

Algorithms for Intelligent Systems

Series Editors: Jagdish Chand Bansal · Kusum Deep · Atulya K. Nagar

Brahmjit Singh

Carlos A. Coello Coello

Poonam Jindal

Pankaj Verma *Editors*

Intelligent Computing and Communication Systems



Springer

Algorithms for Intelligent Systems

Series Editors

Jagdish Chand Bansal, Department of Mathematics, South Asian University,
New Delhi, Delhi, India

Kusum Deep, Department of Mathematics, Indian Institute of Technology Roorkee,
Roorkee, Uttarakhand, India

Atulya K. Nagar, School of Mathematics, Computer Science and Engineering,
Liverpool Hope University, Liverpool, UK

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

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

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

More information about this series at <http://www.springer.com/series/16171>

Brahmjit Singh · Carlos A. Coello Coello ·
Poonam Jindal · Pankaj Verma
Editors

Intelligent Computing and Communication Systems



Springer

Editors

Brahmjit Singh
Department of Electronics
and Communication Engineering
National Institute of Technology
Kurukshetra, India

Poonam Jindal
Department of Electronics
and Communication Engineering
National Institute of Technology
Kurukshetra, India

Carlos A. Coello Coello
Departamento de Computación
CINVESTAV-IPN
Mexico City, Mexico

Pankaj Verma
Department of Electronics
and Communication Engineering
National Institute of Technology
Kurukshetra, India

ISSN 2524-7565

Algorithms for Intelligent Systems

ISBN 978-981-16-1294-7

<https://doi.org/10.1007/978-981-16-1295-4>

ISSN 2524-7573 (electronic)

ISBN 978-981-16-1295-4 (eBook)

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2021

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Committees

Chief Patron

Dr. Satish Kumar, Director, NIT Kurukshetra

Patron

Dr. N. P. Singh, Head, ECE Department, NIT Kurukshetra

Organizing Committee

Organizing Chairs

Prof. Brahmjit Singh, NIT Kurukshetra

Dr. Poonam Jindal, NIT Kurukshetra

Dr. Pankaj Verma, NIT Kurukshetra

Organizing Secretaries

Dr. Gaurav Saini, NIT Kurukshetra

Dr. T. N. Sasamal, NIT Kurukshetra

Advisory Committee

- Dr. Carlos A. Coello Coello, CINVESTAV-IPN, Mexico
Dr. Gurdeep Singh Hura, University of Maryland Eastern Shore, USA
Dr. Garenth Lim King Hann, Curtin University, Malaysia
Dr. Madhu Chetty, Federation University Australia
Dr. Lau Siong Hoe, Multimedia University, Malaysia
Dr. San Murugesan, Western Sydney University, Australia
Dr. A. Carlos, CINVESTAV, Mexico, North America
Dr. Arokiaswami Alphones, NTU, Singapore
Dr. Ljiljana Trajkovic, Simon Fraser University, Canada
Dr. Anthony C. Boucouvalas, University of Peloponnese, Greece
Dr. Anand Mohan, IIT BHU
Dr. A. K. Singh, NIT Kurukshetra
Dr. N. S. Rajput, IIT BHU
Dr. Amit Kumar, IIT Jodhpur
Dr. Monika Aggarwal, IIT Delhi
Dr. Sukumar Nandi, IIT Guwahati
Dr. Y. N. Singh, IIT Kanpur
Dr. Sudeb Dasgupta, IIT Roorkee
Dr. Ravi Panwar, IIIT Jabalpur
Dr. Rajoo Pandey, NIT Kurukshetra
Dr. Umesh Ghanekar, NIT Kurukshetra
Dr. A. K. Singh, NIT Kurukshetra
Dr. R. K. Sharma, NIT Kurukshetra
Dr. Vikas Mittal, NIT Kurukshetra
Dr. Vrinda Gupta, NIT Kurukshetra
Dr. Sudeep, Nirma University
Dr. Sanjeev Sofat, PEC Chandigarh
Dr. J. C. Bansal, South Asian University, New Delhi
Dr. Ashwani Rana, NIT Hamirpur
Dr. Rajeevan Chandel, NIT Hamirpur
Dr. Tarun Rawat, NSUT Delhi
Dr. Anshul Gupta, NIT Raipur
Dr. Amit Kumar Singh, NIT Patna
Dr. Navneet Gupta, BITS Pilani
Dr. Rajesh Khanna, TIET Patiala
Dr. Rajesh Kumar, MNIT Jaipur

