, and performance are three of the most significant features. The performance of our machine learning model may be measured in various ways, including the Confusion Matrix, Precision, Recall, F1-Score, and Accuracy. In this study, we evaluated the effectiveness of our machine learning model by calculating ROC and AUC scores [9].

3 Phases Incorporated in Developed System

3.1 Text Normalization

Normalizing the text is a necessary step before beginning its analysis of it. Cleaning the data begins with this phase, which is the first stage. Pronouns and articles are removed from the text when converted from uppercase to lowercase. The initial nouns and verbs are converted into appropriate forms [10].

See Tables 1, 2 and 3.

3.2 Text Vectorization

Since tokens are taken as an input and a feature vector is produced as an output, text vectorization is sometimes referred to as feature extraction. The algorithms that power machine learning cannot comprehend the alphabet or the English language words; they can only grasp numbers (0 and 1). The requirements for the normalized software are subsequently translated into numerical vectors. The BoW and TF-IDF

Table 1 Identify the requirements

F	210
US	63
O	58
SE	57
PE	54
LF	34
A	21
SC	21
MN	17
L	10
FT	9
PO	2

Table 2 Normalization and cleaning and assign values 1 for functional and 0 for non-functional requirements

Class	Text
0	The system shall refresh the display every 60 …
0	The application shall match the color of the s …
0	If projected the data must be readable. On a …
0	The product shall be available during normal b …
0	If projected the data must be understandable …
0	The product shall ensure that it can only be a …
0	The product shall be intuitive and self-explain …
0	The product shall respond fast to keep up-to-d …
1	The system shall have a MDI form that allows f …
1	The system shall display Events in a vertical…

Table 3 Pre-processing text

Class	Text	Pre-processed text
0	The system shall refresh the display every 60 …	System shall refresh display every second …
0	The application shall match the color of the s …	Application shall match color schema set forth …
0	If projected the data must be readable. On a …	Projected data must be readable projection screen …
0	The product shall be available during normal b …	Product shall be available normal business hours …
0	If projected the data must be understandable …	Projected data must be understandable projection …

```
(249,) (249,)
(184,) (184,)
(123,) (123,)
================================================================================
====================
After vectorizations
(249, 912) (249,)
(123, 912) (123,)
(184, 912) (184,)
================================================================================
====================
```

Fig. 2 Text vectorization using BoW

```
some sample features(unique words in the corpus) ['able', 'access', 'allow', 'a
vailable', 'data', 'disputes', 'information', 'interface', 'must', 'new']
==================================================
After vectorizations
(249, 31) (249,)
(123, 31) (123,)
(184, 31) (184,)
================================================================================
====================
```

Fig. 3 Text vectorization using TF-IDF

conversion methods are utilized. Text vectorization using BoW and TF-IDF are the program's products [11].

See Figs. 2 and 3.

3.3 Classification

Text vectorization techniques ultimately result in supervised classification algorithms such as SVM, KNN, and the Naive Bayes Algorithm. The ultimate result of these strategies is these algorithms, in addition to the criteria for BoW and TF-IDF classification software [12].

3.3.1 Naïve Bayes with Bow

See Figs. 4 and 5.

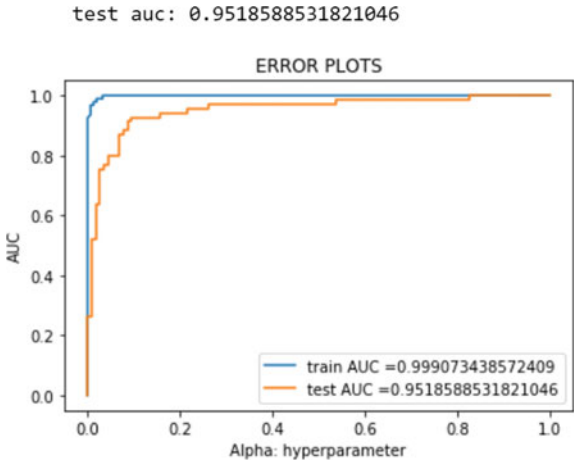
3.3.2 Naive Bayes with TF-IDF

See Figs. 6 and 7.

```
100%|
     | 15/15 [00:00<00:00, 206.26it/s]
The 'α' value 0.5 with highest roc_auc Score is 86.61814109742441 %
```

Fig. 4 Here alpha is a hyperparameter. For BoW, the alpha is 0.5. It will be applied on test data

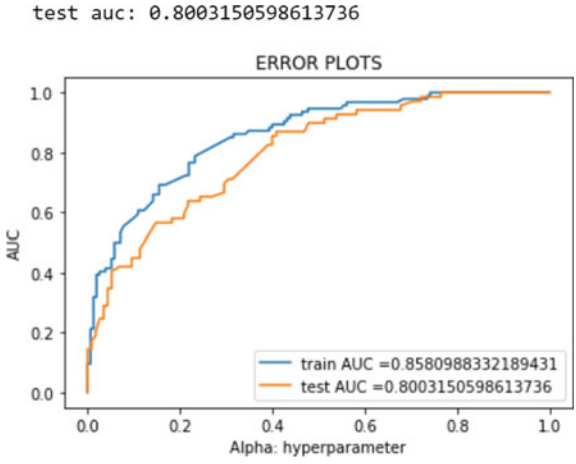
Fig. 5 BoW has predicted 95.18% accuracy on test data using Multinomial Naive Bayes




```
100%|
     | 15/15 [00:00<00:00, 192.83it/s]
The 'α' value 0.01 with highest roc_auc Score is 73.51623740201568 %
```

Fig. 6 The best alpha is 0.01 and applies it to test data

Fig. 7 TF-IDF has predicted 80.03% accuracy on test data using Multinomial Naive Bayes



```
The 'K' value 3 with highest roc_auc Score is 72.32642777155655
```

Fig. 8 Where k is a hyperparameter. For Bow 3 is the best value for k

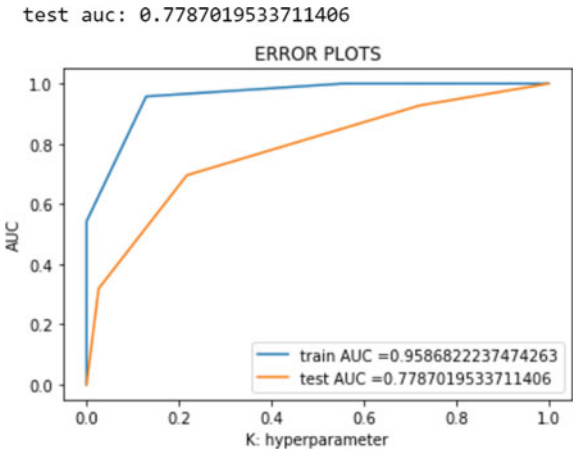


Fig. 9 BoW has predicted 77.87% accuracy on test data using K-NN Brute force

3.3.3 K-NN Brute Force with BoW

See Figs. 8 and 9.

3.3.4 K-NN Brute Force with TF-IDF

See Fig. 10.

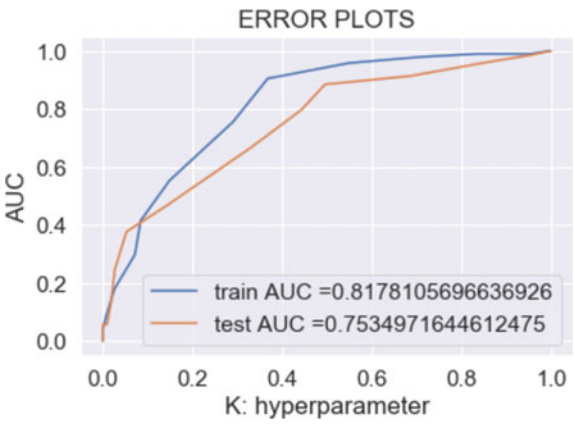


Fig. 10 TF-IDF has predicted 75.34% accuracy on test data using K-NN Brute force with k value 15

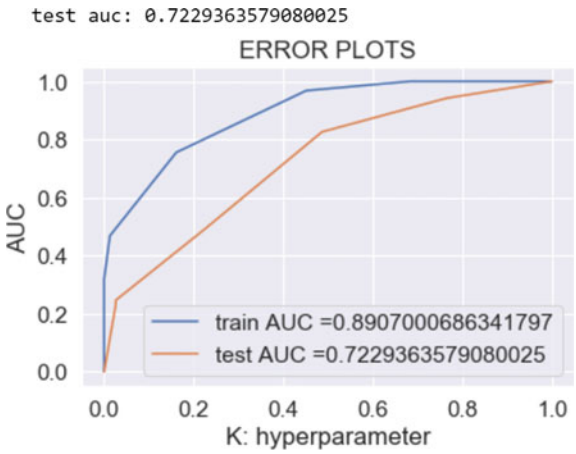


Fig. 11 BoW has predicted 72.29% accuracy on test data using K-NN Kd_tree with k value 5

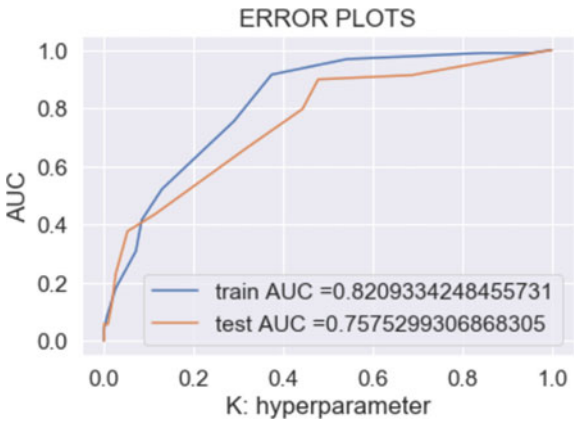


Fig. 12 TF-IDF has predicted 75.75% accuracy on test data using K-NN Kd_tree

3.3.5 K-NN Kd-Tree with BoW

See Fig. 11.

3.3.6 K-NN Kd_tree with TF-IDF

See Fig. 12.

3.3.7 SVM with BoW

See Figs. 13 and 14.

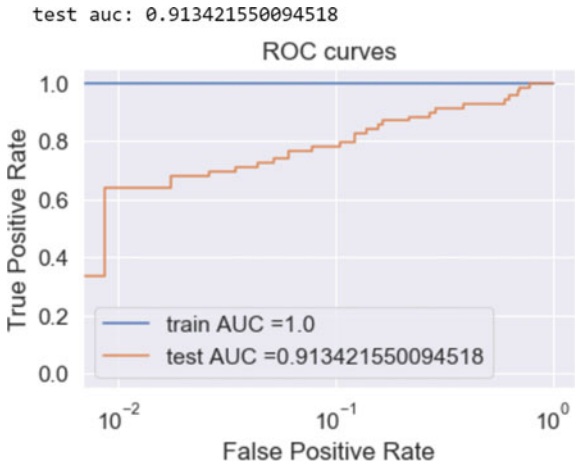


Fig. 13 BoW has predicted 91.34% accuracy on test data using SVM with L1 regularize

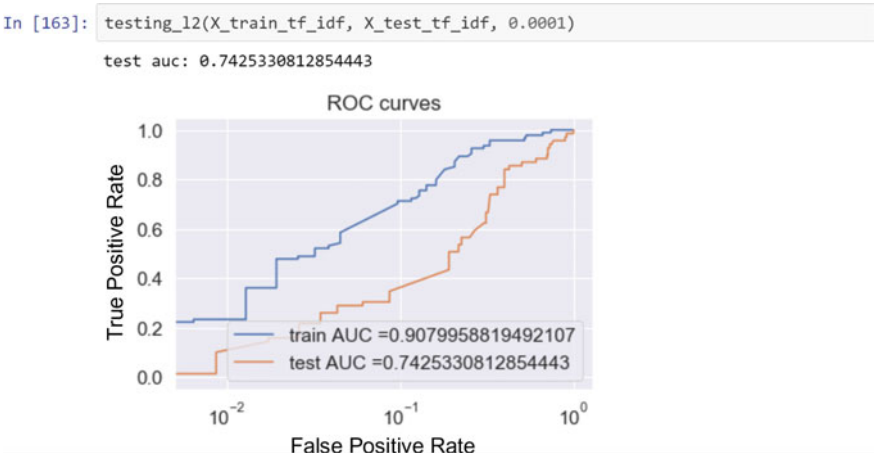


Fig. 14 TF-IDF has predicted 74.25% accuracy on test data using SVM with L2 regularizer

4 Result and Discussion

The following Tables 4, 5, 6 and 7 show the accuracy of the above models with their parameters.

Table 4 Result of using Multinomial Naive Bayes model

Vectorizer	Hyperparameter (α)	Test AUC (%)
BoW	0.5	95.18
TF-IDF	0.01	80.03

Table 5 Result of using K-NN Brute model

Vectorizer	Hyperparameter (k)	Test AUC (%)
BoW	3	77.87
TF-IDF	15	75.75
Avg Word2Vec	9	59.94
TF-IDF Word2Vec	11	64.93

Table 6 Result of using KD tree used with KNN model

Vectorizer	Hyperparameter (k)	Test AUC (%)
BoW	5	72.29
TF-IDF	15	75.75
Avg Word2Vec	9	64
TF-IDF Word2Vec	11	67

Table 7 Result of using SVM model

Vectorizer	L1 (alpha)	Test AUC (L1) (%)	L2 (alpha)	Test AUC (L2) (%)
BoW	0.0001	91.34	0.0001	90.79
TF-IDF	0.0001	79.27	0.05	74.25

In this study, machine learning techniques were used to anticipate output through text categorization across various software needs accurately. This was accomplished by using a large number of different software requirements. After we have finished training the model with the help of the training data, the model will need to be cross-validated. When it happens, we will finally be able to figure out the hyperparameters. With the assistance of cross-validation, we can determine the values of our model's hyperparameters. Ultimately, we put the model through its paces to assess its accuracy and see how well it performs. The performance of our machine learning classifier can be evaluated by looking at the area under the receiver operating characteristic curve (AUC-ROC) for binary classification. This gives us an idea of how well the classifier performs its job. When applied to test data, we observed that the Multinomial Naive Bayes model that BoW utilized attained a level of predicted accuracy that was 95% accurate. Tf-predictions made using Multinomial Naive Bayes and based on test data were accurate 80.03% of the time. When using K-NN Brute force on BoW's test data, the results of those predictions were accurate 77.87% of the time. When applied to the test data, the K-NN Kd tree's TF- and IDF predictions proved to be accurate 75.75% of the time. The SVM showed that the bow had a prediction accuracy of 90.79% based on the data that was tested. The Tf-predictions and Idf's that were performed using SVM on the test data were accurate 74.25% of the time. It is demonstrated by auto detection and classification of functional and non-functional requirements and subclasses using an unbalanced dataset.

5 Conclusion

The accuracy that was predicted by BoW using Multinomial Naive Bayes was 95.18%, the accuracy that was forecast by BoW using K-NN Brute Force was 77.87%, the accuracy that was projected by TF-IDF using K-NN Kd tree was 75.75%, and the accuracy that was predicted by Bow using SVM was 90.79%. It has been found that the combination of BoW and Multinomial Naive Bayes provides the greatest performance for binary classification. We have created and deployed a model that predicts an outcome of recognizing and categorizing functional and non-functional needs and subclasses in an unbalanced dataset. This model was developed based on the requirements that the users provided. This article aims to provide insights into stakeholders' requirements, document software, minimize the amount of work that needs to be redone, and make the process of developing an application as simple as possible.

References

1. Algabri HK, Kharade KG, Kamat RK (2021) Promise, threats, and personalization in higher education with artificial intelligence. Webology 18(6):2129–2139
2. Ali H, Kharade KG, Kamat RK (2022) An analysis and evaluation of vulnerability assessment tools. Cyberpsychol Behav Soc Netw 25(4):448–456. https://doi.org/10.1166/cyber.2022.29299.editorial
3. Binkhonain M, Zhao L (2019) A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Syst Appl X 1:100001. https://doi.org/10.1016/j.eswax.2019.100001
4. Chougale NP, Kharade KG, Kharade SK, Ghatage SR, Mendagudli MG, Yuvaraj S, Vengatesan K (2021) Deployment of computer assisted instruction in higher educational organization. In: Recent trends in intensive computing. IOS Press, pp 461–466
5. Chung L, Nixon BA (1995) Dealing with non-functional requirements: three experimental studies of a process-oriented approach. In: Proceedings of the 17th international conference on software engineering (ICSE'95), pp 25–37. https://doi.org/10.1145/225014.225017
6. Kharade KG, Katkar SV, Patil NS, Sonawane VR, Kharade SK, Pawar TS, Kamat RK (2021) Text summarization of an article extracted from Wikipedia using NLTK Library. In: Singh M, Tyagi V, Gupta PK, Flusser J, Ören T, Sonawane VR (eds) Advances in computing and data sciences. ICACDS 2021. Communications in computer and information science, vol 1441. Springer, Cham, pp 195–207. https://doi.org/10.1007/978-3-030-88244-0_19
7. Nadeem MA, Lee SU-J (2020) Machine learning evaluation of the requirement engineering process models for cloud computing and security issues. Appl Sci 10(5851):1–13.https://doi.org/10.3390/app10175851
8. Mills C, Escobar-Avila J, Haiduc S (2018) Automatic traceability maintenance via machine learning classification. In: 2018 IEEE international conference on software maintenance and evolution (ICSME). IEEE, pp 369–380. https://doi.org/10.1109/ICSME.2018.00045
9. Kurtanović Z, Maalej W (2017) Automatically classifying functional and non-functional requirements using supervised machine learning. In: 2017 IEEE 25th international requirements engineering conference (RE). IEEE, pp 490–495. https://doi.org/10.1109/RE.2017.82
10. Kharade KG, Kamat RK, Kharade SK (2019) Online library package to boost the functionality and usability of the existing libraries. Int J Future Revolution Comput Sci Commun Eng 5(8):5–7

11. Patil BP, Kharade KG, Kamat RK (2020) Investigation on data security threats and solutions. Int J Innovative Sci Res Technol 5(1):79–83
12. Katkar SV, Kharade KG, Patil NS, Sonawane VR, Kharade SK, Kamat RK (2021) Predictive modeling of tandem silicon solar cell for calculating efficiency. In: Singh M, Tyagi V, Gupta PK, Flusser J, Ören T, Sonawane VR (eds) Advances in computing and data sciences. ICACDS 2021. Communications in computer and information science, vol 1441. Springer, Cham, pp 183–194. https://doi.org/10.1007/978-3-030-88244-0_18

Fake News Detection in Dravidian Languages Using Transformer Models

Eduri Raja, Badal Soni, and Samir Kumar Borgohain

Abstract Nowadays, fake news is spreading rapidly. Many resources are available for fake news detection in high-resource languages like English. Due to the lack of annotated data and corpora for low-resource languages, detecting fake news in low-resource languages is difficult. There is a need for a system for fake news detection in low-resource languages like Dravidian languages. In this research, we used Telugu, Kannada, Tamil, and Malayalam languages and tested with four transformer models: mBERT, XLM-RoBERTa, IndicBERT, and MuRIL. MuRIL gives the best accuracy in these models compared to the remaining models.

Keywords Fake news · Transformers · Low-resource languages

1 Introduction

False information that is presented as news is called fake news [1]. Technological improvements have made much information available on various digital platforms, but there are no suitable methods for filtering or validating this information. As a result, consumers are frequently unaware of the authenticity of news articles. According to several researchers, fake news spreads more quickly than actual news. Fake news can also disturb governments, which hurts communities and people. In this research, we concentrate on identifying fake news in low-resourced Dravidian languages; we have compared some sophisticated pre-trained NLP models with the Dravidian fake news dataset, including mBERT (multilingual Bidirectional Encoder Representations from Transformers), XLM-RoBERTa (Robustly optimized BERT

E. Raja (✉) · B. Soni · S. K. Borgohain
National Institute of Technology Silchar, Silchar, Assam, India
e-mail: eduri_rs@cse.nits.ac.in

B. Soni
e-mail: badal@cse.nits.ac.in

S. K. Borgohain
e-mail: samir@cse.nits.ac.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_39

training approach), IndicBERT, and MuRIL (Multilingual Representations for Indian Languages).

Social media use has been widespread among people of all ages in the Internet age. The web has developed into a one-stop destination for many kinds of activity, from education to entertainment. However, it is also filled with misinformation and offensive and distressing information that could be dangerous to everyone. This can be avoided by developing an automated method for identifying and flagging fake news content. Although extensive research has been done to determine fake news in high-resource languages like English, locating and identifying objectionable information in languages with low resources is challenging. The lack of resources and annotated data makes it difficult to develop a fake news detection model in Dravidian languages.

The paper is organized as follows. Section 2 contains related works in fake news detection system using Natural Language Processing (NLP). Section 3 presents our work's methodology. Section 4 contains the results and discussions part. Section 5 concludes with a summary and recommendations for further research.

2 Related Works

There has been numerous research on how to identify fake news. Most of these experiments are based on different CNN and LSTM-based architectures. A few researchers developed models using the BERT architecture. In the pre-training stage of BERT, the authors incorporated more data from Wikipedia and Book Corpus to improve its detection capabilities. On real-world datasets for fake news, several studies developed models that integrated various parallel blocks of a single-layer CNN with BERT and outperformed the current models. Other research evaluated various models based on CNN, LSTM, and BERT on the fake news corpus and found that BERT performed well among these models. These studies collectively show the effectiveness of BERT-based algorithms in identifying fake news.

Horne et al. [2] developed a model to identify psychological and stylometric features in the news headlines to assess if the news is accurate or false. Przybyla et al. [3] developed a fake news classifier which is a style-based text classifier using BiLSTM algorithm in order to extract the style-based elements from news articles. Zellers et al. [4] developed a model to find the accuracy of news reported in news articles using neural networks. Silva et al. [5] developed automatic filtering of fake news in Portuguese using different word embedding methods with machine learning models. Potthast et al. [6] analyzed the extremely one-sided news (Hyperpartisan), whether fake or real, using a meta-learning approach.

Cui et al. [4] proposed DEFEND: an explainable LSTM-based fake news detecting system using long short-term memory (LSTM), to assess whether any news is real or false, this model takes into account user comments. Nguyen et al. [7] proposed FANG: a fake news detection model to learn the representations in social contexts using graph learning framework. In recent years, pre-trained models have received more attention in NLP research. Modern pre-trained language models include BERT

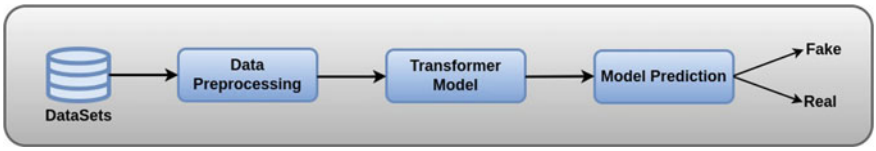


Fig. 1 Transformer-based fake news detection model architecture

[8] and GPT-2 [9]. To extract the most knowledge from the data, the language model (such as BERT or GPT-2) is pre-trained on the unlabeled text in the first stage (unsupervised learning). A small-labeled dataset is used in the second stage to refine the model for various applications. Semi-supervised sequence learning is used in this task. These models are also utilized in the investigation of fake news.

Some fake news detection techniques employ BERT [10–12] to determine whether a piece of news is authentic or fake. BERT is more suited to NLP tasks like text classification and translation since it employs bidirectional representations to learn information. On the other hand, the GPT-2 utilizes a unidirectional representation and left-to-right context to predict the future and is more effective for auto-regressive tasks where timeliness is essential.

3 Methodology

An overview of the approach we used for this study is given in this section, including data preparation, modeling techniques, and training methods are shown in Fig. 1.

3.1 Datasets

In this research, we used four datasets for fake news detection Dravidian languages: Telugu, Kannada, Tamil, and Malayalam. These datasets are taken from GitHub.[1] Table 1 represents the dataset statics. The datasets are slightly imbalanced. We performed data preprocessing on these datasets and used balanced data in this research. Figure 2 represents the sample data for Dravidian language datasets.

[1] https://github.com/NLP-Researcher/Indic-fake-news-datasets.

Text	Label
ఫ్రాన్స్‌కు చెందిన దసాల్ట్ సంస్థ ఉత్పత్తి చేస్తున్న టీబీఎం ఎయిర్‌క్రాఫ్ట్‌లు భారత్‌కు రానున్నాయి.	Real
అజర్ మసూద్‌ను ఉగ్రవాదిగా భారత ప్రతిపక్షాలు పరిగణించకపోతే మేం ఎలా అంగీకరిస్తామని ఐరాసలో చైనా వాదించింది.	Fake
ಒಂದುಗೂಡಿವೆ ಜಿಯೋ ಮ್ಯೂಸಿಕ್ ಸಾವನ್ ಕಲಾವಿದರಿಗಾಗಿ ದಕ್ಷಿಣ ಏಷ್ಯಾದಲ್ಲೇ ಅತಿದೊಡ್ಡ ವೇದಿಕೆ ಸೃಷ್ಟಿ	Real
ಮುಂಬೈನಲ್ಲಿ 188 ಕರೋನಾ ರೋಗಿಗಳಿಗೆ ಚಿಕಿತ್ಸೆ ನೀಡಿದ ನಂತರ ಡಾ. ಮನೀಶಾ ಪಾಟೀಲ್ ಅವರು ತಮ್ಮನ್ನು ಉಳಿಸಿಕೊಳ್ಳಲು ಸಾಧ್ಯವಾಗಲಿಲ್ಲ	Fake
பேட்டரி வாகனத்துக்கு அரசு முன்னுரிமை: அச்சத்தில் ஆட்டோமொபைல் துறை	Real
கோவை குடிநீர் கோவிந்தா சர்ச்சையை கிளப்பும் பேஸ்புக் பதிவு	Fake
ഐ എസ് എല് ആദ്യ സെമിയില് നോര്ത്ത് ഈസ്റ്റ് ഇന്നലെ ബാംഗളൂര് എഫ് സിയെ പരാജയപ്പെടുത്തി	Real
പിണറായിയുടെ വാദം തെറ്റ്? സംസ്ഥാനത്ത് ദാരിദ്ര്യം കുറവെന	Fake

Fig. 2 Sample data for Dravidian fake news datasets

Table 1 Dataset statistics

Datasets	Real	Fake	Total
Telugu	3236	3245	6481
Kannada	3059	3220	6279
Tamil	3195	3086	6281
Malayalam	4398	3913	8311

3.2 Data Preprocessing

In this step, we removed special characters, symbols, white spaces, etc., using the Python RegEx module. Preprocessing is the transformation of data in order to prepare it for embedding. These preprocessed data provide input to the transformer models.

3.3 Transformer Models

We use pre-trained transformer models for identifying fake news in Dravidian languages. We have used MuRIL, IndicBERT, mBERT, and XLM-RoBERTa pre-trained models from huggingface.

mBERT: Multilingual BERT [14] is a transformer model that has been self-supervised pre-trained on a sizable corpus of multilingual data. Utilizing a Masked Language Modeling (MLM) objective, a trained model was created using the top 104 languages with the most Wikipedia pages. WordPiece is used to tokenize and lowercase the texts, and a shared vocabulary with a size of 110,000 is used.

XLM-RoBERTa: A multilingual version of RoBERTa is called XLM-RoBERTa [16]. 100 languages from 2.5TB of filtered CommonCrawl data were used as its pre-training material. RoBERTa is a transformer model that was self-supervised and pre-trained on a sizable corpus.

IndicBERT: IndicBERT [13] is a multilingual ALBERT model that has only been trained in the 12 major Indian languages. It is tested on a wide range of task sets after being pre-trained on 9 billion tokens on the monolingual corpus. IndicBERT performs as well as or better than other multilingual models (mBERT, XLM-RoBERTa, etc.) despite having fewer parameters.

MuRIL: Multilingual Representations for Indian Languages (MuRIL) [15] is a pre-trained model. This model is pre-trained from scratch using a BERT base architecture. This model is pre-trained on 17 Indian languages and their transliterated data. For 17 Indian languages, this model has been pre-trained using Wikipedia, Common Crawl, PMINDIA, and Dakshina corpora.

3.4 Hyperparameter Tuning

Due to catastrophic forgetting, transformer models are still unstable. We chose our experimental hyperparameters for this fake news detection task very carefully. In this experiment, we split the dataset into 80:20 ratio. 80% for training and validation and 20% for testing purposes. We implemented our model with Pytorch using Quadro P4000 GPU in the Jupyter notebook. We used the pre-trained models from huggingface. We tried different batch sizes 128, 64, 32, and 16. Batch sizes 128 and 64 did not work, so we decreased batch sizes to 32 and 16. For batch sizes 32 and 16, our model works well. We tried batch sizes 32 and 16, and batch size 32 gave better results than batch size 16. Therefore, during the training and validation processes, the batch size of 32 is maintained. The maximum sequence length is 128, and there are 10 training epochs. We tried different learning rates; for this research best optimal training rate is 2e−5.

4 Results and Discussions

This section presents our experimental results. We experimented with the four transformer models with different batch sizes, epochs, and learning rates. In this research, we used Accuracy, F1-Score, Precision, and Recall classification metrics for the

Table 2 Evaluation metrics for Telugu dataset

Models	Accuracy	F1-Score	Precision	Recall
mBERT	86.49	86.95	88.98	85.01
XLM-RoBERTa	88.26	88.21	86.81	89.66
IndicBERT	88.93	90.24	92.65	87.96
MuRIL	89.86	91.33	93.72	88.23

Table 3 Evaluation metrics for Kannada dataset

Models	Accuracy	F1-Score	Precision	Recall
mBERT	86.08	86.03	85.97	86.09
XLM-RoBERTa	86.97	87.29	89.81	84.92
IndicBERT	87.26	87.4	88.63	86.21
MuRIL	88.07	88.02	87.89	88.15

Table 4 Evaluation metrics for Tamil dataset

Models	Accuracy	F1-Score	Precision	Recall
mBERT	85.75	85.77	87.24	84.35
XLM-RoBERTa	86.79	86.69	87.43	85.96
IndicBERT	87.72	87.81	89.9	85.82
MuRIL	88.66	88.74	90.86	86.73

Table 5 Evaluation metrics for Malayalam dataset

Models	Accuracy	F1-Score	Precision	Recall
mBERT	82.2	82.9	85.73	80.24
XLM-RoBERTa	82.91	83.52	86.02	81.16
IndicBERT	83.7	83.77	83.59	83.95
MuRIL	84.85	85.38	87.87	83.02

model evaluation. In these four models, language-wise performance metrics are described in Tables 2, 3, 4, and 5. Classification metrics graphs are illustrated in Figs. 3, 4, 5, and 6. Among the four Dravidian languages, the Telugu language has better accuracy than the other three. Comparing the four models, the MuRIL transformer model provides the best accuracy because MuRIL is trained in 16 Indian languages, including four Dravidian languages and is trained on various corpora. MuRIL and IndicBERT gave the best accuracy and F1-Scores for the Dravidian languages compared to mBERT and XLM-RoBERTa because MuRIL and IndicBERT are pre-trained models trained on the large corpora of Indian languages.

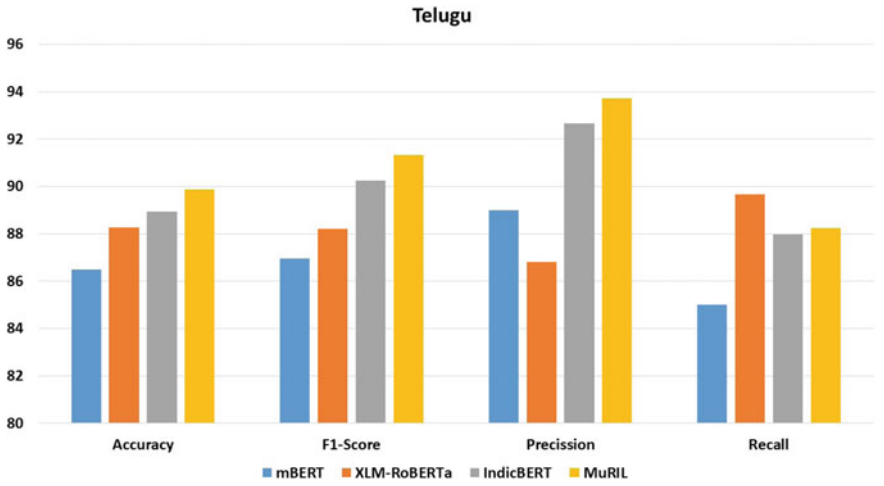


Fig. 3 Classification metrics for Telugu

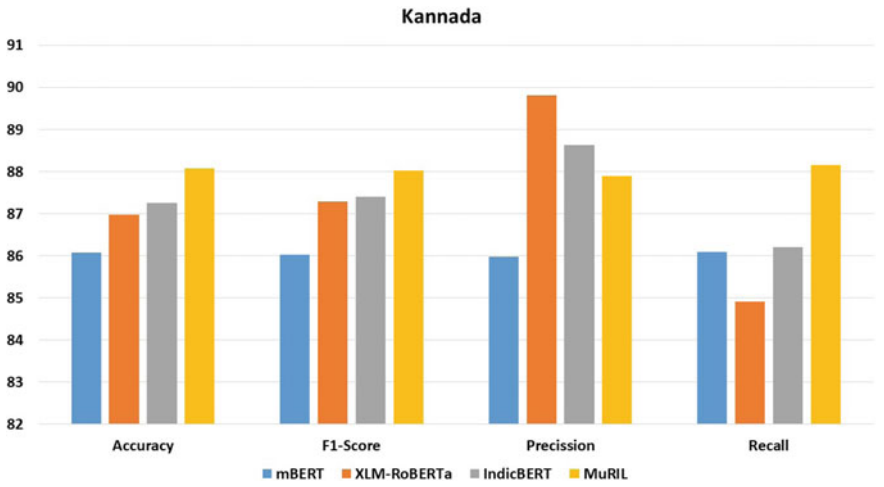


Fig. 4 Classification metrics for Kannada

5 Conclusion

This paper presents a fake news detection system in Dravidian languages using pre-trained transformer models: mBERT, XLM-RoBERTa, IndicBERT, and MuRIL. In the four pre-trained transformer models, the MuRIL transformer model gave the best results for the Telugu, Kannada, and Tamil datasets with 89.86, 88.07, and 88.66 percent accuracy, and it gave less accuracy for the Malayalam dataset. F1-Score shows that MuRIL and IndicBERT models perform very well in the Dravidian language's

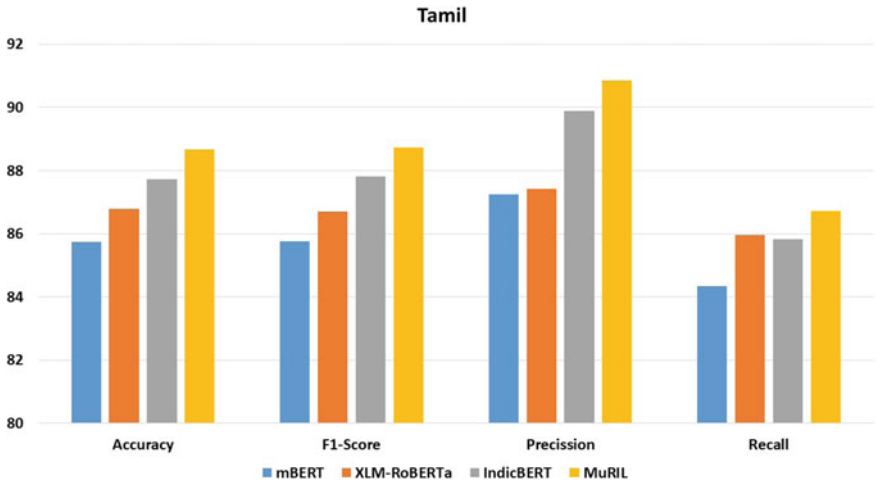


Fig. 5 Classification metrics for Tamil

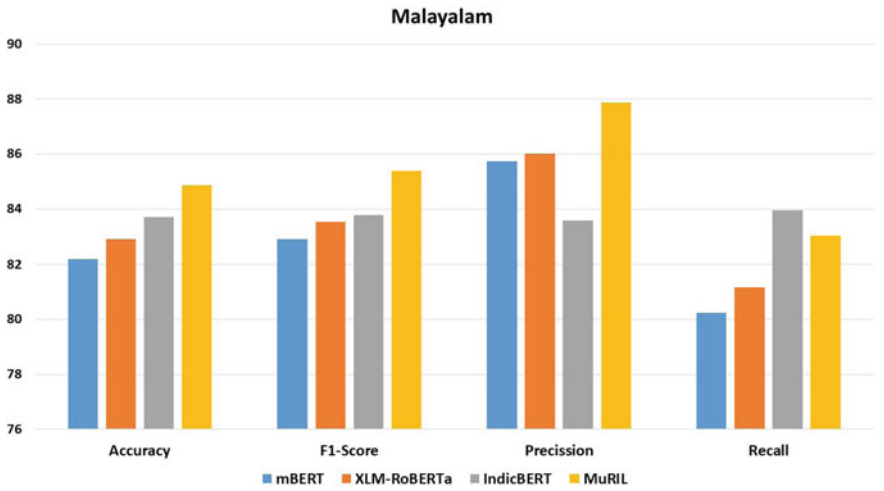


Fig. 6 Classification metrics for Malayalam

fake news datasets to identify fake news compared to mBERT and XLM-RoBERTa. We will consider improving the results for future research using customized pre-trained models and ensemble methods with larger datasets.