Preface

The next-generation communication technologies are envisaged to offer increased throughput by an order of magnitude and greater efficiency in terms of both frequency spectrum utilization and energy consumption. However, these communication systems including Internet of things and the fifth-generation wireless communication technologies are expected to operate in a very complex, uncertain, dynamic and diverse environment. Such an operating environment along with hardware intricacies of the end-to-end communication infrastructure limits the system performance. In addition, the computing functionality of these networking platforms is highly distributed and decentralized in nature. Designing these systems while ensuring consistent performance in complex and vulnerable environment is indeed a very challenging task. Intelligent algorithms are the enablers to manage the vast complexity of the operating environment and optimize the overall performance of the communication technologies.

This book on *Intelligent Computing and Communication Systems* aims at providing a comprehensive and insightful understanding of intelligent algorithms in context with their applications in the recently developed and the immediate future communication technologies under development as well. It also covers the topics on how to develop intelligent algorithms for computing functionality in the end-to-end networking platforms. It illustrates the recent developments, open technological challenges and future directions in the areas of data analysis, applications of the game theory, autonomous entities, evolutionary computation, smart ubiquitous computing and intelligent architectures with major focus on communication technologies and computing platforms. Capability and capacity enhancement in the next-generation communication technologies subsequent to the application of novel intelligent algorithms are also covered in the interesting and illustrative manner for the readers.

The book embodies an edited volume of the Proceedings of the International Conference on Cutting-Edge Technologies in Computing and Communication Engineering (IC4E-2020) organized at National Institute of Technology, Kurukshetra, during 6–7 November 2020. The conference received overwhelming response from

281 authors from across the world. Having exercised rigorous review process, high-quality 42 research articles accepted and presented in the conference are being brought out as the chapters in the present volume.

The salient features of the book include the following:

- Critical review of research efforts in the area of next-generation intelligent communication technologies
- Thought-provoking discussion on the recent developments generating interest in the next wave of the technology space
- Understanding of the capabilities and limitations of the contemporary and recently developed intelligent algorithms for communication and computing platforms for future problem formulation for the interested researchers
- Includes the state-of-the-art research methodologies and their utilization in the development of intelligent systems
- Covers leverage of the big data analysis to design robust and resilient communication systems.

The book aims to familiarize the readers including academics, researchers, practitioners and industry professionals with the most imminent wave of communication technology-a pre-empt to walk through the future way of communication and computing.

We the team of IC4E-2020 express our sincere thanks to one and all particularly the authors for being a part of this conference and making it indeed a successful and very rich in research contents in the area of computing and communication technologies.

Kurukshetra, India
Mexico City, Mexico
Kurukshetra, India
Kurukshetra, India

Brahmjit Singh
Carlos A. Coello Coello
Poonam Jindal
Pankaj Verma

Acknowledgements

At the outset, we express our sincere gratitude to Editors of the Book Series *Algorithms for Intelligent Systems* Dr. Jagdish Chand Bansal, Prof. Kusum Deep and Prof. Atulya K. Nagar for accepting our proposal for publishing the proceedings of the International Conference on Cutting-Edge Technologies in Computing and Communication Engineering (IC4E-2020). We also acknowledge the help and cooperation extended by the Team Springer Nature, especially Sh. Aninda Bose, Senior Publishing Editor, for his valuable inputs, guidance and administrative support throughout the period of the conference organization. We are also grateful to Ms. Silky Abhay Sinha for her prompt and diligent communication guiding us for preparing the manuscripts in conformity with the standard norms. We thank all eminent speakers from academia, industry and R&D organizations for delivering the keynote talks. Our special thanks are due to Prof. Carlos A. Coello Coello, CINVESTAV-IPN, Mexico; Dr. Amit Bhardwaj, IIT Jodhpur; Dr. N. S. Rajput, IIT BHU; Dr. Lalit Singh BARC, Department of Atomic Energy, Government of India; and Dr. J. C. Bansal, South Asian University, Delhi.