References

1. Shu K, Mahudeswaran D, Wang S, Lee D, Liu H (2020) FakeNewsNet: a data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data 8:171–188. https://doi.org/10.1089/big.2020.0062
2. Horne B, Adali S (2017) This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: Proceedings of the international AAAI conference on web and social media (2017)
3. Przybyla P (2020) Capturing the style of fake news. In: Proceedings of AAAI conference on artificial intellegence, vol 34, pp 490–497. https://doi.org/10.1609/aaai.v34i01.5386
4. Zellers R, Holtzman A, Rashkin H, Bisk Y, Farhadi A, Roesner F, Choi Y (2020) Defending against neural fake news. Neurips
5. Silva RM, Santos RLS, Almeida TA, Pardo TAS (2020) Towards automatically filtering fake news in Portuguese. Expert Syst Appl 146:113199. https://doi.org/10.1016/j.eswa.2020.113199
6. Potthast M, Kiesel J, Reinartz K, Bevendorff J, Stein B (2017) A stylometric inquiry into hyperpartisan and fake news. arXiv Preprint http://arxiv.org/abs/1702.05638
7. Nguyen VH, Sugiyama K, Nakov P, Kan MY (2020) FANG: leveraging social context for fake news detection using graph representation. In: International conference on information & knowledge management. Proceedings. https://doi.org/10.1145/3340531.3412046
8. Devlin J, Chang MW, Lee K, Toutanova K (2018) BERT: pretraining of deep bidirectional transformers for language understanding. arXiv Preprint http://arxiv.org/abs/1810.04805
9. Wu X, Lode M (2020) Language models are unsupervised multitask learners (summarization). OpenAI Blog 1:1–7
10. Liu C, Wu X, Yu M, Li G, Jiang J, Huang W, Lu X (2019) A two-stage model based on BERT for short fake news detection. In: Lecture notes in computer science (Including subseries Lecture notes in artificial intelligence and lecture notes in bioinformatics, vol 11776 LNAI, pp 172–183. https://doi.org/10.1007/978-3-030-29563-9_17
11. Jwa H, Oh D, Park K, Kang JM, Lim H (2019) exBAKE: automatic fake news detection model based on bidirectional encoder representations from transformers (BERT). Appl Sci 9:4062. https://doi.org/10.3390/app9194062
12. Vijjali R, Potluri P, Kumar S, Teki S (2020) Two stage transformer model for covid-19 fake news detection and fact checking. arXiv Preprint http://arxiv.org/abs/2011.13253
13. Kakwani D, Kunchukuttan A, Golla S, Gokul NC, Bhattacharyya A, Khapra MM, Kumar P (2020) IndicNLPSuite: Monolingual Corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In: Findings of EMNLP
14. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
15. Khanuja S, Bansal D, Mehtani S, Khosla S, Dey A, Gopalan B, Margam DK, Aggarwal P, Nagipogu RT, Dave S, Gupta S, Gali SCB, Subramanian V, Talukdar P (2021) Muril: multilingual representations for Indian languages. arXiv:2103.10730
16. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzman F, Grave E, Ott M, Zettle-moyer L, Stoyanov V (2019) Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116

A Review on Artificial Intelligence Techniques for Multilingual SMS Spam Detection

E. Ramanujam, K. Shankar, and Arpit Sharma

Abstract With social networks' increased popularity and smartphone technology advancements, Facebook, Twitter, and short text messaging services (SMS) have gained popularity. The availability of these low cost text-based communication services has implicitly increased the intrusion of spam messages. These spam messages have started emerging as an important issue, especially to short-duration mobile users such as aged persons, children, and other less skilled users of mobile phones. Unknowingly or mistakenly clicking the hyperlinks in spam messages or subscribing to advertisements puts them under threat of debiting their money from either the bank account or the balance of the network subscriber. Different approaches have been attempted to detect spam messages in the last decade. Many mobile applications have also evolved for spam detection in English, but still, there is a lack of performance. As English has been completely covered under natural language processing, other regional languages, such as Urdu and Hindi variants, have specific issues detecting spam messages. Mobile users suffer greatly from these issues, especially in multilingual countries like India. Thus, this paper critically reviews the artificial intelligence-based spam detection system. The review lists out the existing systems that use machine and deep learning techniques with their limitations, merits, and demerits. In addition, this paper covers the scope for future enhancements in natural language processing to efficiently prevent spam messages rather than detect spam messages.

E. Ramanujam (✉)
Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India
e-mail: ramanujam@cse.nits.ac.in

K. Shankar
Electronics and Instrumentation Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India
e-mail: shankar@ei.nits.ac.in

A. Sharma
Department of Computer Science and Engineering, Christ (Deemed to be) University, Bengaluru, India
e-mail: arpt.sharma@mtech.christuniversity.in

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_40

Keywords Deep learning · Ham · Machine learning · NLP · Phishing · Spam detection · Smishing · SMS · Spam

1 Introduction

The origin of smart devices has led to smart everywhere and everything such as smart homes, cities, and automation. As part of smart devices, smartphone also has a glorious growth and development and now becomes a user-friendly pocket device for the day-to-day activities. Smartphones facilitate communication, social media, entertainment, banking, health monitoring, etc. Though these devices have extraordinary technological and socioeconomic growth in everyday life, it creates increasing vulnerability to substantially and impenetrably diverse threats to the user, i.e., mobile devices have more vulnerability to attack than computers, desktops, or laptops. Short Text Messages (SMS) are one such crucial attacking agent (an in-built application of smartphone) through which hackers can easily attack the users and steal the information. It has become popular over time as SMS are substantially less expensive than other forms of communication.

As of April 2022, there are 6.648 billion smartphone users and 23 billion SMS messages are sent everyday worldwide. Almost 60% consumers check their SMS notifications within five minutes. A person sends an average of 72 text messages per day, and this has increased the volume of messages sent on an average to 9200% from the year 2010 [1]. Especially, in the business marketing, text messages had a high reach in connecting with the customers due to less cost and the reason that SMS be read by them immediately. Marketing SMS has an average conversion rate of 29%, whereas the marketing e-mail has only 3.26%, Facebook ads have 9.21%, and Google Ads have even worse of just 3.17% [1]. Thus, SMS have stepped into all the commercial industries for advertisements and offers, etc., [2]. Unfortunately, this leads to annoying tens of users with the spam SMS every day [1]. It is referred to as annoying, irrelevant, or unwanted text messages delivered using mobile networks [2]. Spam messages are mostly annoying to users. For instance "Enjoy unlimited banking with zero paperwork. Click http://kotk.in/297E0s to open 811 Digital Bank A/c. *T&C Apply", "Hello 9944XXXX16 Get the Big Jackpot Register today and get Rs. 5500 Reward. Play and Win Real Cash. http://a2fn.com/ljmahaizmn".

Spam messages have been an effective weapon for phishing by attackers to extract the personal, confidential, and financial data illegally without knowing to the user. Such spam messages have come from both domestic and international senders. In general, detecting smishing is a challenging task, as the attacker will share only minimal information. For instance, text messages in concise forms, smileys, idioms, abbreviations, etc., contain only a few smishing features. The spam detection has certain limitations with limited features and scarcity of real-time smishing dataset.

In the last decade, various researchers and mobile organizations have taken measures against spam message detection as quoted in [3–7]. However, a fine-grained filter system that is highly required with high throughput and performance for SMS

filtering systems. Spam filtering is not particularly effective, despite the fact that e-mails have been quite supportive of sophisticated spam filtering approaches. The reason behind this challenge is the manual feature engineering process used [10–13]. Deep learning methods have obviated this feature engineering process, and it has been a black box model which classifies the messages directly to spam or ham [14–17]. However, still there are certain challenges associated with multilingual spam SMS detection. Though, there are various review works already available in SMS spam detection such as [14–17] to list the research works. This paper critically reviews the research works that deals only with multilingual dataset and not only with English. This is the first time of such review which concentrates on multilingual dataset. This work lists out the publicly available spam message (smishing) datasets in various languages. Moreover, the methodologies have been categorized into feature, machine, and deep learning mechanism, quoted with its limitations and future scope for enhancement.

2 Benchmark Datasets

A quantitative analysis has been performed to collect benchmark datasets from various research/review papers [3–6] and given in Table 1. Protecting user's privacy in collecting a dataset is a major challenge. As a result, the majority of the SMS database is compiled using a smaller corpus of SMS text messages and is predicated on re-usability.

3 Categorization of Spam Detection Methods

The SMS spam detection methods are quantitatively analyzed from various literature works of research and mobile applications and categorized into feature engineering, machine learning (ML), and deep learning (DL) methods as shown in Fig. 1.

Table 1 Publicly available SMS corpus or datasets

Dataset	Total	Spam	Ham
SMS spam corpus v 0.1 small [18]	1084	82	1002
SMS spam Corpus V 0.1 Big [18]	1324	322	1002
Spam Collection V.1 [2]	5574	747	4827
Caroline Tag Campus [19]	450	NIL	450
Grumbletext [2]	425	425	NIL
Dublin Institute of Technology [20]	1353	1353	NIL
British English Corpus [21]	875	425	450

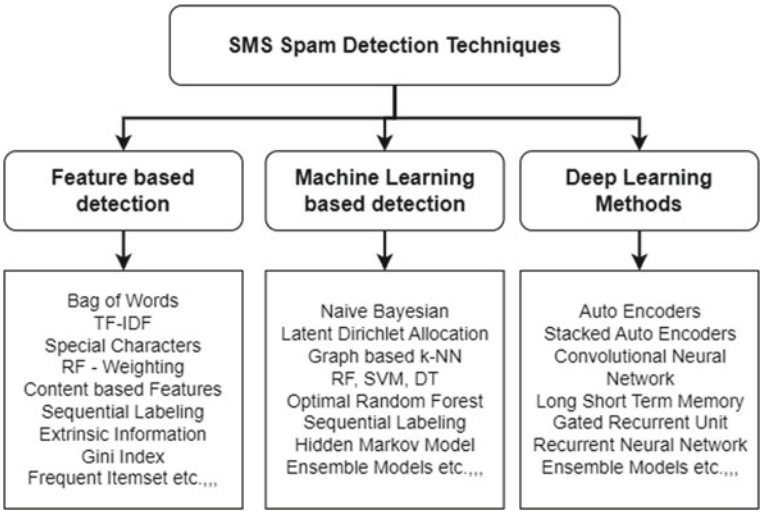


Fig. 1 Categorization of SMS Spam detection systems

3.1 Feature Engineering-Based Spam Detection

Feature engineering is extracting valuable and reliable features for efficient classification. The classifier majorly depends on the quality of features extracted during the feature engineering process. Some specific statistical techniques select efficient and quality features from the feature extraction termed feature selection process. In these SMS spam detection methods, the feature engineering process primarily deals with extracting features that depend on the natural language processing (NLP) model, as the SMS messages are entirely textual. The textual messages also contain numbers, digits, and letters combination. Thus, this section briefs the NLP-based feature engineering process that has shown better performance classified using machine learning classifiers.

Certain research works have utilized basic NLP processes to improve the SMS spam classification performance during 2010. However, Narayan and Saxena [22] have initially utilized a reverse engineering process with classification algorithms for spam detection. Considering the result, the proposed approach develops a two-level stacked classifier for SMS spam detection using traditional Bayesian e-mail filters. Research work [23] has proposed opinion-based spam detection that uses term frequency-inverse document frequency (*tf-idf*) and a bag-of-words (BoW) methods for feature extraction. The extracted features are then classified using Naïve Bayesian (NB) classifier. The research work [24] has compared the performance of *tf-idf* and relevance frequency term weighting methods to classify spam SMS using five different classifiers.

Karami et al. [25] have proposed latent content-based features to detect static spam messages. The efficacy of the extracted features has been experimented with

using multiple classification methods. Research work by [26] has proposed a method to preserve the sequential order and nature of text by the writing style for feature space generation and classification. In this approach, sequential label extraction, term clustering, extrinsic source information techniques have been used for feature extraction.

To analyze the efficiency of classification methods in the imbalanced dataset, Eshmawi and Nair [27] have studied the effect of Synthetic Minority Oversampling TEchnique (SMOTE) over SMS spam detection. SMOTE has improvised the classifier performance classifier in an imbalanced dataset. Research work [28] has utilized the GentleBoost algorithm, an ensemble method for classifying spam messages. Giannella et al. in [29] have developed a content-based Bayesian classification, a form of unsupervised and semi-supervised approach. It is a probabilistic generative model that estimates the parameters using the Gibbs sampling method. Ahmed et al. in [30] have proposed a similar semi-supervised learning approach that uses a frequent itemset for the feature selection from the SMS dataset and is classified using an ensemble machine learning approach.

The length of the SMS messages also plays a vital role in classifying spam messages. Chan et al. in [31] have presented feature re-weighting and good word attack for the SMS spam filtering process. This approach focuses on the length and weight of the message. Li and Zeng [32] have introduced a vector space model (VSM) that acts as an anti-spam filter to filter out spam messages based on the length of the messages. This approach leverages the messages to be spam if they contain more than 70 Chinese characters or over 140 English letters. The VSM model has been slightly modified in this approach to overcome the issues in SMS spam classification. Silva et al. [33] have proposed MDLText, a minimum description length (MDL)-based spam message detection technique. This technique has given more accurate results on large SMS (spam) dataset at computationally low cost.

Zainal and Jali [34] have studied the discriminating control of features based on their influence value in the SMS classification. This approach excavated the feature extraction technique for classifying the spam messages at the user end. One step further, Almeida et al. [35] have expanded the original and messy text messages using the normalization method to extract better features. Then, the lexicographic and semantic dictionaries were used for feature analysis and selection. Similarly, El-Alfy and AlHasan [11] have extracted a standard set of 11 features to classify spam messages. Performances of the extracted features are evaluated using standard SMS datasets using various machine learning classifiers.

To overcome certain drawbacks of supervised machine learning technique, Ma et al. [36] have developed intelligent spam detection using the message topic model. This method depends on the likelihood hypothesis of various features of textual messages. The extracted features are clustered as spam and non-spam messages using k-means clustering technique which eliminates the sparse problem by grouping spam messages based on inter- and intra-class similarity. Similarly, Nagwani and Sharaff [37] have utilized non-negative matrix factorization and k-means clustering technique for clustering of spam messages.

Though the unsupervised models have shown promising results, the performance of supervised models has been improved using varying features. Research work [38] has investigated the performance of models using BoW and word count with SVM and RF. The proposed model achieved 80–90% and 60–70% detection accuracy for BoW and word count, respectively. Jain et al. [39] have introduced a rule-based classification model that uses various rules on feature selection for the spam message classification. The rules verify the occurrence of each feature to appear in every spam and ham message. The proposed research work has extended its rule-based feature selection as reported in [40] to improvise classification performance.

Like feature selection and machine learning classifiers, certain automated machine learning tools are also available for classifying spam messages. Suleiman and Al-Naymat [41] have proposed a classifier based on the H_2O framework to compare with traditional learning algorithms. In addition, the H_2O framework-based classifier has been compared to RF and NB classifiers. Saeed [6] has investigated the performances of automated spam message filtering tool such as mljar, H_2O supervised AutoML, and tree-based pipeline optimization. The models are stacked to form an ensemble on the H_2O AutoML tool to achieve the best performance. Bosaeed et al. [42] have proposed a system that comprises NB, SVM, and NB multinomial classifiers. Five different pre-processing techniques were used along with the feature extraction method to classify the incoming and outgoing messages. The proposed system is compatible with executing in cloud, fog, and edge systems. Rojas-Galeano [43] has investigated that Google's BERT language model is sensitive to overcoming adversarial attacks (Mad-lib). In this proposed approach, the BoW and *tf-idf* substitution models are used for feature extraction, and the selected features are refined using the BERT model for classification.

In addition to the above techniques, Xia and Chen [44] have introduced a weighted feature algorithm for extracting features from spam SMS databases. The extracted features are selected using Hidden Markov Model (HMM), which is more suitable even for massive spam collection datasets. To classify the spam messages from smishing messages, Mishra and Soni [45] determine whether the URL in the message is legitimate. The five most deterministic features from the text messages have been extracted for classification using the backpropagation algorithm.

The existing technique mainly deals with the dataset as tabulated in Table 1. There is no specialized technique that analyzes or classifies multilingual SMS messages due to restrictions in data collection. Pham and Le-Hong [46] have proposed a first system to filter Vietnamese spam SMS. The model proposes an appropriate pre-processing method that enhances the accuracy of the dataset. Finally, complexity analysis has a significant impact on any algorithm. Thus, Waheeb et al. [47] has analyzed the time and space complexity of Artificial Neural Networks with the Scaled Conjugate Gradient method (ANN-SCG) to classify content-based SMS spam filtering with limited resources. The proposed approach uses a Gini Index metric to select features efficiently.

3.2 Machine Learning-Based Spam Detection

This section discusses the earlier works on classification algorithms works with baseline feature extraction process for spam detection. Research work by [48] has developed a mobile application, "SMSAssassin," to filter real-time spam messages. This mobile application uses NB and SVM classifiers for the classification of messages. Additionally, the crowd-sourcing mechanism in this approach updates the patterns and keyword that occurs newly for the efficient spam detection. Research work [49] has introduced an Optimum-Path forest classifier to filter spam context from ham SMS messages. Experimentation has shown more promising results than the traditional classification algorithms. To implement the system in real time, Bozan et al. [50] have proposed an expert system and a classification method as prototype software for the Jelly Bean version for real-time SMS spam detection. This approach uses SVM, NB, and k-NN classification methods to efficiently classify messages.

Joo et al. [51] used a conditional NB classifier to determine the probability of different words in the text messages. This approach considers the probability of messages for the classification. To analyze cross-platform spam detection, Ali and Maqsood [52] have developed a RF classifier trained using C# library for efficiency. In addition to the above techniques, Sethi et al. [53] have analyzed the various machine learning algorithms to classify spam messages in publicly available datasets. The authors of the proposed approach have also analyzed the performance of features on classification. Gupta et al. 2018 compare the traditional classifiers on various datasets, as reported in Table 1, using standard evaluation metrics such as precision, recall, accuracy, and CAP curve.

3.3 Deep Learning-Based Spam Detection

The traditional machine learning classifier has certain limitations and challenges in the feature engineering process. Thus, deep learning evolves into research as it has automatic and self-trained feature extraction. Though various deep learning techniques are available for the feature engineering process, only a limited number of research works have been carried out for SMS spam detection.

Al Moubayed et al. [54] have presented a novel approach to filter spam messages with minimum feature engineering on a set of labeled data samples. The topic modeling-based Latent Dirichlet allocation extracts the features and stacked denoising autoencoder classifies the data. Research work [55] has proposed a model to classify ten days of real data spam messages from Twitter. The Word Vector training model has been used to learn the sentence structure of each tweet and, in turn, train the deep learning model for binary classification. Lee and Kang [56] have explored a word embedding method, Continuous BoW (CBOW), for constructing feature vectors and classifying using a deep learning method. The feed-forward neural network method acts as classification model.

Uddin et al. [57] have introduced an SMS spam detection technique for the Bengali dataset (Bangladesh) using the LSTM and GRU, and the accuracy achieved is 99%. The proposed approach also demonstrates various activation functions and optimizers. Roy et al. [1] have employed CNN and LSTM models to classify textual messages. Experimentation was carried out on self-extracted benchmark dataset consisting of 747 spam and 4827 non-spam messages. CNN has achieved a remarkable accuracy of 99.44%. Ghourabi et al. [58] have proposed a model that integrates CNN with LSTM to form a hybrid deep learning model. Experimentation has been carried out with Arabic and English spam datasets. The intention of CNN-LSTM integration is to deal with bilingual messages. Performances are compared with machine and deep learning algorithms, and the hybrid CNN-LSTM model achieved 98.37% accuracy. Wei and Nguyen [16] proposed a lightweight model named lightweight GRU for SMS spam detection. The model incorporated the enhanced semantics from WordNet and external knowledge for better classification performance with a understanding of SMS text inputs.

Tong et al. [59] have proposed a capsule network that combines LSTM with attention mechanism to classify Chinese spam messages. The textual messages have been represented using a multi-channel structure and trained using a structural capsule for mining and classification of features. Karasoy and Balli [60] introduced a content-based SMS classification on Turkish language. Five different structural features and two features from Word2Vec, 45 features of word index value have been generated and classified using machine and deep learning models. In this approach, CNN has shown better classification accuracy. Sousa et al. [61] have embedded the SMS message into VSM model for spam detection. This approach relies on embedded patterns to classify spam from legitimate messages. Specific patterns are represented as skip grams for token generalization, allowing a distance greater than matched tokens in the text. The patterns are combined with the external networks that achieve the highest accuracy.