We are immensely grateful to Dr. Satish Kumar, Director, NIT Kurukshetra, for his constant support and motivation for organizing IC4E-2020. We are also thankful to Dr. N. P. Singh, Head, ECE Department, NIT Kurukshetra, for his unconditional support and cooperation extended towards successful organization of the conference. Heartfelt acknowledgement is due to the members of Advisory/Technical Programme Committee, reviewers, session chairs for their valued contribution and of course the authors for sharing their research work on the platform of IC4E-2020. We thank to one and all who have contributed directly or indirectly in the smooth conduct and grand success of the conference organized through online mode. Finally, we the team of IC4E-2020 expresses our gratitude to the Almighty for blessing us all with good health during the challenging times of the COVID-19 pandemic situation.

Contents

Part I Intelligent Communication Systems

1 Spectral Efficiency Analysis of Massive MIMO Systems	3
Vaibhav Thakur and Manoranjan Rai Bharti	
2 Deep Learning-Enabled Interference Management in LTE-D2D Communication	15
Sandeepika Sharma and Brahmjit Singh	
3 Traffic Violations Prediction System on the Basis of Human Behaviour	25
Deepti Goel, Rajesh Bhatia, and Kashish Bhatia	
4 Recent Developments and Challenges in Intelligent Transportation Systems (ITS)—A Survey	37
Vishal Sharma, Love Kumar, and Sergey Sergeyev	
5 A Modified YOLO Model for On-Road Vehicle Detection in Varying Weather Conditions	45
Rajib Ghosh	
6 Different Techniques Used in Smart Traffic Light Management System Using CCTV	55
Riddhika Rawat, Shruti Singh, and Sumit Kumar	
7 A Novel Rule-Based Expert System for Penalty Prediction for Two-Wheeler's Traffic Rules Violation in India	67
Anoop Sahani, Niranjan Panigrahi, Jagadish Mohanty, Anita Moharana, and Diptimayee Sahoo	
8 Hybrid Material-Based Dual-Band Yagi-Uda Antenna with Enhanced Gain for the Ku-Band Applications	77
Rajesh Yadav, Shailza Gotra, V. S. Pandey, and Brahmjit Singh	

9	Broadband Electromagnetic Performance Analysis of Radome Structures Realized Using Hybrid Equilibrium Optimization Strategy	87
	Anindya Midya Chowdhury, Ravi Yadav, Varun Chaudhary, and Ravi Panwar	
10	Design, Optimization, and Critical Analysis of Metamaterial Superstrate-Coupled High-Gain Microstrip Patch Antenna	97
	Atipriya Sharma, Ravi Panwar, and Rajesh Khanna	
11	Cheating-Tolerant and Threshold-Based Secure Information Exchange Among Propinquity of Adversaries	105
	Anindya Kumar Biswas and Mou Dasgupta	
12	Physical Layer Security-Based Relay Selection for Wireless Cooperative Networks: A Reinforcement Learning Approach	115
	Anil Kumar Kamboj, Poonam Jindal, and Pankaj Verma	
13	A Review of Security Threats in Software-Defined Networking	123
	Sukhveer Kaur, Krishan Kumar, and Naveen Aggarwal	
14	Hardware-Based Analysis of PCG Signal for Heart Conditions	133
	Takhellambam Gautam Meitei, Sinam Ajitkumar Singh, and Swanirbhar Majumder	
15	Elliptic Curve Cryptography: A Software Implementation	143
	Sumit Singh Dhanda, Brahmjit Singh, and Poonam Jindal	
16	When Distributed Ledger Technology Meets Traditional Payment Systems—Benefits and Challenges	149
	Saurabh Jain, Adarsh Shukla, and Kashish Srivastava	
17	Wireless Lighting System for Rural Households in India	159
	Shantanu Acharya, Priya Debnath, Dipta Chakraborty, Rimpi Baishya, and Sayan Deb	
18	A Study of OpenStack Networking and Auto-Scaling Using Heat Orchestration Template	169
	Karamjeet Kaur, Veenu Mangat, and Krishan Kumar	