Liu et al. [62] have explored the possibility of a transformer model detecting the SMS messages as either spam or ham. Shaaban et al. [17] introduced a dynamic deep ensemble model that has a capacity to extract dynamic feature at low cost and complexity. The proposed model integrates the convolution, pooling layers for feature engineering process. The feature maps generated are classified using highly randomized trees also termed as modified random forest for classification of spam messages. In addition, the model involves boosting and bagging to improvise the performance of classification.

4 Scope for Future Enhancements

Though deep learning and machine learning techniques have evolved over the last decade and shown promising results in SMS spam detection techniques, there are still certain open challenges that require researchers to solve. Numbered (ordered) lists are easy to create:

1. Linguistic (Multi): The existing approaches have proposed various models for English, Spanish, Arabic, or single languages. Countries like India have varying numbers of regional languages, such as Tamil, Hindi, Marathi, Kannada, Telugu, etc. There needs a specific generic model to be designed for all the languages.
2. Dataset size: Due to the security and privacy of users' data, the size of the SMS dataset is significantly less in numbers. Specific datasets even do not have spam messages. This kind of dataset cannot analyze the performance of deep learning to a larger extent. Thus, the dataset has to be generated or grumbled in a larger size for complete analysis.
3. Data imbalance: The size of spam messages is still less in number compared to ham messages. Thus, a unique number of a balanced dataset has to be generated in the future for further deep analysis of any classification models.
4. Computational complexity: The complexity in analyzing the feature engineering process and training deep learning models is very high. In the future, the computational cost of the deep learning process has to be reduced using proper optimization techniques to improve the classification accuracy performance.
5. False positive rate: Most machine learning and deep learning techniques have shown a high false positive rate in the case of messages with URLs and idioms. It needs to be rectified with a proper understanding of features and layer construction in the deep learning model.

5 Conclusion

The availability of SMS facilities by mobile subscribers at a low cost has increased the intrusion of spam messages by attackers. Regarding e-mails, SMS are the most dedicated text-based communication channel, with higher response rates. This makes the attackers think wise in attacking mobile users through spam messages. Various spam message detection techniques have been proposed in the last decade. However, it has many limitations in handling multilingual spam data. Thus, a niche technique is highly required to detect spam messages with a reasonable accuracy scale. This paper critically reviewed various machine and deep learning-based techniques with their performances, limitations, and chances for future enhancements to novice researchers in this field.

References

1. Roy PK, Singh JP, Banerjee S (2020) Deep learning to filter SMS spam. Future Gener Comput Syst 102:524–533
2. Almeida TA, Hidalgo JMG, Yamakami A (2011) Contributions to the study of SMS spam filtering: new collection and results. In: Proceedings of the 11th ACM symposium on document engineering, pp 259–262

3. Gupta M, Bakliwal A, Agarwal S, Mehndiratta P (2018) A comparative study of spam SMS detection using machine learning classifiers. In: 2018 eleventh international conference on contemporary computing (IC3). IEEE, pp 1–7
4. Abayomi-Alli O, Misra S, Abayomi-Alli A, Odusami M (2019) A review of soft techniques for SMS spam classification: methods, approaches and applications. Eng Appl Artif Intell 86:197–212
5. Shafi'I MA, Abd Latiff MS, Chiroma H, Osho O, Abdul-Salaam G, Abubakar AI, Herawan T (2017) A review on mobile SMS spam filtering techniques. IEEE Access 5:15650–15666
6. Saeed W (2021) Comparison of automated machine learning tools for SMS spam message filtering. In: International conference on advances in cyber security. Springer, Singapore, pp 307–316
7. Gadde S, Lakshmanarao A, Satyanarayana S (2021) SMS spam detection using machine learning and deep learning techniques. In: 2021 7th international conference on advanced computing and communication systems (ICACCS), vol 1. IEEE, pp 358–362
8. Deshpande VP, Erbacher RF, Harris C (2007) An evaluation of Naïve Bayesian anti-spam filtering techniques. In: 2007 IEEE SMC information assurance and security workshop. IEEE, pp 333–340
9. Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10(5):1048–1054
10. Gupta V, Mehta A, Goel A, Dixit U, Pandey AC (2019) Spam detection using ensemble learning. In: Harmony search and nature inspired optimization algorithms. Springer, Singapore, pp 661–668
11. El-Alfy ESM, AlHasan AA (2016) Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm. Future Gener Comput Syst 64:98–107
12. Fu J, Lin P, Lee S (2014) Detecting spamming activities in a campus network using incremental learning. J Netw Comput Appl 43:56–65
13. Kim SE, Jo JT, Choi SH (2015) SMS spam filterinig using keyword frequency ratio. Int J Secur Its Appl 9(1):329–336
14. Jain G, Sharma M, Agarwal B (2019) Optimizing semantic LSTM for spam detection. Int J Inf Technol 11(2):239–250
15. Nguyen DT, Al Mannai KA, Joty S, Sajjad H, Imran M, Mitra P (2017) Robust classification of crisis-related data on social networks using convolutional neural networks. In: Eleventh international AAAI conference on web and social media
16. Wei F, Nguyen T (2020) A lightweight deep neural model for sms spam detection. In: 2020 International symposium on networks, computers and communications (ISNCC). IEEE, pp 1–6
17. Shaaban MA, Hassan YF, Guirguis SK (2022) Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. Complex Intell Syst, 1–13
18. Corpus v0.1 : http://www.esp.uem.es/jmgomez/SMSspamcorpus
19. Tagg C (2009) A corpus linguistics study of SMS text messaging. Doctoral dissertation, University of Birmingham
20. DIT : http://www.dit.ie/computing/research/resources/SMSdata/
21. BEC : https://mtaufiqnzz.wordpress.com/british-english-SMS-corpora/
22. Narayan A, Saxena P (2013) The curse of 140 characters: evaluating the efficacy of SMS spam detection on android. In: Proceedings of the third ACM workshop on Security and privacy in smartphones & mobile devices, pp 33–42
23. Patel R, Thakkar P (2014) Opinion spam detection using feature selection. In: 2014 international conference on computational intelligence and communication networks. IEEE, pp 560–564
24. Kural OE, Demirci S (2020) Comparison of term weighting techniques in spam SMS detection. In: 2020 28th signal processing and communications applications conference (SIU). IEEE, pp 1–4
25. Karami A, Zhou L (2014) Exploiting latent content based features for the detection of static sms spams. Proc Am Soc Inf Sci Technol 51(1):1–4

26. Serrano JM, Hernández Palancar J, Cumplido R (2014) The evaluation of ordered features for sms spam filtering. In: Iberoamerican congress on pattern recognition. Springer, Cham, pp 383–390
27. Eshmawi A, Nair S (2014) Semi-synthetic data for enhanced SMS spam detection: [using Synthetic Minority Oversampling TEchnique (SMOTE]). In: Proceedings of the 6th international conference on management of emergent digital ecosystems, pp 206–212
28. Akbari F, Sajedi H (2015) SMS spam detection using selected text features and boosting classifiers. In: 2015 7th conference on information and knowledge technology (IKT). IEEE, pp 1–5
29. Giannella CR, Winder R, Wilson B (2015) (Un/Semi-) supervised SMS text message SPAM detection. Nat Lang Eng 21(4):553–567
30. Ahmed I, Ali R, Guan D, Lee YK, Lee S, Chung T (2015) Semi-supervised learning using frequent itemset and ensemble learning for SMS classification. Expert Syst Appl 42(3):1065–1073
31. Chan PP, Yang C, Yeung DS, Ng WW (2015) Spam filtering for short messages in adversarial environment. Neurocomputing 155:167–176
32. Li W, Zeng S (2016) A vector space model based spam SMS filter. In: 2016 11th international conference on computer science & education (ICCSE). IEEE, pp 553–557
33. Silva RM, Almeida TA, Yamakami A (2017) MDLText: an efficient and lightweight text classifier. Knowl-Based Syst 118:152–164
34. Zainal K, Jali MZ (2016) A review of feature extraction optimization in SMS spam messages classification. In: International conference on soft computing in data science. Springer, Singapore, pp 158–170
35. Almeida TA, Silva TP, Santos I, Hidalgo JMG (2016) Text normalization and semantic indexing to enhance instant messaging and SMS spam filtering. Knowl-Based Syst 108:25–32
36. Ma J, Zhang Y, Liu J, Yu K, Wang X (2016) Intelligent SMS spam filtering using topic model. In: 2016 international conference on intelligent networking and collaborative systems (INCoS). IEEE, pp 380–383
37. Nagwani NK, Sharaff A (2017) SMS spam filtering and thread identification using bi-level text classification and clustering techniques. J Inf Sci 43(1):75–87
38. Etaiwi W, Awajan A (2017) The effects of features selection methods on spam review detection performance. In: 2017 international conference on new trends in computing sciences (ICTCS). IEEE, pp 116–120
39. Jain AK, Gupta BB (2018) Rule-based framework for detection of smishing messages in mobile environment. Procedia Comput Sci 125:617–623
40. Jain AK, Yadav SK, Choudhary N (2020) A novel approach to detect spam and smishing SMS using machine learning techniques. Int J E-Serv Mobile Appl (IJESMA) 12(1):21–38
41. Suleiman D, Al-Naymat G (2017) SMS spam detection using H_2O framework. Procedia Comput Sci 113:154–161
42. Bosaeed S, Katib I, Mehmood R (2020) A fog-augmented machine learning based SMS spam detection and classification system. In: 2020 fifth international conference on fog and mobile edge computing (FMEC). IEEE, pp 325–330
43. Rojas-Galeano S (2021) Using BERT encoding to tackle the Mad-lib attack in SMS spam detection. arXiv preprint arXiv:2107.06400
44. Xia T, Chen X (2021) A weighted feature enhanced Hidden Markov Model for spam SMS filtering. Neurocomputing 444:48–58
45. Mishra S, Soni D (2021) DSmishSMS—a system to detect smishing SMS. Neural Comput Appl, 1–18
46. Pham TH, Le-Hong P (2016) Content-based approach for Vietnamese spam SMS filtering. In: 2016 international conference on Asian Language Processing (IALP). IEEE, pp 41–44
47. Waheeb W, Ghazali R, Deris MM (2015) Content-based sms spam filtering based on the scaled conjugate gradient backpropagation algorithm. In: 2015 12th international conference on fuzzy systems and knowledge discovery (FSKD). IEEE, pp 675–680

48. Yadav K, Kumaraguru P, Goyal A, Gupta A, Naik V (2011) SMSAssassin: crowdsourcing driven mobile-based system for SMS spam filtering. In: Proceedings of the 12th workshop on mobile computing systems and applications, pp 1–6
49. Fernandes D, Da Costa KA, Almeida TA, Papa JP (2015) SMS spam filtering through optimum-path forest-based classifiers. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA). IEEE, pp 133–137
50. Bozan YS, Çoban Ö, Özyer GT, Özyer B (2015) SMS spam filtering based on text classification and expert system. In: 2015 23nd signal processing and communications applications conference (SIU). IEEE, pp 2345–2348
51. Joo JW, Moon SY, Singh S, Park JH (2017) S-Detector: an enhanced security model for detecting smishing attack for mobile computing. Telecommun Syst 66(1):29–38
52. Ali SS, Maqsood J (2018) .Net library for SMS spam detection using machine learning: a cross platform solution. In: 2018 15th international Bhurban conference on applied sciences and technology (IBCAST). IEEE, pp 470–476
53. Sethi P, Bhandari V, Kohli B (2017) SMS spam detection and comparison of various machine learning algorithms. In: 2017 international conference on computing and communication technologies for smart nation (IC3TSN). IEEE, pp 28–31
54. Al Moubayed N, Breckon T, Matthews P, McGough AS (2016) Sms spam filtering using probabilistic topic modelling and stacked denoising autoencoder. In: International conference on artificial neural networks. Springer, Cham, pp 423–430
55. Wu T, Liu S, Zhang J, Xiang Y (2017) Twitter spam detection based on deep learning. In: Proceedings of the Australasian computer science week multiconference, pp 1–8
56. Lee HY, Kang SS (2019) Word embedding method of sms messages for spam message filtering. In: 2019 IEEE international conference on big data and smart computing (BigComp). IEEE, pp 1–4
57. Uddin MM, Yasmin M, Khan MSH, Rahman MI, Islam T (2020) Detecting Bengali spam SMS using recurrent neural network. J Commun 15(4):325–331
58. Ghourabi A, Mahmood MA, Alzubi QM (2020) A hybrid CNN-LSTM model for SMS spam detection in Arabic and English messages. Future Internet 12(9):156
59. Tong X, Wang J, Zhang C, Wang R, Ge Z, Liu W, Zhao Z (2021) A content-based Chinese spam detection method using a capsule network with long-short attention. IEEE Sens J 21(22):25409–25420
60. Karasoy O, Ballı S (2022) Spam SMS detection for Turkish language with deep text analysis and deep learning methods. Arab J Sci Eng 47(8):9361–9377
61. Sousa G, Pedronette DCG, Papa JP, Guilherme IR (2021) SMS spam detection through skip-gram embeddings and shallow networks. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp 4193–4201
62. Liu X, Lu H, Nayak A (2021) A spam transformer model for SMS spam detection. IEEE Access 9:80253–80263

Fusion of LBP and Median LBP for Dominant Region Based Multimodal Recognition Using Imperfect Face and Gait Cues

K. Annbuselvi, N. Santhi, and S. Sivakumar

Abstract The paper provides a novel approach to recognize imperfect face and gait cues by using only the dominant region of such cues which contain more information than that possessed by the other regions. To enhance features in such dominant region, a fusion of Local Binary Pattern (LBP) and Median Local Binary Pattern (Median LBP) procedure is proposed. Initially, the given imperfect face and gait probe images are divided into six overlapped half regions. After partition, the dominant overlapped half regions of face and gait are selected by using content based image retrieval process. Subsequently, the features of dominant overlapped imperfect face and gait regions are enhanced by fusing the feature vectors obtained by using fusion of LBP and Median LBP methods. Next the Eigen feature vectors followed by Fisher's vectors are constructed using Principal Component Analysis (PCA) followed by Linear Discriminant Analysis (LDA) dimensionality reduction algorithms. The decisions from face and gait biometric systems are obtained separately by using Euclidean distance measure. Finally, the decisions of face and gait classifiers are fused at decision level for recognition using AND/OR rule. The method is verified on freely available ORL face and CASIA B gait datasets.

Keywords Dominant region · LBP · Median LBP · Principal Component Analysis (PCA) followed by Linear Discriminant Analysis (LDA) · Multimodal fusion at decision level

K. Annbuselvi (✉) · N. Santhi
V.V.Vanniaperumal College for Women, Virudhunagar, Madurai Kamaraj University, Madurai, Tamilnadu, India
e-mail: annbuselvi@vvvcollege.org

N. Santhi
e-mail: santhi@vvvcollege.org

S. Sivakumar
Cardamom Planters' Association College, Bodinayakanur, Madurai Kamaraj University, Madurai, Tamilnadu, India
e-mail: sivaku2002@yahoo.com

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_41

1 Introduction

Numerous multimodal approaches combining physical biometrics face and behavioral biometrics gait have been presented with the aim of investigating if such a combination will increase upon the rate of recognition methods which solely use a single cues [1]. The integration of these two biometrics also improves the responsiveness of the system with respect to change the distance among the camera and subject. As such, multimodal human recognition is mostly used for perfect images of face and gait data. In practice, lighting variations, poses, partial occlusions, etc. often result in imperfect face and gait images. Therefore, the biometric systems using imperfect face and gait probe images are an emerging research area. So this paper is focused to identify such face and gait imperfect probe cues with respect to increase the performance rate.

In Biometric system, the performance rate highly depends on features of biometric cues. LBP has come out as one of the powerful approach for biometric feature representation. The LBP enhances the local structures by thresholding over a 3×3 neighborhood associated to each pixel [2]. The modified LBP is being the choice of threshold value against the local median i.e., Median LBP [3]. In this, instead of gray value of the central pixel, local median is used as the threshold. This paper focused to fuse the features obtained from such LBP and Median LBP.

In this paper, the imperfect face and gait probe cues are recognized by using regions which contain dominant features, instead of considering the whole image. Firstly the given imperfect face probe and gait probe images are divided into six overlapped half regions. They are denoted as top, bottom, left, right, vertical center, and horizontally center. After partition, the dominant overlapped region is selected by using content based image retrieval process, where the region with highest variance is considered as dominant region. Later, the features of such dominant region are extracted by using LBP and Median LBP separately. Next, the proposed fusion of LBP and Median LBP in a 3×3 pixel window is applied to get a more robust feature vector. Next the feature extraction and dimensionality reduction algorithms PCA followed by LDA [4] are applied to construct and reduce Eigen feature vectors of face and gait separately. Then, decisions of face and gait biometric systems are obtained separately by using similarity distance measure i.e., Euclidean distance. Lastly, the recognition is done at decision level by integrating face and gait decisions.

The rest of the paper is organized as follows. Section 2 deals with the review of literature. Section 3 provides an outline of LBP, Median LBP, proposed fusion of LBP and Median LBP, PCA followed by LDA based Face and Gait Recognition using dominant regions of imperfect probe images. Section 4 offers results and discussion, and Sect. 5 gives the conclusion.

2 Literature Review

As far as an automatic biometric system using improper biometric cue is concerned, research on this topic appears to be very sparse and inconsistent. When images are imperfect, finding the dominant regions which contain comparatively more information than the other regions is a challenging task. In latest years, several approaches are presented to resolve such issues. To deal identification based on imperfect or partial images, Iwashita et al. [5] offer an approach to recognize gait biometric, where the human body is partitioned into blocks; subsequently, features are extracted from each blocks. Next, the similarity measure is applied between training template and the retrieved features to set consistent weights on each block. Finally, humans are identified by weighting and combining the similarities of all components.

Yet, in Bianconi et al. [6], observe the problem of learning a set of characteristic patterns from a LBP called the "dominant local binary pattern" (DLBP). The approaches to get the dominant patterns can either keep knowledge of the patterns labels or discard it. Finally, it is determining which option the best one. Furthermore, in Bamini and Kavitha [7], provided a method by means of dominant local binary patterns (DLBP) that reflects the greatest often happening patterns in facial images. Using this method, they determine the strongest dominant LBP pattern in the image based on the central and neighboring pixel values, LBP is determined.

Further in Annbuselvi et al. [2, 3] propose new approaches Median-LBPF and Median-LBPG to efficiently enhance the features of face and gait imperfect images for better recognition.

Rao et al. [8] presented a way to retrieve an image by dynamic dominant features of images. As a first step, the image is evenly divided into 8 partitions. Then each partition centroid (MPEG-7 "color bin") is chosen as the dominant color. Image textures are obtained using a GLCM and shape information is obtained by the gradient vector flow field. Combining and integrating image color and texture features with shape features provides a healthy set of features for searching images. Weighted Euclidean distances are used in obtaining similar images.

Yet in Huang et al. [9], offer a way to identify the dominant region of imperfect probe images by using content based image retrieval process, later the dominant features are extracted and enhanced using Median-LBPF and Median-LBPG. Then the dimensions of features are decreased by using PCA followed by LDA. For recognition, Euclidean similarity distance measure is used. Based on the work carried out on imperfect or partial face or gait images, our focus in this paper is to improve the recognition rates.