Part II Intelligent Algorithms: Recent Developments

19	Recent Development, Challenges and Futuristic Trends in Cloud Computing—A Survey	179
	Sahul Goyal and Lalit K. Awasthi	
20	A Comparative Approach for Email Spam Detection Using Deep Learning	187
	Akhil Pratap Singh, Ashish Singh, and Kakali Chatterjee	

21 Sentiment Analysis Algorithms: Classifiers and Their Comparison	201
Shubham Joshi, Rochit Dubey, Aryav Tiwari, and Poonam Jindal	
22 Email Sentiment Classification Using Lexicon-Based Opinion Labeling	211
Ulligaddala Srinivasarao and Aakanksha Sharaff	
23 Real-Time Facial Emotion Recognition Using Deep Learning	219
Shruti Chand, Apoorva Singh, Ria Bhatia, Ishween Kaur, and K. R. Seeja	
24 Clustering of Single-Cell Transcriptome Data Based on Evolutionary Algorithm in Assimilation with Fuzzy C-Means	227
Amika Achom and Ranjita Das	
25 Waste Segregation to Ease Recyclability	237
Rahul Kumar Verma and Suneeta Agarwal	
26 Comparative Analysis of Intelligent Systems using Support Vector Machine for the Detection of Diabetic Retinopathy	245
G. Sri Venkateswara Reddy, Dolly Das, Saroj Kumar Biswas, B. Sai Prashanth, B. Praful Bhargav, T. Vinay Kumar, Monali Bordoloi, Biswajit Purkayastha, and Tohida Rehman	
27 Case-Based Expert System for Early Detection of Diabetic Retinopathy	259
Rahul Barman, Saroj Kumar Biswas, Dolly Das, Biswajit Purkayastha, and Malaya Dutta Borah	
28 Decision Making Using Interval-Valued Pythagorean Fuzzy Set-Based Similarity Measure	269
G. Punnam Chander and Sujit Das	
29 Forest Combustion Recognition Using Deep Learning	279
Kota Jahnavi, Hari Kishan Kondaveeti, and Asish Kumar Dalai	
30 The Importance of Diversity in Multi-objective Evolutionary Algorithms	291
Carlos A. Coello Coello and Ma. Guadalupe Castillo Tapia	
31 Viewer's Sentiments on Game of Thrones: An Automated Lexicon-Based Sentiment Analysis on Real-Time YouTube Comments	299
Shivam Sharma and Hemant Kumar Soni	
32 Improving the Accuracy of Writer Detection of Handwritten Text Using Image Hashing	313
Devvrat Bhardwaj and Prateek Thakral	

33 Software Defect Prediction by Strong Machine Learning Classifier	321
Meetesh Nevendra and Pradeep Singh	
34 FECG Extraction Using 1D Convolution Neural Network	331
Yojana Sharma, Shashwati Ray, and Om Prakash Yadav	
35 HMM Model for Brain Tumor Detection and Classification	339
Parth Sharma and Rakesh Sharma	
36 ANN Control Algorithms with Different Training Methods as Applied to PMSM Drive	347
Krishna Kokre and S. V. Jadhav	
37 An Analysis and Comparison of Community Detection Algorithms in Online Social Networks	363
Sanjeev Dhawan, Kulvinder Singh, and Amit Batra	
38 A Deep Learning Approach for Network Intrusion Detection Using Non-symmetric Auto-encoder	371
Divya Nehra, Veenu Mangat, and Krishan Kumar	
39 Efficient Deep Learning Framework with Group Convolution for Segmentation of Histopathology Image	383
Amit Kumar Chanchal, Aman Kumar, Kumar Alabhyा, Shyam Lal, and Jyoti Kini	
40 Dynamic Reusability Measurement Using Machine Learning Algorithms in Object-Oriented Environment	393
Manju and Pradeep Kumar Bhatia	
41 Comparison of Different Machine Learning and Deep Learning Emotion Detection Models	401
Akanksha Aggarwal, Sahil Garg, Raghav Madaan, and Rajender Kumar	
42 An Improvised Particle Swarm Optimization Using Balanced Local Best	409
Bubul Doley, Arpita Nath Boruah, Sazidur Rahman, Saroj Kumar Biswas, Manomita Chakraborty, Sunita Sarkar, and Biswajit Purkayastha	

Editors and Contributors

About the Editors

Brahmjit Singh (Senior Member IEEE) is the Professor of Electronics and Communication Engineering, National Institute of Technology (NIT), Kurukshetra-136119 (India). He has published 171 research papers in international/national journals/conferences and co-authored four book chapters. He has been the Dean of Research and Consultancy and Chairman, Departments of Electronics and Communication Engineering and Computer Engineering NIT Kurukshetra. He is recipient of the Best Faculty Award (Administration)-2019 conferred by the Institute and the best research paper award from the Institution of Engineers (India). He obtained B.E. in Electronics Engineering from MNIT Jaipur, M.E. in Electronics and Communication Engineering with specialization in Microwave and Radar from Indian Institute of Technology Roorkee and Ph.D. from GGS IP University, Delhi. His current research interest includes machine learning in wireless communication, security in wireless networks, multimedia sensor networks and vehicular communication. He is the reviewer for several IEEE journals and conferences and presently serving as Vice Chair of IEEE ComSoc Delhi Chapter.

Dr. Carlos A. Coello Coello received a Ph.D. in Computer Science from Tulane University (USA) in 1996. He is currently Full Professor with distinction (Investigador Cíavestav 3F) at CINVESTAV-IPN, Mexico City, Mexico. He has done pioneering research work on evolutionary multi-objective optimization and developed the first micro-genetic algorithm for multi-objective optimization and the first Pareto-based multi-objective artificial immune system. He has published over 480 papers in international peer-reviewed journals, book chapters and conferences and co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems*. With 46,000 citations, his h-index rises to 83. He received 2007 National Research Award, Mexican Academy of Science. He is IEEE Fellow for “contributions to multi-objective optimization and constraint-handling techniques” and the recipient of the prestigious 2013 IEEE Kiyo Tomiyasu Award (2012), National Medal of Science and Arts in the area of Physical, Mathematical and Natural Sciences and TWAS Award

in Engineering Sciences. He serves as Associate Editor of several journals including IEEE Transactions on Evolutionary Computation, Computational Optimization and Applications, Pattern Analysis and Applications, Journal of Heuristics, Evolutionary Computation and Applied Soft Computing.

Dr. Poonam Jindal is working with ECE Department National Institute of Technology Kurukshetra. She received her Ph.D. in Electronics and Communication Engineering from NIT Kurukshetra in 2016. She obtained degrees of M.Tech. and B.Tech. in 2005 and 2003, respectively. She has published 60 papers in various international journals and conferences and book chapters. Her research interests include wireless network security, wireless communication, physical layer security, Internet of Things, and security optimization in wireless networks. She is IEEE Member and Regular Reviewer for various reputed international journals and conferences.

Pankaj Verma received AMIETE (Associate Member of Institute of Electronics and Telecommunication Engineering) degree from Institute of Electronics and Telecommunication Engineers, New Delhi in 2009, M Tech in Microwave and Optical Communication from Delhi Technological University (formerly Delhi College of Engineering) in 2011 and Ph.D. from National Institute of Technology, Kurukshetra in 2017. He has been awarded Gold Medal in AMIETE (2009). Currently, he is working as an Assistant Professor in Electronics and Communication Engineering Department, National Institute of Technology, Kurukshetra, India. He has published several research papers in international/national journals and conferences. He is also having one-year industrial experience in Intellectual Property (IP) domain and earlier served Indian Air Force for three years. His research interests are in Wireless Communications, Cognitive Radio Systems, Optical Communication, Signal Processing, Visible Light Communication, Security, Machine Learning, Artificial Intelligence and Photonics Crystal Fibre Sensors.