3 Materials and Methods

3.1 Local Binary Pattern (LBP)

LBP is an efficient pattern descriptor to define the local patterns of an image. The flowchart to compute LBP code is given in Fig. 1 and the corresponding procedure is given in Algorithm 1.

Algorithm 1: To Compute LBP Code

Input: A Gray scale image.

Output: An Enhanced input image with LBP codes.

1. For each pixel in the given image, select its eight neighbors in a 3 × 3 neighborhood.
2. Take the center pixel and set it as a threshold for its 3 × 3 eight neighbors.
3. The center pixel value is subtracted from each pixel in the 3 × 3 neighborhood, then it assigns 0 if the value is negative otherwise 1 to the pixel.
4. Now compute the LBP code by concatenating all of these binary codes in a clockwise direction, starting with the top-left one, generates a binary number, which is then labeled with the decimal value that corresponds to it.

An example for computation of LBP for a pixel window size 3 × 3 is given in Fig. 2.

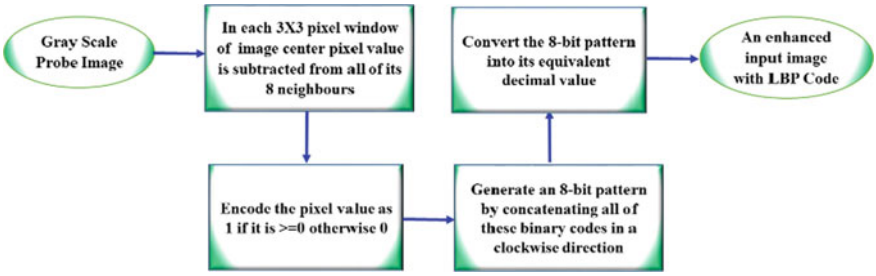


Fig. 1 Flowchart of LBP

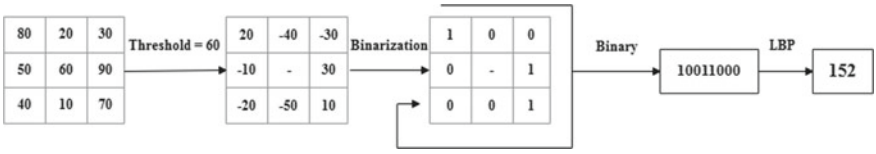


Fig. 2 An example for computation of LBP

3.2 *Median Local Binary Pattern (Median LBP)*

Median LBP is a modified form of LBP to extract more local features in certain situations. In Median LBP, the value of centering pixel is calculated by using the 3 × 3 neighborhoods median value including center pixel. The flowchart and corresponding algorithm are shown in Fig. 3 and Algorithm 2.

Algorithm 2: To Compute Median LBP

Input: A Gray scale image.

Output: An Enhanced input image with Median LBP codes.

1. For each pixel in the given image, select its eight neighbors in a 3 × 3 neighborhood including center pixel.
2. Calculate the value of centering pixel by finding the median value of all 3 × 3 neighborhoods including center pixel.
3. Then, compare the median intensity of all the pixels in the 3 × 3 neighborhood to all the pixels (including the center pixel). Then assigns 0 if the pixel value is less than the median value otherwise 1 to that pixel.
4. Now compute the Median LBP code by concatenating all of these binary codes in a clockwise direction, starting with the top-left one and ending with the center pixel, generates a binary number, which is then labeled with the decimal value that corresponds to it.

A sample computation of Median LBP for a pixel window size 3 × 3 is shown in Fig. 4.

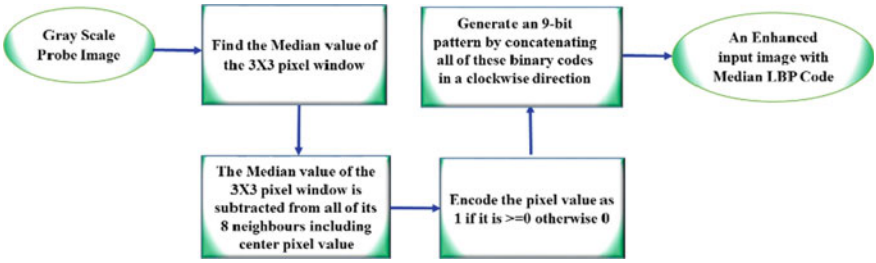


Fig. 3 Flowchart of Median LBP

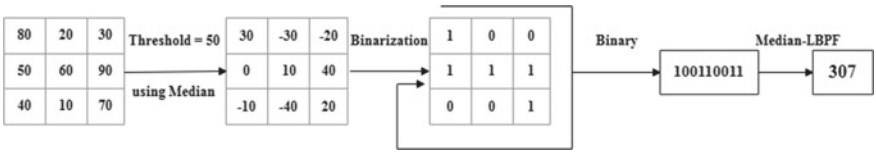


Fig. 4 Sample computation of Median LBP

3.3 *Fusion of LBP and Median LBP*

The proposed fusion of LBP and Median LBP is integrating the feature vectors obtained by LBP given in Sect. 3.1 and Median LBP given in Sect. 3.2 in order to acquire an additional vigorous feature vector. The resulting feature vector is got by averaging the LBP and Median LBP codes by using the following formula.

$$\text{Fusion (LBP, Median LBP)} = \frac{\text{LBP features} + \text{Median LBP features}}{2} \quad (1)$$

The procedure is given in Algorithm 3 and the flowchart is given in Fig. 5.

Algorithm 3: Fusion (LBP, Median LBP)

Input: A Gray scale image.

Output: An Enhanced input image with fused LBP and Median LBP features.

1. Obtain LBP features by using Local Binary Pattern.
2. Obtain Median-LBP features by using Median Local Binary Pattern.
3. Fuse LBP and Median-LBP features using the formula (1) to obtain a more discriminative robust feature vector.

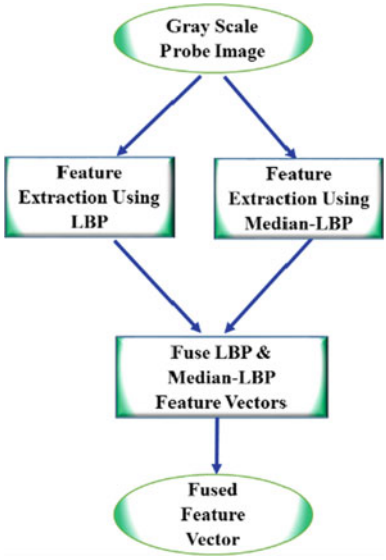


Fig. 5 Flowchart of fusion of LBP and Median LBP

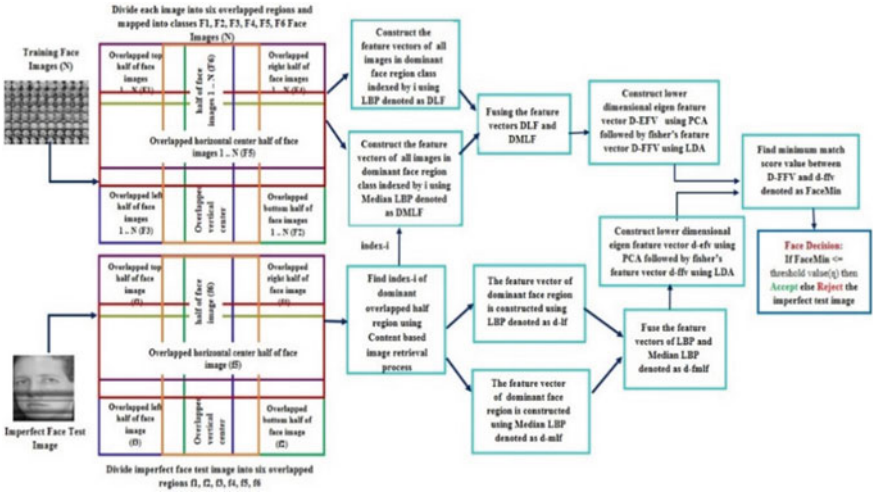


Fig. 6 Dominant region based imperfect face classifier

3.4 Dominant Region Based Imperfect Face and Gait Recognition Using Fusion of LBP + Median-LBP Based PCA Followed by LDA

The Dominant Region based Imperfect face/gait classifier comprises face/gait training dataset and testing probe images partition, dominant imperfect face/gait probe image region and corresponding dominant class in the training face/gait dataset selection, face/gait features extraction using LBP and Median LBP, dimensionality reduction using PCA followed by LDA [10] classification using Euclidean distance and finally Imperfect face/gait probe decision w.r.t. threshold value. The framework of the method is given in Figs. 6 and 7 and the procedure is given in Algorithm 4.

Algorithm 4: Recognition of Imperfect Face and Gait Probe Images

Input:

Face Biometric System:

Perfect Face Trained Dataset, A Gray scale Imperfect Face probe image.

Gait Biometric System:

Perfect Gait Silhouette Trained Dataset, A Gray scale Imperfect Gait Silhouette probe image.

Output:

Dominant Region based Imperfect Face and Gait Recognition based on LBP and Median LBP based PCA followed by LDA.

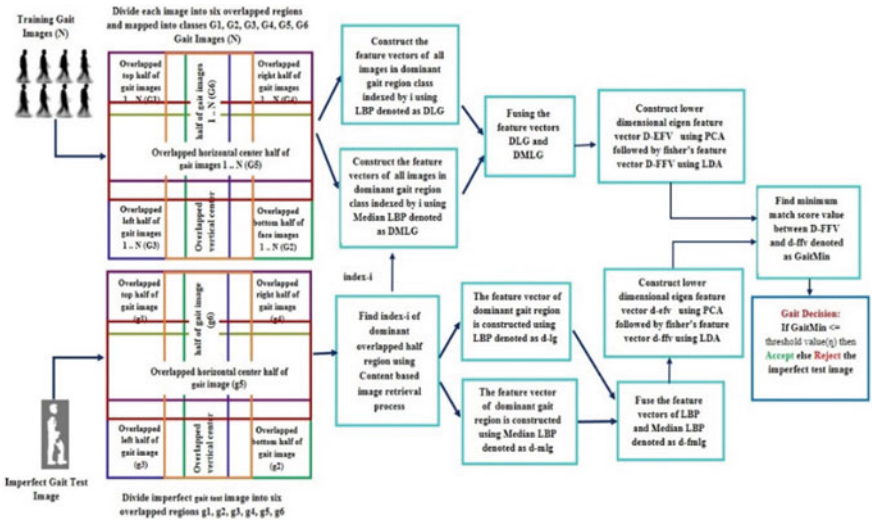


Fig. 7 Dominant region based imperfect gait classifier

Steps:

1. Restructuring of face and gait training datasets:
   (a) Each face/gait image in face/gait datasets is partitioned into six overlapped regions as top, bottom, left, right, vertical center, and horizontal center.
   (b) Construct six classes of overlapped regions of face/gait images as top, bottom, left, right, vertical center, and horizontal center where all top, bottom, left, right, vertical center, and horizontal center regions are mapped into the corresponding classes.
2. Restructuring of given imperfect face/gait probe images:
   (a) Partition the imperfect probe face/gait images into six overlapped regions denoted as top, bottom, left, right, vertical center, and horizontal center.
3. Selection of dominant region of imperfect face/gait probe region:
   (a) Select dominant overlapped face/gait region named as index-i by applying conventional content based image retrieval procedure as discussed in our paper [11].
4. Selection of dominant class from the face/gait training datasets:
   (a) Using the index-i, select the corresponding face/gait class from the restructured face/gait training dataset.
5. Face/gait features extraction using fusion of LBP + Median LBP based PCA followed by LDA.

(a) For Dominant region of Imperfect Face/Gait testing phase:
   (1) Enhance the features of dominant face/gait region of probe images indexed by i using LBP method (d-lf/d-lg) and Median LBP method (d-mlf/d-mlg) discussed in our paper [11] separately.
   (2) Fuse the dominant LBP and Median LBP based face/gait feature vectors d-lf/d-lg and d-mlf/d-mlg to obtain d-fmlf/d-fmlg using the method given in Sect. 3.3.
   (3) Subsequently construct lower dimensional face/gait Eigen feature vector using PCA followed by Fisher's feature vector using LDA for better face/gait classification.

(b) For Dominant class of Imperfect Face/Gait training phase:
   (1) Enhance the features of each images of dominant face/gait class indexed by i using LBP method (D-LF/D-LG) and Median LBP method (D-MLF/D-MLG) discussed in our paper [11] separately.
   (2) Fuse the feature vectors of dominant class D-LF/D-LG and D-MLF/D-MLG to obtain D-FMLF/D-FMLG using method discussed in Sect. 3.3.
   (3) Subsequently construct lower dimensional face/gait Eigen feature vector using PCA followed by Fisher's feature vector using LDA for better face/gait classification.

6. Dominant region based imperfect face/gait classification:
   (a) Find match score values by applying Euclidean Similarity/Distance Classifier to find distance measures between dominant class indexed by i of face/gait feature set with given dominant overlapped region of imperfect face/gait probe image.
   (b) Identify the imperfect face image with minimum score value (FaceMin) and Imperfect gait image with minimum score value (GaitMin) from face and gait training datasets for which, the Euclidean Distance of the corresponding Imperfect face and gait probe images is minimum.

7. LBP and Median-LBP based dominant region based imperfect face and gait decision:
   (a) In this Dominant Region based Imperfect Face and Gait Recognition system based on LBP + Median-LBP and PCA followed by LDA, the face decision and gait decision are obtained using match score values of FaceMin and GaitMin.

Face decision:

Accept (Genuine) $\rightarrow$ FaceMin $<=$ threshold (η).
Reject (Imposter) $\rightarrow$ FaceMin $>$ threshold (η).

Gait decision:

Accept (Genuine) $\rightarrow$ GaitMin $<=$ threshold (η).
Reject (Imposter) $\rightarrow$ GaitMin $>$ threshold (η).

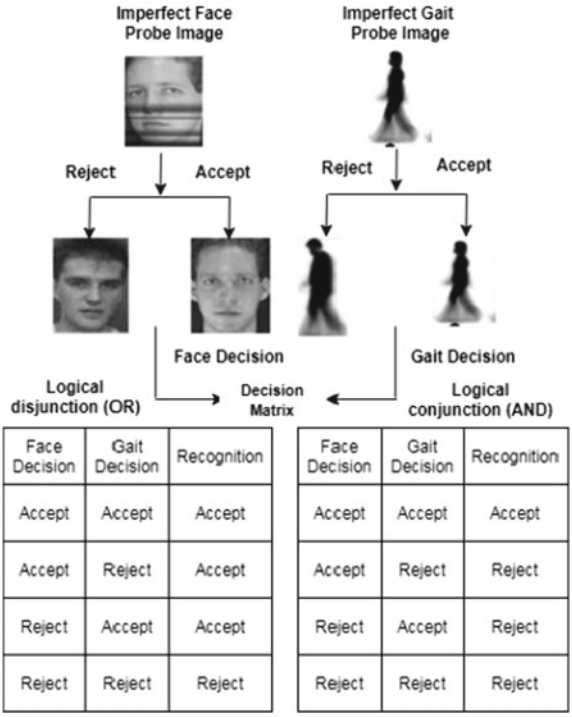


Fig. 8 Fusion of imperfect face and gait decisions

8. Integrating face and gait decisions at decision level:
   (a) Imperfect Face and Gait decisions are fused at decision level as discussed in paper [11]. The framework of face and gait decision fusion is shown in Fig. 8.

4 Results and Discussion

The offered method is tested on openly available ORL face [12] and CASIA gait [13] datasets. We report a whole set of experiments using various portions of imperfect face and gait biometrics for recognizing a person. To recognize such imperfect biometric cues, an improper biometric cue dataset was made with illumination variant, partial images with portion of face and gait images.

During testing, datasets with 50 face and gait correct images are equivalent to 10 persons by considering five images for each person and probe imperfect face and gait data sets with 20 images by considering two images for each person. With respect to the offered method, each image is divided into six overlapped half regions. As the result, the training data sets contain $50 \times 6 = 300$ images and are assembled into respective six regions classes. Finally, there are 50 overlapped half images in each

class. In testing phase also the given imperfect probe images are divided into six overlapped half regions.

Recognition rates are assessed to measure the performance of the proposed methods. They are calculated in percentage using total number of images recognized against total number of images available in the dataset. The rates of recognition of face classifier are given in Table 1 and visualized in Fig. 9. Similarly Table 2 shows the results of gait biometric classifier and is pictorially presented in Fig. 10. Likewise Table 3 shows rates of recognition of multimodal classifier results at decision level using AND and OR respectively and visually presented in Fig. 11.

As can be seen from the graph, the dominant multimodal classifier based on regions at decision level fusion by logical OR does fine with rate of recognition 98% compared to the result obtained with our previous method Median LBP [11]. The proposed methods show that the recognition rates of the multimodal system using imperfect probe images are near to system using perfect probe images. The offered approach also has the merit of decreasing the feature size by selecting dominant regions. This also reduces computation time and memory requirements.