Contributors

Shantanu Acharya Department of Electrical and Electronics Engineering, Institute of Chartered Financial Analysts of India (ICFAI) University, Agartala, Tripura, India

Amika Achom National Institute of Technology, Mizoram, Aizawl, India

Suneeta Agarwal Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India

Akanksha Aggarwal National Institute of Technology, Kurukshetra, Haryana, India

Naveen Aggarwal UIET, Panjab University, Chandigarh, India

Kumar Alabhyā Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, Karnataka, India

Lalit K. Awasthi Dr. B. R. Ambedkar, National Institute of Technology, Jalandhar, India

Rimpi Baishya Department of Electrical and Electronics Engineering, Institute of Chartered Financial Analysts of India (ICFAI) University, Agartala, Tripura, India

Rahul Barman Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Amit Batra Department of Computer Science and Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

Devvrat Bhardwaj Department of Computer Science and Engineering, Jaypee University of Information Technology, Solan, India

Manoranjan Rai Bharti Electronics and Communication Engineering Department, National Institute of Technology, Hamirpur, India

Kashish Bhatia UCOE, Punjabi University, Patiala, India

Pradeep Kumar Bhatia Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, India

Rajesh Bhatia PEC University of Technology, Chandigarh, India

Ria Bhatia Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, India

Anindya Kumar Biswas Department of Computer Application, National Institute of Technology, Raipur, India

Saroj Kumar Biswas Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Malaya Dutta Borah Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Monali Bordoloi Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Arpita Nath Boruah Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India

Ma. Guadalupe Castillo Tapia Departamento de Administración, UAM Azcapotzalco, México City, Mexico

Dipta Chakraborty Department of Electrical and Electronics Engineering, Institute of Chartered Financial Analysts of India (ICFAI) University, Agartala, Tripura, India

Manomita Chakraborty Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India

Amit Kumar Chanchal Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, Karnataka, India

Shruti Chand Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, India

G. Punnam Chander Department of Computer Science and Engineering, NIT Warangal, Warangal, India

Kakali Chatterjee Computer Science and Engineering, National Institute of Technology Patna, Patna, Bihar, India

Varun Chaudhary Electronics and Communication Engineering, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Madhya Pradesh, India

Carlos A. Coello Coello Departamento de Computación, CINVESTAV-IPN, Mexico City, Mexico

Asish Kumar Dalai Department of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

Dolly Das Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Ranjita Das National Institute of Technology, Mizoram, Aizawl, India

Sujit Das Department of Computer Science and Engineering, NIT Warangal, Warangal, India

Mou Dasgupta Department of Computer Application, National Institute of Technology, Raipur, India

Sayan Deb Department of Electrical and Electronics Engineering, Institute of Chartered Financial Analysts of India (ICFAI) University, Agartala, Tripura, India

Priya Debnath Department of Electrical and Electronics Engineering, Institute of Chartered Financial Analysts of India (ICFAI) University, Agartala, Tripura, India

Sumit Singh Dhanda National Institute of Technology, Kurukshetra, India

Sanjeev Dhawan Department of Computer Science and Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

Bubul Doley Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India

Rochit Dubey Electronics and Communication Department, National Institute of Technology, Kurukshetra, India

Sahil Garg National Institute of Technology, Kurukshetra, Haryana, India

Rajib Ghosh National Institute of Technology Patna, Patna, India

Deepti Goel PEC University of Technology, Chandigarh, India

Shailza Gotra Department of ECE, NIT Delhi, Delhi, India

Sahul Goyal Dr. B. R. Ambedkar, National Institute of Technology, Jalandhar, India

S. V. Jadhav Department of Electrical Engineering, College of Engineering, Pune, India

Kota Jahnavi Department of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

Saurabh Jain School of Computer Science, University of Petroleum and Energy Studies, Bidholi, Dehradun, India

Poonam Jindal Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, Haryana, India

Shubham Joshi Electronics and Communication Department, National Institute of Technology, Kurukshetra, India

Anil Kumar Kamboj Department of Electronics and Communication, NIT, Kurukshetra, Haryana, India

Ishween Kaur Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, India

Karamjeet Kaur UIET, Panjab University, Chandigarh, India

Sukhveer Kaur UIET, Panjab University, Chandigarh, India

Rajesh Khanna Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

Jyoti Kini Department of Pathology, Kasturba Medical College, Mangalore, Manipal Academy of Higher Education, Manipal, India

Krishna Kokre Department of Electrical Engineering, College of Engineering, Pune, India

Hari Kishan Kondaveeti Department of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

Aman Kumar Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, Karnataka, India

Krishan Kumar University Institute of Engineering and Technology, Panjab University, Chandigarh, India

Love Kumar Department of Electronics and Communication, DAVIET, Jalandhar, Punjab, India

Rajender Kumar National Institute of Technology, Kurukshetra, Haryana, India

Sumit Kumar Department of Computer Engineering, Women Institute of Technology, Dehradun, India

Shyam Lal Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, Karnataka, India

Raghav Madaan National Institute of Technology, Kurukshetra, Haryana, India

Swanirbhar Majumder Department of IT, Tripura University, Agartala, Tripura, India

Veenu Mangat University Institute of Engineering and Technology, Panjab University, Chandigarh, India

Manju Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, India

Takhellambam Gautam Meitei Department of Photonics, National Chiao Tung University, Hsinchu, Taiwan

Anindya Midya Chowdhury Mechatronics, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Madhya Pradesh, India

Jagadish Mohanty Department of CSE, Parala Maharaja Engineering College, Berhampur, Odisha, India

Anita Moharana Department of CSE, Parala Maharaja Engineering College, Berhampur, Odisha, India

Divya Nehra University Institute of Engineering and Technology, Panjab University, Chandigarh, India

Meetesh Nevendra Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

V. S. Pandey Department of Applied Sciences, NIT Delhi, Delhi, India

Niranjan Panigrahi Department of CSE, Parala Maharaja Engineering College, Berhampur, Odisha, India

Ravi Panwar Discipline of Electronics and Communication Engineering, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, Madhya Pradesh, India

B. Praful Bhargav Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Biswajit Purkayastha Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India

Sazidur Rahman Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India

Riddhika Rawat Department of Computer Engineering, Women Institute of Technology, Dehradun, India

Shashwati Ray Bhilai Institute of Technology, Durg, CG, India

Tohida Rehman Information Technology, Jadavpur University, Kolkata, India

Anoop Sahani Department of CSE, Parala Maharaja Engineering College, Berhampur, Odisha, India

Diptimayee Sahoo Department of CSE, Parala Maharaja Engineering College, Berhampur, Odisha, India

B. Sai Prashanth Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Sunita Sarkar Department of Computer Science, Assam University, Silchar, Assam, India

K. R. Seeja Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, India

Sergey Sergeyev Aston Institute of Photonics and Technologies (AiPT), Aston University, Birmingham, UK

Aakanksha Sharaff Department of Computer Science and Engineering, National Institute of Technology Raipur, Raipur, Chhattisgarh, India

Atipriya Sharma Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

Parth Sharma Department of Electronics and Communication Engineering, National Institute of Technology, Hamirpur, Hamirpur, India

Rakesh Sharma Department of Electronics and Communication Engineering, National Institute of Technology, Hamirpur, Hamirpur, India

Sandeepika Sharma ECE Department, NIT Kurukshetra, Kurukshetra, India

Shivam Sharma Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Madhya Pradesh, Maharajpura, Gwalior, MP, India

Vishal Sharma Aston Institute of Photonics and Technologies (AiPT), Aston University, Birmingham, UK

Yojana Sharma Bhilai Institute of Technology, Durg, CG, India

Adarsh Shukla School of Computer Science, University of Petroleum and Energy Studies, Bidholi, Dehradun, India

Akhil Pratap Singh Computer Science and Engineering, National Institute of Technology Patna, Patna, Bihar, India

Apoorva Singh Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, India

Ashish Singh School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India

Brahmjit Singh Department of ECE, National Institute of Technology, Kurukshetra, India

Kulvinder Singh Department of Computer Science and Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