Table 1 Results of perfect and imperfect dominant region based face classifier

Methods used	Number of features				
	20	40	60	80	100
	Recognition rate (%)				
Perfect face (Median LBP + PCA followed by LDA)	78	81	88	94	99.3
Imperfect face (LBP + PCA followed by LDA)	71	74	76	81	84
Imperfect face (Median LBP + PCA followed by LDA)	74	79	82	87	90
Imperfect dominant region based face (Median LBP + PCA followed by LDA)	80	83	88	90	93
*Imperfect dominant region based face (Fusion of LBP and Median LBP + PCA followed by LDA)	84	87	91	92	94

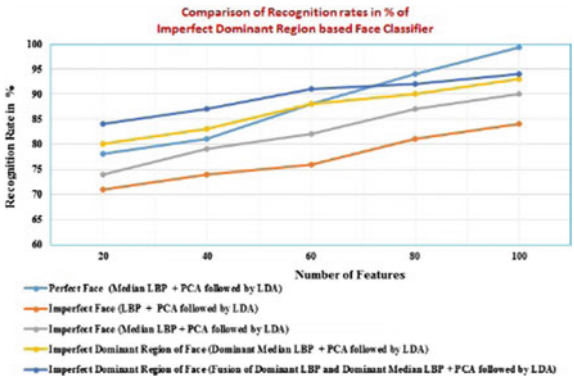


Fig. 9 Comparison of recognition rates in % of imperfect dominant region based face classifier

Table 2 Results of perfect and imperfect dominant region based gait classifier

Method used	Number of features				
	20	40	60	80	100
	Recognition rate (%)				
Perfect gait (Median LBP + PCA followed by LDA)	68	74	85	92	97.3
Imperfect gait (LBP + PCA followed by LDA)	64	70	74	79	84
Imperfect gait (Median LBP + PCA followed by LDA)	69	73	75	82	87
Imperfect dominant region based gait (Median LBP + PCA followed by LDA)	74	79	84	86	91
*Imperfect dominant region based gait (Fusion of LBP and Median LBP + PCA followed by LDA)	78	81	85	88	91.5

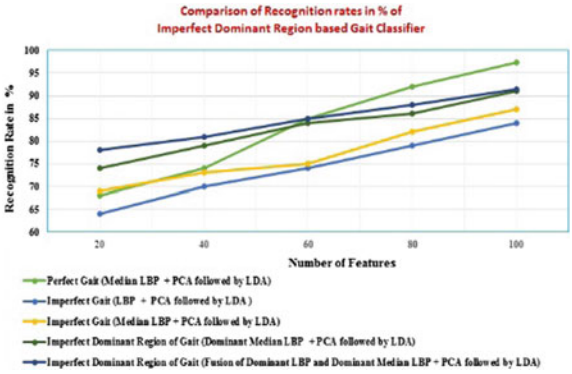


Fig. 10 Comparison of recognition rates in % of imperfect dominant region based gait classifier

Table 3 Results of perfect and imperfect dominant region based face and gait multimodal classifier fused at decision level using logical AND and logical OR

Methods used	Recognition rate (%)	
	Logical AND	Logical OR
Perfect Face + Gait (Median LBP + PCA followed by LDA)	98	98.5
Imperfect Face + Gait (LBP + PCA followed by LDA)	90	92
Imperfect Face + Gait (Median LBP + PCA followed by LDA)	92	94
Imperfect Dominant Region based Face + Gait (Median LBP + PCA followed by LDA)	92.3	94
*Imperfect Dominant Region based Face + Gait (Fusion of LBP and Median LBP + PCA followed by LDA)	92.5	94.3

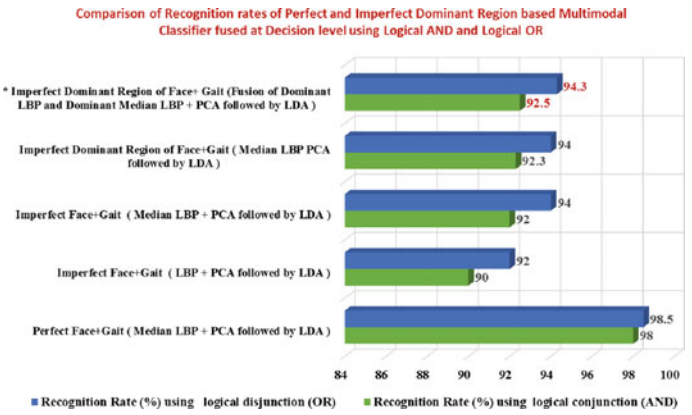


Fig. 11 Comparison of recognition rates in % of perfect and imperfect dominant region based face and gait multimodal classifier

5 Conclusion

This paper provides promising methods to recognize imperfect face and gait test images by utilizing a region which comprises the dominant features and integrates their results at decision level. Even if 75% of the test images are corrupted, it can still recognize humans. The dominant region is identified by using content based retrieval process and the LBP and Median LBP features of that region are fused to get more robust feature vectors. By exploiting the features of LBP and Median LBP, we can achieve 94.3% of recognition rate in dominant region based multimodal classifier fused at decision level using logical disjunction. The results also prove that the recognition rate can be improved by increasing number of features. In future, the work may be extended by considering two or more dominant regions of imperfect face and gait images to produce 100% results.

References

1. Annbuselvi K, Santhi N (2017) Intelligences of fusing face and gait in multimodal biometric system: a contemporary study. Int J Comput Sci Trends Technol (IJCST) 5(2):516–522. ISSN: 2347–8578
2. Annbuselvi K, Santhi N, Sivakumar S (2022) A competent multimodal recognition using imperfect region based face and gait cues using Median-LBPF and Median-LBPG based PCA followed by LDA. Mater Today Proc 62(Part 7):4869–4879. https://doi.org/10.1016/j.matpr.2022.03.505
3. Annbuselvi K, Santhi N, Sivakumar S (2022) Region based feature fusion of imperfect face and gait cues for human recognition using Median-LBPF and Median-LBPG based PCA followed by LDA. Int J Eng Comput Sci 11(1):25483–25492. https://doi.org/10.18535/ijecs/v11i01.4649. ISSN: 2319–7242

4. Annbuselvi K, Santhi N (2018) Role of feature extraction techniques: PCA and LDA for appearance based gait recognition. Int J Comput Sci Eng 6(4):38–44. E-ISSN: 2347–2693
5. Iwashita Y, Uchino K, Kurazume R (2013) Gait-based person identification robust to changes in appearance. Sensors 13:7884–7901. https://doi.org/10.3390/s130607884
6. Bianconi F, González E, Fernández A (2015) Dominant local binary patterns for texture classification: Labelled or unlabeled. Pattern Recognit Lett 65:8–14. https://doi.org/10.1016/j.patrec.2015.06.025
7. Bamini AA, Kavitha T (2010) Dominant local binary pattern based face feature selection and detection. Int J Eng Technol 2(2):77–80
8. Rao MB, Rao BP, Govardhan A (2011) Content based image retrieval using dominant color, texture and shape. Int J Eng Sci Technol (IJEST) 3(4):2887–2896. ISSN: 0975–5462
9. Huang D, Shan C, Ardabilian M, Wang Y, Chen L (2011) Local binary patterns and its application to facial image analysis: a survey. IEEE Trans Syst Man Cybern Part C Appl Rev 41(6):765–781
10. Santhi N, Annbuselvi K, Sivakumar S (2018) Performance analysis of feature extraction techniques: PCA and LDA for face recognition. Int J Eng Res Comput Sci Eng 5(3):289–294. ISSN (Online) 2394–2320
11. Santhi N, Annbuselvi K, Sivakumar S (2022) Multimodal recognition using dominant region of imperfect face and gait cues using Median-LBPF and Median-LBPG based PCA followed by LDA. In: 7th international conference on engineering research and innovations (ICERI-2022). ISBN: 978–93–92105–03–6
12. The Database of Faces (ORL Face Database) (2002) In: AT&T Laboratories Cambridge. Retrieved from http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
13. CASIA Gait Database. Retrieved from http://www.sinobiometrics.com

A Deep Convolutional Neural Network for Breast Cancer Detection in Mammograms

B. Naga Jagadesh, L. Kanya Kumari, and Akella V. S. N. Murthy

Abstract Breast cancer is the most prevalent ailment among women worldwide. The screening for detecting breast cancer is mammograms. The anticipated methodology uses preprocessing, feature extraction and classification. The main objective is early detection of the diseases so that the lifetime can be increased. A popular deep convolutional neural network (DCNN) model is used for mammogram classification that classifies into normal or abnormal. Initially, the mammograms are preprocessed using contrast limited adaptive histogram equalization (CLAHE). These images are fed to the DCNN model which contains five convolutional layers that use rectified linear unit (ReLU) as the activation function, and max-pooling is used to select the best features. The fully connected layer uses softmax as the activation function and classifies the selected features, and these are done with several optimizers, namely stochastic gradient descent (SGD), RMSprop, and Adam. The experiments are done on two benchmark datasets, namely the Mammographic Image Analysis Society (MIAS) and Digital Database for Screening Mammography (DDSM), and performance is measured with sensitivity, specificity, and accuracy. The results show that DCNN with SGD optimizer gives better results compared to the traditional methods.

Keywords Deep convolutional neural networks · Contrast limited adaptive histogram equalization · Preprocessing · Classification · Breast cancer

B. N. Jagadesh (✉)
School of Computer Science and Engineering, Vellore Institute of Technology-Andhra Pradesh, Amaravathi, Andhra Pradesh, India
e-mail: nagajagadesh@gmail.com

L. K. Kumari
Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India

A. V. S. N. Murthy
Department of Mathematics, SAS, Vellore Institute of Technology, Vellore, Tamil Nadu, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_42

1 Introduction

Breast cancer (BC) is the riskiest disease that causes death among women. It is triggered by abnormal cell growth which consequences the formation of tumors and may lead to illness. The symptoms of breast cancer are changes in breast shape, lumps in the breast, and nipple discharge. These breast cells unroll to further portions of the body makes a risk to the patient. So, BC detection is needed to evade the spreading of cancer. By detecting this disease in its early stages, the survival rate may be increased. Several imaging modalities are available such as mammograms, magnetic resonance imaging (MRI), computed tomography (CT-scan), and thermograms are mostly used for breast cancer detection [1].

There is no prevention technique for BC, but the survival rate in women can be improved by early detection. It is also crucial to detect BC in the early stages, and a self-breast test is also important to find irregularities before the tumor gets into the malignant stage [2]. Mammograms are the best screening modality to detect breast cancer in its early stages. It helps in masses and micro-calcification detection. Outstanding to the less cost and high sensitivity, mammograms are helpful for screening. It is the furthermost consistent identification technique that has low radiation compared to other modalities [3, 4].

Mammograms illustrate the masses, and the outlines are oval, irregular, and lobular that can be ill-defined, circumscribed, and speculated [5]. Artificial intelligence (AI)-based CAD system is used to provide good accuracy for early detection of BC [6]. In recent days, deep convolutional neural networks (DCNNs) are playing an important role in the detection of BC in its early stages [7]. DCNN is a classifier that uses machine learning algorithms that analyzes the mammograms and classifies them into benign or malignant. Depending on the classification finished by DCNN, the physicians predict the BC in the early stages which help physicians and patients in the early stages. The main aim is to design a novel approach to detect BC. To achieve the objective, we analyzed the existing CNN architectures and compared them with the proposed method. The proposed framework works in the following phases:

1. The mammograms are collected from publicly available datasets that contain cancerous and non-cancerous images.
2. Design and development of a mechanism that classifies the breast masses into cancerous or non-cancerous images.
3. The proposed method is compared with the existing CNN algorithms and the proposed method achieved good accuracy in mammogram classification.

This article is structured as follows: Sect. 2 deliberates the related work, Sect. 3 signifies the dataset, Sect. 4 designates the proposed methodology, Sect. 5 concentrates on the experimental results and performance analysis, and lastly presented the conclusion and future scope.

2 Related Work

Several authors have performed research in mammogram classification that classifies the images into benign or malignant.

A CAD system was designed to classify the mammograms. A DCNN was developed to extract the features in mammograms, and support vector machine (SVM) was in fully connected layer. The experiment was done on the DDSM dataset and obtained a good result [3]. Mammograms were classified using the VGG16. This model outperformed well compared to EfficientNet, AlexNet, and GoogleNet which helps radiologists [8].

A faster R-CNN was used to identify the abnormalities in mammograms and classified them into benign or malignant [9]. A multi-layer deep learning architecture was used to extract the features from mammogram images and used CNN-based classifier to classify into benign or malignant to detect breast cancer [10]. A novel recurrent neural network (RNN) was used to identify and classify the mammograms. The features were retrieved by RNN, and the hyperparameters were tuned by animal migration optimization (AMO) algorithm that uses a softmax classifier. The performance was evaluated by f-score, sensitivity, precision, and accuracy [11].

A CAD system was designed to reduce the false-positives rate by using modified entropy whale optimization algorithm. A fine-tuned MobilenetV2 was applied for simulation. The maximum accuracy was obtained for the MIAS dataset [12]. A deep learning model based on CNNs was designed for mammogram classification. The experiments were done on the Image Retrieval in the Medical Application (IRMA) dataset, and the results were assessed using tenfold cross-validation on simple logistics, SVM, and KNN and procured better results [13].

A DCNN with transfer learning was used for breast cancer detection. The VGG-16 network was used for mammogram classification that classifies it into benign or malignant. Results show that VGG-16 provides better results and evaluated on the MIAS dataset and obtained 82.5% accuracy [14]. A deep learning algorithm was designed to predict breast cancer with sensitivity of 87%, specificity of 77.3% [15].

A novel hybridized CNN model was designed for early detection of breast cancer. This model used ResNet50, and evaluation was done on two datasets, namely MIAS and DDSM. The results showed that an Improved Marine Predators Algorithm with ResNet50 gives the better results [16]. A CNN architecture was designed for mammogram classification. Deep learning-based computer-assisted systems assist the physicians in early detection. The performance was measured by accuracy, sensitivity, and specificity as 92.84%, 92.3%, and 96.72% respectively [17]. Four deep learning models Inception V4, DenseNet121, ResNet-164, and VGG-11 were used to detect malignancy in the breast. The fuzzy ensemble techniques and advanced ensemble strategies were used. The Inception V4 ensemble model with fuzzy rank obtained the better results in breast cancer detection [18].

A novel approach was proposed to improve the life time of women. The features were extracted based on CNN, and optimal features were selected using principal component analysis (PCA) and classified using KNN. The performance of the model

was measured by tenfold cross-validation and used MIAS dataset for evaluation and obtained the good results [19]. A deep CNN was proposed to extract the features from mammograms using AlexNet. A fine-tuned parameters were calculated with different optimizers and learning rates that classify into malignant and benign with DDSM dataset. The classification was done with SVM and extreme learning machine and procured the better results [20].

A novel model was designed for mammogram classification. The DDSM dataset was considered for the model evaluation. The results sensitivity 86.7%, specificity 96.1%, and AUC of 0.95 were obtained from experiments. The findings signify the automatic deep learning algorithms which will give better results on mammograms and to reduce the false positives and false negatives [21].

3 Methodology

3.1 Dataset

Mammographic Image Analysis Society (MIAS)
The MIAS dataset contains 322 mammogram images with 207 images being normal and 115 images being malignant [22]. Each image is of size 1024 × 1024 dimension. These images are retrieved in medio-lateral view (MLO), and these are digitized by Joyce-Loebl scanning microdensitometer with linear response ranging from 0 to 3.2. The MIAS mammogram images are represented in Fig. 1.

Digital Database for Screening Mammography (DDSM)
The mammogram dataset is considered from another database, namely Digital Database for Screening Mammography (DDSM). It contains 1566 patients' mammograms that contain two different views of the breast called craniocaudal (CC) and medio-lateral oblique (MLO). These images are also grayscale images [23]. The example DDSM images are represented in Fig. 2.

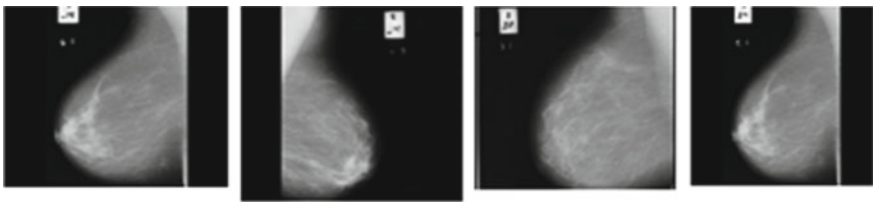


Fig. 1 Sample images-MIAS

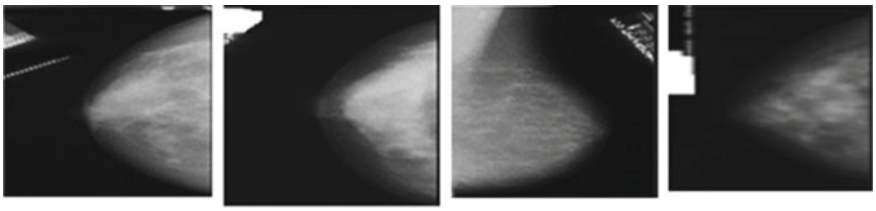


Fig. 2 Sample images -DDSM

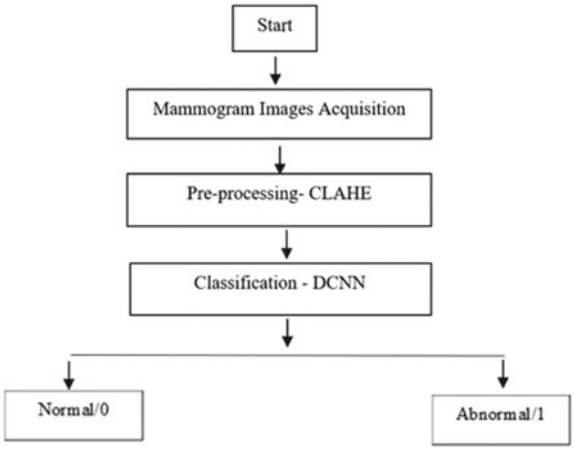


Fig. 3 Proposed methodology

3.2 Proposed Methodology

The proposed methodology was applied to the benchmark datasets MIAS and DDSM. The different phases of the proposed methodology are demonstrated in the following Fig. 3 and followed by a detailed discussion of several stages.

Preprocessing
This phase involves the preprocessing of mammograms for better feature extraction. These images contain low contrast and more noise which makes tumor detection difficult. The contrast of the mammogram is increased based on CLAHE that improves the contrast of each pixel.

Data Augmentation
Data augmentation is a procedure that makes the new training data from the available data. This is done by applying transformations using rotation. The images in the datasets are 1024 × 1024 in size which makes the training not feasible. So, these images' size is reduced to 64 × 64 for fast processing. These are augmented through rotating by applying transformations from 0 to 360 °. The angles used for transformation in our research are 90, 180, and 270 °.

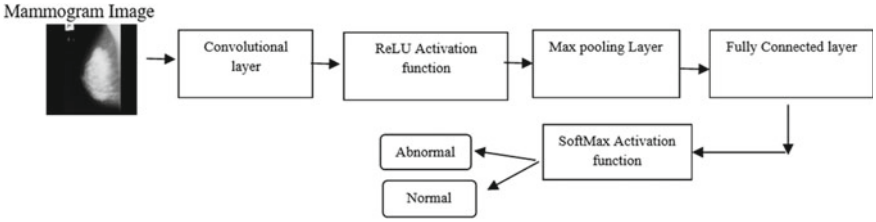


Fig. 4 Proposed DCNN architecture

Deep Convolutional Neural Networks (DCNN)

The deep convolutional layers are the most prevalent type of deep learning framework that is useful for medical imaging analysis. The architecture contains mainly a convolutional layer, a pooling layer, and a fully connected layer [24]. A convolutional layer is the main constituent of DCNN. This layer contains the weights that are called filters or kernels. The input is split into small components called receptive fields which are multiplied by the filter to generate the output. The distance between the filters is called stride which is less than the filter size. The pooling layer is downsampling the spatial dimension of the input. This layer's main objective is to reduce the spatial size of the representation and also helps in the reduction of computations.

The max-pooling layer is mostly used layer that computes the maximum value from the input. It is used for reduction in the dimensions of feature maps. It also reduces the computation time. This builds the network more powerful in the position of the features. The fully connected layer is the multi-layer perceptron that employs the softmax as the activation function in the output layer. The neurons present in this layer have fully connected to all activations in the previous layer. The main aim of this layer is to classify the image using the high-level features which are extracted from previous layers.

The proposed framework contains five convolutional layers. The initial layer is to find the high-level features like intensity, texture, and color. The next step is to move these features to ReLU activation function which is used for element-wise operation and sets the negative pixel values to 0. It gives nonlinearity to the model, and it is called rectified feature map. The next layer is to find the small features that have different shapes and sizes. The final layer combines entire features that are calculated from the previous layers and are used for the classification of mammograms into normal or abnormal. The proposed DCNN model is represented in Fig. 4.

4 Experimental Results and Discussion

Our research is to design a novel computer-aided diagnosis (CAD) model that detects breast cancer. The main objective is for mammograms classification that classified into normal or abnormal. Initially, the images are resized into 64×64 to reduce the computational complexity. The learning rate is considered as 0.01, and the number

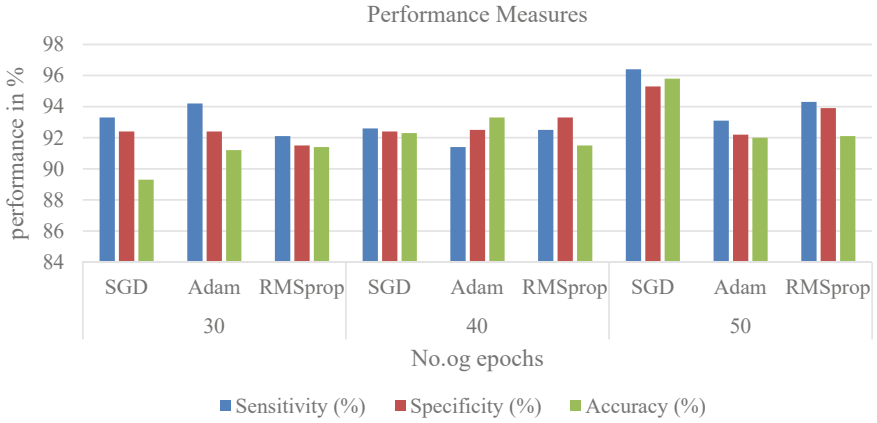


Fig. 5 Graphical representation of performance measures for MIAS dataset

of epochs considered is 100. The experimentations are done with several optimizers like RMSprop, SGD, and Adam with PC specifications of a minimum of 8 GB RAM. With augmentation, 2000 samples are created that are divided into 70% training and 30% as testing. The performance is measured by confusion matrix parameters like true positives (TPS), true negatives (TNS), false positives (FPS), and false negatives (FNS). The performance metrics used are sensitivity, specificity, and accuracy [25]. The results are represented in Fig. 5.

$$\text{Sensitivity} = \frac{\text{TPS}}{\text{TPS} + \text{FNS}} \tag{1}$$

$$\text{Specificity} = \frac{\text{TNS}}{\text{FPS} + \text{TNS}} \tag{2}$$

$$\text{Accuracy} = \frac{\text{TPS} + \text{TNS}}{\text{TPS} + \text{FPS} + \text{TNS} + \text{FNS}} \tag{3}$$

where TPS is true positives, FPS is false positives, TNS is true negatives, and FNS is false negatives.

The experimental results are represented in following Table 1 for MIAS dataset.

The experimental results are represented in following Table 2 for DDSM dataset.

By observing the above results, the classification of mammograms is done with Deep CNN gives better results with SGD optimizer for 50 iterations.

Table 1 Performance measures with different optimizers—MIAS

Experiment no.	Epoch	Optimizer	Sensitivity (%)	Specificity (%)	Accuracy (%)
1	30	SGD	93.3	92.4	89.3
2		Adam	94.2	92.4	91.2
3		RMSprop	92.1	91.5	91.4
4	40	SGD	92.6	92.4	92.3
5		Adam	91.4	92.5	93.3
6		RMSprop	92.5	93.3	91.5
7	50	SGD	96.4	95.3	95.8
8		Adam	93.1	92.2	92.0
9		RMSprop	94.3	93.9	92.1

Table 2 Performance measures with different optimizers—DDSM

Experiment no.	Epoch	Optimizer	Sensitivity (%)	Specificity (%)	Accuracy (%)
1	30	SGD	95.6	93.1	90.6
2		Adam	92.6	94.5	91.4
3		RMSprop	90.6	92.4	92.3
4	40	SGD	91.1	91.7	90.4
5		Adam	92.3	92.5	92.6
6		RMSprop	92.8	91.7	92.6
7	50	SGD	94.2	93.6	93.1
8		Adam	92.6	90.4	91.6
9		RMSprop	93.5	94.7	92.2

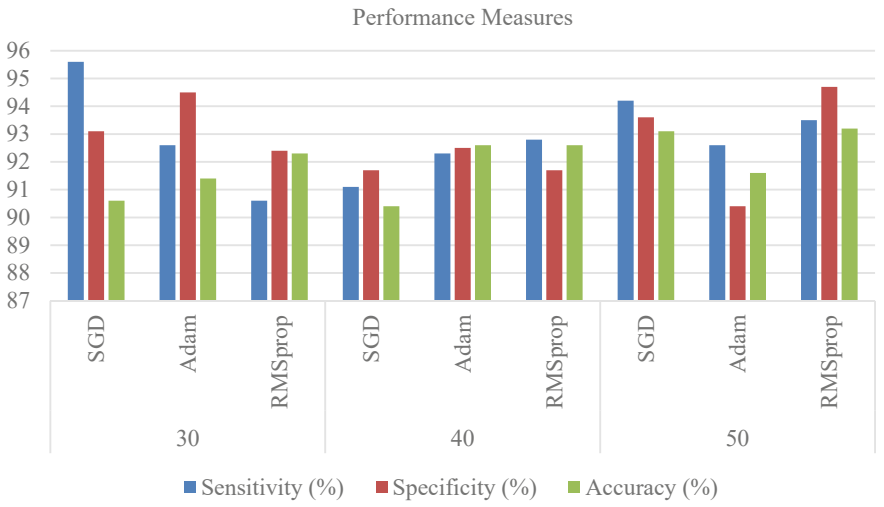


Fig. 6 Graphical representation of performance measures

5 Conclusion

The automatic classification of mammograms is designed by DCNN that tends tremendous progress in the last few years. Recently, many researchers developed DCNN to identify breast cancer in its early stages. In this research, initially the mammograms are preprocessed using CLAHE approach. These images are fed to DCNN with five convolutional layers that uses ReLU activation function. Out DCNN tested with different optimizers, namely SGD, RMSprop, and Adam. The filter sizes are 5×5 and 3×3 for the first four layers. The fully connected layer is used as the classifier and obtained the accuracy of 95.8% and 93.1% for MIAS and DDSM datasets respectively with SGD optimizer that results the improvement over the state-of-the-art methods in mammogram classification.

References

1. Aly GH, Marey M, El-Sayed SA, Tolba MF (2020) YOLO based breast masses detection and classification in full-field digital mammograms. Comput Methods Programs Biomed
2. Shallu S, Rajesh M (2018) Breast cancer histology images classification: training from scratch or transfer learning. ICT Express 4:247–254
3. Ragab DA, Sharkas M, Marshall S, Ren J (2019) Breast cancer detection using deep convolutional neural networks and support vector machines. PeerJ 7:e6201
4. Kumari LK, Jagadesh BN (2022) An adaptive teaching learning based optimization technique for feature selection to classify mammogram medical images in breast cancer detection. Int J Syst Assur Eng Manag
5. Dhungel N, Carneiro G, Bradley AP (2015) Automated mass detection in mammograms using cascaded deep learning and random forests. In: Conference 2015, DICTA
6. Kumari LK, Jagadesh BN (2022) Classification of mammograms using adaptive binary TLBO with ensemble classifier for early detection of breast cancer. Int J Inf Tecnol
7. Nguyen PT, Nguyen TT, Nguyen NC, Le TT (2019) Multiclass breast cancer classification using convolutional neural network. In: Conference 2019, ISEE, pp 130–134
8. Pillai A, Nizam A, Joshee M, Pinto A, Chavan S (2022) Breast cancer detection in mammograms using deep learning. In: Confernce 2022, advances in intelligent systems and computing, vol 1354. Springer, Singapore
9. Hadush S, Girmay Y, Sinamo A, Hagos G (2020) Breast cancer detection using convolutional neural networks. arXiv:200307911
10. Abbas Q (2016) Deepcad: a computer-aided diagnosis system for mammographic masses using deep invariant features. Computers 5(4)
11. Prakash S, Sangeetha K (2021) An early breast cancer detection system using recurrent neural network (rnn) with animal migration optimization (AMO) based classification method. J Med Imaging Health Inf 12:2950–2965
12. Zahoor S, Shoaib U, Lali IU (2022) Breast cancer mammograms classification using deep neural network and entropy-controlled whale optimization algorithm. Diagnostics 12(2):557
13. Gardezi SJS, Awais M, Faye I, Meriaudeau F (2017) Mammogram classification using deep learning features. In: Confernce ICSIPA, pp 485–488
14. Jayandhi G, Jasmine JSL, Joans SM (2022) Mammogram image classification system using deep learning for breast cancer diagnosis. AIP Conf Proc 2519:030066
15. Akselrod-Ballin A et al (2019) Predicting breast cancer by applying deep learning to linked health records and mammograms. Radiology 292:331–342

16. Houssein EH, Emam MM, Ali AA (2022) An optimized deep learning architecture for breast cancer diagnosis based on improved marine predators algorithm. Neural Comput Appl 34:18015–18033
17. Altan G (2020) Deep learning-based mammogram classification for breast cancer. Int J Intell Syst Appl Eng 8(4):171–176
18. Altameem A, Mahanty C, Poonia RC, Saudagar AKJ, Kumar R (2022) Breast cancer detection in mammography images using deep convolutional neural networks and fuzzy ensemble modeling techniques. Diagnostics 12:1812
19. Debelee TG, Amirian M, Ibenthal A, Palm G, Schwenker F (2018) Institute for computer sciences, social informatics and telecommunications engineering 2018. In: Confernce, Mekuria F et al. (eds) ICT4DA 2017, LNICST 244, pp 89–98
20. Nirmala G, Kumar PS (2020) Deep convolutional neural network for breast mass classification from mammogram. Biosc Biotech Res Comm 13(13):203–208
21. Shen L, Margolies LR et al (2019) Deep learning to improve breast cancer early detection on screening mammography. Sci Rep 9
22. Suckling J, Parker J, Dance D, Astley S, Hutt I, Boggis C, Ricketts I, Stamatakis E, Cerneaz N, Kok S et al, Mammographic image analysis society (mias) database v1 21, 201
23. Almeida R, Chen D, Filho A, Brandão W (2021) Machine learning algorithms for breast cancer detection in mammography images: a comparative study. Conf Int Conf Enterprise Inf Syst 1:660–667
24. Mechria H, Gouider MS, Hassine K (2019) Breast cancer detection using deep convolutional neural network. In: 11th International proceedings on agents and artificial intelligence (ICAART 2019), pp 655–660
25. Kumari LK, Jagadesh BN (2022) A robust feature extraction technique for breast cancer detection using digital mammograms based on advanced GLCM approach. EAI phat 22(30)

Malicious Social Bots Detection in the Twitter Network Using Learning Automata with URL Features

R. Kiran Kumar, G. Ramesh Babu, G. Sai Chaitanya Kumar, and N. Raghavendra Sai

Abstract By pretending to be a follower or making a huge number of false registrations through malicious activities, malicious social bots automate their social interactions and send out bogus tweets. The most well-known form of malware in online entertainment environments is social bots. They will disseminate rumors, convey false information, and even influence public opinion. Social bots are utilized to deliver better customer service by automating logical procedures. Additionally, spiteful social bots utilize malicious shortened URLs in tweets to send users' requests for peer-to-peer online communication to malicious servers. As a result, Twitter company's main goal is probably to distinguish damaging social bots from actual users. Compared to dangerous social bots that use social graph-based highlights, URL-based social bots (such as URL redirection, shared URL recursion, and URLs with spam content) require less investment (depending on the social associations of the clients). Vengeful social bots are also inappropriate for quickly inspecting URL redirect strings. By merging a trust calculation model with URL-based milestones, we suggest a learning automata-based Social Bot Recognition (LA-MSBD) calculation to identify trustworthy members (customers) in the Twitter organization. The employment of vindictive social bots to propagate false information has had negative real-world effects. One of the most challenging aspects of spotting bots in web-based entertainment is comprehending what social bots are capable of and looking at the quantitative highlights of their behavior. This concept made a distinction between typical clients and social bots. The strategy for spotting Twitter bots

R. K. Kumar
Department of Computer Science, Krishna University, Machilipatnam, Andhra Pradesh, India

G. R. Babu
Department of CSE, Shri Vishnu Engineering College for Women(A), Bhimavaram, India
e-mail: grameshcse@svecw.edu.in

G. S. C. Kumar (✉)
DVR and Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
e-mail: saijntuhphd@gmail.com

N. R. Sai
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India

R. Malhotra et al. (eds.), *High Performance Computing, Smart Devices and Networks*, Lecture Notes in Electrical Engineering 1087,
https://doi.org/10.1007/978-981-99-6690-5_43

using AI computations is suggested in this article. There is a detailed analysis of the K-NN algorithm, SVM algorithm, Naive Bayes algorithm, Random Forest algorithm, decision tree algorithm, and custom computing algorithm. The most effective learning model will be applied to the test data.

Keywords Bot · Malicious · SVM · Random forest

1 Introduction

A product software that poses as a real client in an unofficial Internet-based organization is known as a harmful social bot. Similar to this, harmful social bots carry out a variety of assaults, such as disseminating social spam content, fabricating characters, looking up online grades, and carrying out phishing attempts. When a Twitter user (customer) has to send out a tweet to URLs with nearby users (such as followers or devotees), the user shortens the URL using the abbreviated URL guidance (since a tweet is limited to 140 characters). Robots that engage with online entertainment clients are controlled by computers. Similar to how phishing URLs might be tweeted by a rogue social bot. By impersonating followers or making many false registrations, they make bogus tweets and automate their social connections. This is one example of their malevolent behavior. Additionally, dangerous social bots employ shortened versions of harmful URLs in tweets to forward users' requests for direct online communication to vindictive servers. Therefore, distinguishing damaging social bots from actual users is arguably the Twitter organization's primary duty. These bots are frequently modified to act partially or entirely independently and to resemble human customers. While some ingenious web-based entertainment bots exist, others are employed in unreliable and intrusive ways. According to several reports, the majority of the accounts used for online amusement are controlled by these harmful bots. Chatbots are bots that can keep a vocal match for a longer period of time and repeatedly, even though this is not what is typically expected of web-based entertainment bots.

Online entertainment is becoming an increasingly necessary part of our daily lives. Ordinary people will use this medium to access and disseminate information because billions of customers are continually producing and consuming data. Robots that can be used for virtual amusement on the web are modest projects that can be used to carry out a variety of useful and harmful tasks, reinforcing human behavior. Some online entertainment bots provide useful services, such as weather information and sports results. Anyone who interacts with these fantastic online entertainment bots will immediately recognize them as bots because they clearly bear their names. However, many of the bots used for online amusement are malicious bots that impersonate real customers. Customers lose faith in the ability of online entertainment venues to provide factual news because they believe manipulator bots are responsible for "pushing" the narratives at the top of their channels. Because there are so many individuals participating in web-based entertainment, bad clients like bots have started to sway the conversations of those who created them. These malevolent

bots have been used to silence protestor and activist communications, disseminate false information about upcoming politicians, boost the perceived notoriety of well-known figures, encourage social web spamming with links to commercial websites, and sway industry sectors. They make financial efforts to reduce the cost of inventories. These machines can also influence how regular virtual entertainment tests turn out. One of the most well-known forms of online amusement is the use of idle bots, which sleep for extended periods of time before rising up to send a flood of thousands of postings in a short amount of time (perhaps as a spam attack) and then reverting to a slow state. Bot attack tactics centered on entertainment. When an attacker guesses or observes which websites an organization frequently visits and infects at least one of them with malware, it is said to be conducting an initial attack. The jacking pattern makes use of moving objects to draw attention to a certain group of people. Utilizing appropriate hashtags to engage with a target audience is crucial for effective online marketing. It's important to focus on ethical strategies and avoid spam, malicious activities, or artificial methods like click farming to enhance your website's reputation or visibility.

2 BOTS

In virtual entertainment, the location of the bot is crucial. Twitter, a popular virtual entertainment venue, is full of bot accounts. About 15% of Twitter accounts, according to some surveys, work accordingly or semi-automatically. The qualities of Twitter may have added to the expansion of the number of bots. It is equally important that a Twitter bot is seen as a trusted source of data. Additionally, bot-linked accounts can be 2.5 $\times$ more persuasive than human-linked accounts. Vengeful bots can check Twitter metrics like moving dots. These bots can also influence insights based on Twitter insights like popular hashtags and the most persuasive customers. Boycott strategies have been routinely used to recognize malicious social bots. This is basically a variety of URLs sorted by a notorious malicious enemy of infection associations. While these strategies are fast (a simple dataset query is required) and have low false positive rates, they have a critical drawback in that they bombard newly created URLs. Since new URLs are constantly being produced, this is a huge restriction. Some efforts have been made to address these restrictions by addressing the issue using AI.

Web bots are programming programs that computerize commissions on the Internet. Web bots, WWW bots, or essentially bots have different names for them. Bots typically perform simple and mostly boring tasks at a much faster rate than people. The most widely recognized use of bots is web spidering, in which a robotic script retrieves, examines, and documents data from web servers at the speed of a human. Each server can have a log called robots.txt that contains tracking rules for the bot to maintain. Notwithstanding the applications described above, bots can be used in circumstances that require a faster reaction rate than humans (e.g., gaming

bots and bots for retail sites), or, less generally, in circumstances that require the copy of human movement.

3 Related Works

In view of long short-term memory (LSTM) design, Sneha Kudugunta and Emilio Ferrara suggested a deep brain network that can recognize bots at the tweet level utilizing both tweet content and metadata in [1]. Context-oriented highlights are maintained in LSTM deep networks using the text of tweets as supporting input after being extracted from the client information. From a single tweet, the model may achieve an astonishingly high accuracy of over 96% AUC. Additionally, they suggested methods for enhancing already-existing datasets by creating models with new names in light of synthetic minority oversampling technique (SMOTE). Both methods rely on specific components that can be directly accessed from the tweet and its metadata. The painting outperforms earlier avant-garde methods by employing a sparse, comprehensible arrangement of highlights that necessitates unnecessary preparation information.

A regulated AI characterization model was put up by Mohammed AL-Janabi, Ed de Quincey, and Peter Andras to distinguish the propagation of spiteful drugs in unofficial web-based networks in [2] (ONS). We identified online entertainment posts that contained malicious Uniform Resource Locators using highlights from a variety of sources (URLs). These URLs could lead customers to fraudulent websites, drive-by download attempts, phishing scams, spam, or other traps. The data array was created using the Twitter application programming interface (API) endpoint, and the dataset was tagged using Virus Total. Reflections were collected from multiple sources and employed in an irregular forest characterization model. A value revision of 0.89 was produced by the rough wood pattern with scarcely any adjustments or component choices. In any case, we got the chance to work on improving the classifier to 0.92 following more investigation and the usage of boundary and element selection strategies.

Using machine learning approaches, Chongzhen Zhang, Yanli Chen, and Yang Meng established a novel framework for network intrusion detection in [3]. We provide a special outage tracking system to enhance ordering capabilities. The dataset module and the criticism module both acknowledge the requalification of the classifier in the characterization module, which continuously ensures the high accuracy rate of the clustering module. We developed a novel characterization method by fusing SAE and RF. For typical ML computations, our methodology observes strong rendering capability and a reduction in dimensionality in dual and multiclass arrays. To fully recover the previous pressure flow, which can be employed for post-occasional examination and investigation, we fully utilize the attributes of the SAE model in relation to the component library in the dataset module. We assess our suggested approach and offer preparation and testing times using

the CICIDS2017 dataset. We fully utilize dual and multiclass characterization in comparison to different approaches in related work using the same dataset.

In [4], SylvioBarbon JR, Gabriel FC Campos, and Gabriel M. Tavares suggested employing wavelets to identify human, good, and malevolent bots in online social networks. The five steps in the suggested method were get, configure the profile, extract functions, choose functions, and prioritize. Since our concept applies to all OSNs, the acquisition phase may be tailored to any OSN API. Additionally, flexible is the ordering stage. In this review, we used Random Forests (RF) as classifiers. This decision was based on Singh et al. Ro et al. Igawa et al. which specifically address bots in OSN and showed outstanding performance with RF under enormous data settings. We have suggested a formula to determine whether a creator is a human.

A bot in an OSN can be malicious or real. An illustration of a compositional style that was included into the publishing content was obtained by calculation using the discrete wavelet transform. The classifiers were put to the test using separate and combined datasets. The proposed method produced high normal characterization corrections of 94.47% for both sets of data. According to the findings, the textual model we developed can classify consumer kinds according to how they compose their texts. We acknowledge that the suggested computation will greatly aid in the effort to combat OSN extortion. More thorough research into various AI methodologies might produce more persuasive findings.

Barbon [5] at OSN, a computation was presented to classify customers as real, genuine, or malicious bots. The algorithm used the discrete wavelet transform to illustrate a layout style integrated in post content. The tests are supplemented by classifiers that use single- and mixed-subject datasets, and it has been discovered that using an organized technique results in high normal characterization corrections of 94.47% for each dataset. The results show that, given their compositional style, the textual model they created has a promising level of accuracy in ordering clients.

An essential information method to examine social bots on Twitter has been put forth by Xia Liu [6]. This section includes information on Twitter's data collection, testing, opinion research, and econometric demonstrations. Let's look at the sample data's meaning to better understand bias and dispersion in bot data as well as why Twitter is such a crucial interpersonal platform for social bot research. We present the maximum entropy key new and the procedures to prepare and test the social robots under consideration in light of the measurements to evaluate the accuracy of the model in this review because the examination of sensations is significant in the measurement of a large amount of unstructured information. Finally, we summarize the main conclusions on friendly robots from an economic analysis of the material gathered from weekly message boards. While Twitter is a useful platform to concentrate on amiable bots and in-depth analysis of user-generated material, additional data from other stages, such as Facebook, will enhance our understanding of what social bots might mean for the caliber of content virality and data. Second, despite being a useful tool for automatically categorizing textual big data, sentiment analysis is constrained by the nuances and complexities of human language.

A deep learning-based social bot detection technique was put forth by Heng Ping and Sujuan Qin (2018) in [8]. The DeBD identification approach described in this

article is broken up into three sections: the position of the social bot in relation to the tweets' link components; the finding of social bots in relation to the tweets' metadata; and the combination of the two. A term implanted and connected in the opening portion is created from the client's tweet. Using CNN, the tweet's content component and its connections are then separated. The next section treats customer tweet metadata as passing information shared by amiable customers rather than as blatant cutting-edge highlights counting the transient client data and taking care of them in the LSTM brain's arrangement. The results of these two sections' exploratory tests demonstrate that the proposed DeBD is effective at identifying bots. CNN and LSTM combine the highlights from CNN and LSTM eliminated and concentrate on various portions of the elements of the tweet [9, 10]. Despite the fact that deep learning-based social bot discovery obtains the best accuracy across various information collections, it necessitates a substantial amount of user tweet data. In order to discriminate social bots as little as possible while keeping a high discovery rate, we may use tweets and customer data in further work.

A LA-MSBD calculation for MSBD is presented by RashmiRanjan Rout, Lingam and Somayajulu [1], which combines a computational trust model with a number of URL-based features. In addition, we investigate the reliability of tweets using Bayesian learning and DST (posted by each member). A set number of learning activities are also replicated using the suggested LA-MSBD computation depending on the likelihood that an activity has to be updated (i.e., the probability of a member posting malicious URLs in tweets). The benefits of ongoing learning are realized by the suggested LA-MSBD computation. In order to determine the exposure of our suggested LA-MSBD formula, two Twitter data collections are employed [11]. The study's findings demonstrate that the suggested LA-MSBD calculation outperforms existing and other computations in accuracy by up to 7%. The accuracy of the LA-MSBD calculation proposal for The Fake Project and Social Honeypot information indices was 95.37% and 91.77% for MSBD, respectively. Additionally, we might wish to investigate the relationships between the elements and how they affect.

4 Proposed Methodology

To overcome strategic constraints, we have promoted a deep learning method in view of faster RCNN and RNN. We can post tweets with hash tags for explicit points using the Twitter API[12]. We created an index of Twitter information to recognize people and bots. Enlisted clients or a PC program can post tweets. PC programs are modified to persistently tweet from a particular record for or against a particular topic for some non-partisan tweets. By the time typical customers tweet or reply to a particular topic, the recurrence of tweets is usually normal. The informative indices will be prepared using a calculation based on RCNN. To decide the sensations, the information will be managed by RCNN. We will actually take a look at time accounts to see if they are man-made or not. Expect it to be in 10 min or less. It was thought that it would take more than 8–10 min. We'll look at the types of gifts posted by a

customer account in the same way. A great deal of information is obtained during the human record misdirection hallmark test. This information is reused because it contains data that is superfluous to the result. The information is extracted, and all tainted information is purged before being put back into a dataset [13]. These data are taken into account in artificial intelligence models, and the results are obtained. Our proposed deep learning approach to identify vengeful social bots in the Twitter network has been isolated into five modules.

1. Acquisition and profiling configuration
2. Bootstrap data
3. Extraction of characteristics
4. Pre-processing of feature
5. Training classifiers.

Our model is applicable to any online social network, such as Twitter, and the acquisition step can be customized using the Twitter API- Tweepy. The classification step is also adaptable. In this work, we used faster region CNN and ResNet, which have shown promising results when used in big data environments to treat bots on online social networks such as Twitter [13, 14]. The following steps are included in the proposed approach pipeline: acquisition and profiling setup, data bootstrap feature extraction, feature pre-processing, and classification [10]. We must create a user profile for each user. The labeling followed the guidelines outlined in related works:

Malicious bots are automated profiles that share repetitive and spammy content, often with redundant information. Legitimate bots, on the other hand, are automated accounts that share innocuous content, typically assisted by auxiliary software. Human accounts, in contrast, are not automated and produce intelligent, original content (Fig. 1).

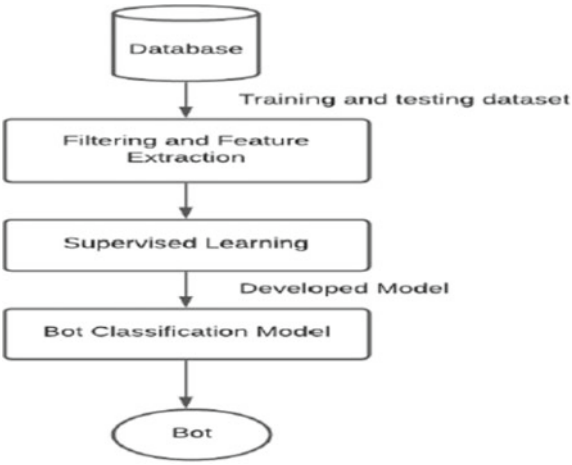


Fig. 1 A proposed framework

5 Configuration and Security Profile

The protection phase consists of obtaining information from an OSN to create an index of literary information. Our model predicts that the collected print set fixates on a lone main subject. This need is necessary as wavelet-based text extraction requires investigation of reported key terms for additional information. If the dataset did not depend on a single primary term, we suggest removing the important terms. To prepare a strategy that uses a regulated deep learning approach, we really want a large dataset. Following the world view based on profiles and a printed repository with some client publications, the text of each client is connected in total [15, 16]. The profile configuration provides the information that will be handled to create the vector of components that each class describes. Bootstrap information consists of untagged social bots. To detect malicious social bots at any given moment, we can search for comprehensive customer data that encompasses various details, including the organization, customer profiles, co-worker interactions, posted content, tweets, sentiments expressed, and ownership information.

5.1 Extraction and Preprocessing Are Characteristics

This phase aims to accumulate relevant data on vengeful robots. This remembers data, for example, the presence of URLs for a boycott highlights obtained from the URL string, host data, substance of the site such as HTML and JavaScript, awareness data, etc. This is just one illustration of the wide range of data types that can be collected by an Internet-based interpersonal organization to achieve item representation.

The unstructured data about the information is properly designed and fully transformed into a mathematical vector at this stage, so it can be handled very well in deep learning computations. For example, mathematical information can be used directly, and most words are used most of the time to address text-based or lexical substance.

6 Classifiers in Training

A general system of online pooling and offline preparation is proposed. The issuance of an order implies the definition of an information bearer of information. It uses a direct gain technique that depends on the preparation on the occurrences of each class. The main trademark of learning is speculation: The computation must create a reasonable result for inputs not tested during learning.

Information Exploration and Analysis.

(1) Identification of missing and unbalanced information, (2) feature extraction, (3) feature design, and (4) remove unnecessary properties.

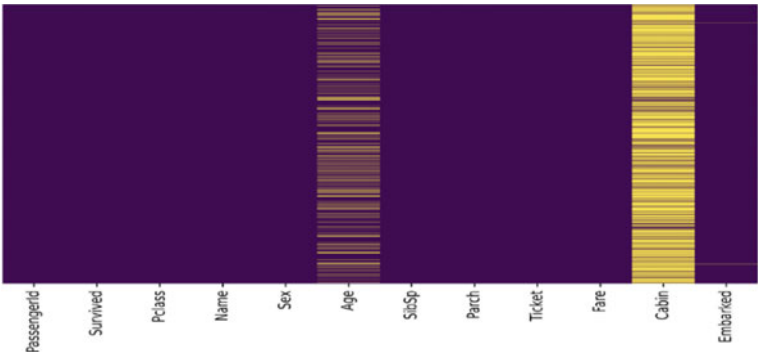


Fig. 2 Heatmap (missingness of data)

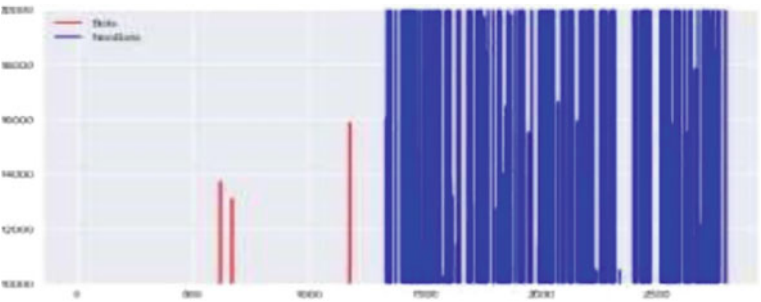


Fig. 3 Imbalance of data

The heatmap shows each of the missing qualities in yellow and the complete non-missing qualities in purple. The rendering of the area in the URL shows our most extreme invalid qualities, as shown by the yellow bars, while the state has a long profile and some missing qualities (Figs. 2 and 3).

Our next step was to identify the data imbalance we discovered whenever the listed count was between 10,000 and 20,000. If the listed count is between 10,000 and 20,000, there is a 5% chance that bots are detected, while the remaining 95% are not (Fig. 4).

We have found that robots have more devotees but fewer companions. So followers and comrades are big brands of bots and not. Also, for the confirmed credits. Then we use the connection with the spear to decide to highlight freedom. We found no connection between count status, identification, default profile, and default profile picture characteristics using Pearson's relationship and Spearman's relationship networks. Next, it should not be seen as a component to distinguish vengeful bots or non-bots. Since there are areas of strength between the properties controlled, the number of companions, and the number of supporters, remember to distinguish the bad bots or not. Since we can't connect to unmodified qualities, we'll use the display name, name representation, a state in the included layout, and registered creation to separate the

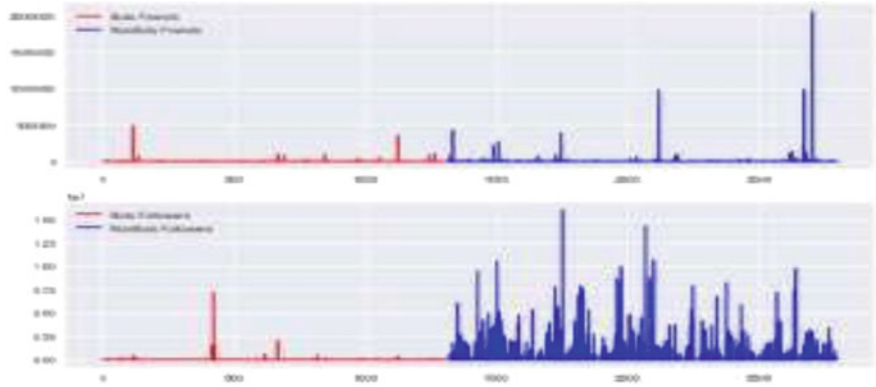


Fig. 4 Bots having more devotees but fewer companions

elements [16]. To complement the engineering element, we have made a series of word patterns that decide if a Twitter account is a bot or not. So we have a name on the screen and a representation of the name in the state that we change to a pair that uses the count vectorizer calculation. We did a component extraction, and the recorded count is parallel to wherever it's around 20,000; we faked and took this multitude of values and created a new element, and then we ran the classifier computation. We use the word pack model to create some vectors and various highlights, such as the feed transport in the representation, and distinguish that it is a real customer in light of the registered count and anticipate the remaining characteristics, as well as recognize harmful bots [17].

Recipient Job Attributes:

(1) False positive rate
(2) True positive rate.

A receiver performance curve, or ROC curve, is a graphical representation of the demonstrability of a dual classifier structure. The false positive ratio is the probability of mistakenly rejecting an invalid guess when processing a true certainty. Since the goal is only to distinguish difficult human records, the corpus is purged of all bot accounts. In fact, even after selection, the corpus may still contain a couple of fake human and bot accounts [18]. 15,000 tricky extras were physically created to make it look like they were created by people rather than robots and added to the corpus for research purposes (Fig. 5).

According to brain research, people lie about these things through web-based entertainment about their name, age, orientation, image, and area [19]. The records made were rich in these characteristics. The similarity of the two different accessible datasets is now tested. The tests carried out are the Mann–Whitney U-test which establishes that the two sets are comparative and the chi-square test to verify if the datasets are autonomous from each other [20].

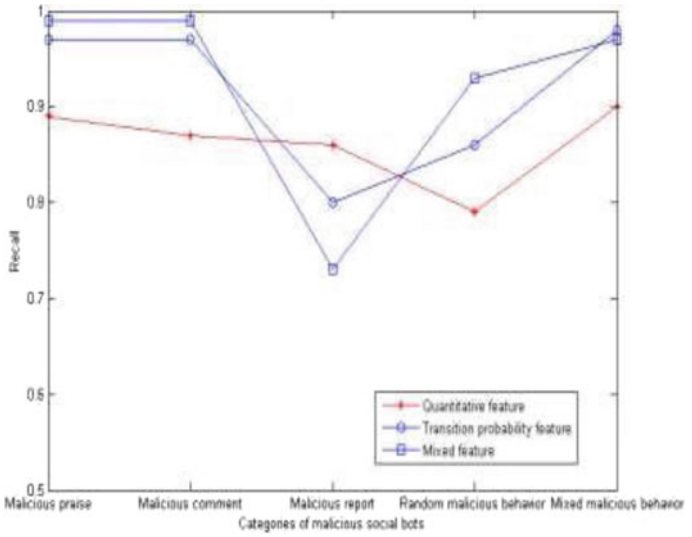


Fig. 5 Remembering multi-item view recognition strategies for various types of malicious social bots

7 Conclusion

The prevalence of contemporary bots in online entertainment spaces like Twitter makes the need for more effective and affordable bot recognition techniques obvious. We have put forth a quicker RCNN and ResNet calculation that enables us to identify bots at the tweet level using both tweet content and metadata. Our model can obtain an accuracy level that is greater than 97%. To update the estimation of the correctness of the activity, the suggested calculation executes a constrained arrangement of learning activities. To get rid of the tweet's content and the relationship between it, faster RCNN is utilized. The suggested calculation takes the use of ongoing learning. Twitter information indices are utilized to assess how well our calculation performed. The findings of the exploratory analysis demonstrate that the calculation improves upon other current algorithms when precision is compared.

References

1. Lingam G, Rout RR, Somayajulu DVLN (2019) Adaptive deep Q-learning model for detecting social bots and influential users in online social networks. Appl Intell 49(11):3947–3964
2. Choi D, Han J, Chun S, Rappos E, Robert S, Kwon TT (2018) Bit.ly/practice: uncovering content publishing and sharing through URL shortening services. Telematics Inform 35(5):1310–1323
3. Madisetty S, Desarkar MS (2018) A neural networkbased ensemble approach for spam detection in Twitter. IEEE Trans Comput Soc Syst 5(4):973–984

4. Jain AK, Gupta BB (2019) A machine learning based approach for phishing detection using hyperlinks information. J Ambient Intell Hum Comput 10(5):2015–2028
5. Wu T, Liu S, Zhang J, Xiang Y (2017) Twitter spam detection based on deep learning. In: Proc. Australas. Comput. Sci. Week Multiconf. (ACSW)
6. Jain AK, Gupta BB (2019) A machine learning based approach for phishing detection using hyperlinks information. J Ambient Intell Hum Comput 10(5):2015–2028
7. Echeverria J, Zhou S (2017) Discovery, retrieval, and analysis of the 'star wars' botnet in twitter. In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp 1–8
8. Dorri A, Abadi M, Dadfarnia M (2018) SocialBotHunter: botnet detection in twitter-like social networking services using semisupervised collective classification. In: Proceedings of IEEE 16th 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp 496–503
9. Agarwal M, Zhou B (2014) Using trust model for detecting malicious activities in Twitter. In: Proceedings of social computing, behavioral-cultural modeling and prediction: 7th International conference, predict. Springer, pp 207–214; [27] Lingam G, Rout RR, Somayajulu DVLN (2018) Detection of social botnet using a trust model based on spam content in twitter network. In Proceedings of IEEE 13th international conference on industrial and information systems (ICIIS)
10. Moayedikia A, Ong K-L, Boo YL, Yeoh WGS (2018) Task assignment in microtask crowdsourcing platforms using learning automata. Eng Appl Artif Intell 74:212–225
11. Lingam G, Rout RR, Somayajulu D (2018) Learning automatabased trust model for user recommendations in online social networks. Comput Electr Eng 66:174–188
12. Chand MS, Kumar B (2018) Target coverage heuristic based on learning automata in wireless sensor networks. IET Wireless Sensor Syst 8(3):109–115
13. Wang G, Zhang X, Tang S, Wilson C, Zheng H, Zhao BY (2017) Clickstream user behavior models. ACM Trans Web 11(4), Art. no. 21
14. Liu Y, Wang C, Zhang M, Ma S (2017) User behaviormodeling for better Web search ranking. Front Comput Sci 11(6):923–936
15. Al-Qurishi M, Hossain MS, Alrubaian M, Rahman SMM, Alamri A (2018) Leveraging analysis of user behavior to identify malicious activities in large-scale social networks. IEEE Trans Ind Inf 14(2):799–813
16. Buczak A, Guven E (2017) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutorials 18(2):1153–1176; [2] Farnaaz N, Jabbar MA (2016) Random forest modeling for network intrusion detection system. Proc Comput Sci 89:213–217
17. Wang H, Gu J, Wang S (2017) An effective intrusion detection framework based on SVM with feature augmentation. Knowl-Based Syst 136:130–139
18. Akashdeep IM, Manzoor I, Kumar N (2017) A feature reduced intrusion detection system using ann classifier. Expert Syst Appl 88:249–257
19. Chuan-Long Y, Yue-Fei Z, Jin-Long F et al (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961
20. Lopez-Martin M, Carro B, SanchezEsguevillas A (2019) Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst Appl 141, Article ID 112963