Pradeep Singh Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

Shrutika Singh Department of Computer Engineering, Women Institute of Technology, Dehradun, India

Sinam Ajitkumar Singh Department of IT, Tripura University, Agartala, Tripura, India

Hemant Kumar Soni Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Madhya Pradesh, Maharashtra, Gwalior, MP, India

G. Sri Venkateswara Reddy Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Ulligaddala Srinivasarao Department of Computer Science and Engineering, National Institute of Technology Raipur, Raipur, Chhattisgarh, India

Kashish Srivastava School of Computer Science, University of Petroleum and Energy Studies, Bidholi, Dehradun, India

Prateek Thakral Department of Computer Science and Engineering, Jaypee University of Information Technology, Solan, India

Vaibhav Thakur Electronics and Communication Engineering Department, National Institute of Technology, Hamirpur, India

Aryav Tiwari Electronics and Communication Engineering Department, National Institute of Technology, Kurukshetra, India

Pankaj Verma Department of Electronics and Communication Engineering, NIT, Kurukshetra, Haryana, India

Rahul Kumar Verma Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India

T. Vinay Kumar Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

Om Prakash Yadav PES Institute of Technology and Management, Shivamogga, Karnataka, India

Rajesh Yadav Department of ECE, NIT Delhi, Delhi, India

Ravi Yadav Electronics and Communication Engineering, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Madhya Pradesh, India

Part I

Intelligent Communication Systems

Chapter 1

Spectral Efficiency Analysis of Massive MIMO Systems



Vaibhav Thakur and Manoranjan Rai Bharti

1 Introduction

In the present time, wireless connectivity has become one of the most important part of our lives. A huge amount of voice and wireless data communications is required these days. This interest for wireless data connectivity will keep on expanding for a long period of time, and maybe forever, since we are moving into a generation where every electronic gadget associates with the Internet. To deal with this issue of exponential growing traffic rate and simultaneously provide good connection, new progressive technologies are required. To compete with the data traffic growth, the main target of the 5G advancements is to improve the area throughput by 100 and even 1000 times [1]. The area throughput of a remote system is estimated in bit/s/km² and can be defined as follows:

$$\text{Area throughput} [\text{bit/s/km}^2] = \text{Bandwidth} [\text{Hz}] * \text{Cell Density} [\text{cells/km}^2] * \text{Spectral Efficiency} [\text{bit/s/Hz/cell}].$$

There are mainly three methods to improve the area throughput of cellular systems [2]: (1) Allocating larger bandwidth; (2) densifying the system by using more base stations; (3) enhancing the spectral efficiency per cell. However, bandwidth cannot be increased without limits, as the spectrum is one of the costliest things in universe. Cell density is the number of base stations (BS) inside the cell. Expanding the quantity of base stations is likewise not doable as it is costly to deploy new BSs. Spectral

V. Thakur (✉) · M. R. Bharti

Electronics and Communication Engineering Department, National Institute of Technology, Hamirpur, India

e-mail: ec15mi412@nith.ac.in

M. R. Bharti

e-mail: manoranjan@nith.ac.in

efficiency is the measure of data that can be transferred per second over one Hz of bandwidth. It is noticed that the enhancements in area throughput in past systems have largely been achieved via cell densification and designation of more bandwidth. It is also been noticed that the spectral efficiency (SE) has not seen any significant upgrades in past wireless communication systems. Therefore, it may be a factor that can be incredibly improved for current systems and has the potential to become the method of choice to accomplish high area throughput in 5G systems.

Massive MIMO is one of the finest approaches to improve SE [3], and furthermore, it works with systems having huge number of antennas as in a 5G framework. Hence, massive MIMO approach will be a better solution for improving SE for 5G systems. There are few techniques proposed in literature to improve the spectral efficiency of massive MIMO systems [4–6]. Optimizing the number of antenna is reported in [3, 7, 8]. The cellular system has several base stations which operate in a coherent fashion. It provides better array gain and spatial resolution that allows maintaining robustness to inter cell interference [9]. The aggressive multiplexing in massive MIMO systems contributes for the betterment of overall efficiency [8–11]. Resource allocation with the aim to improve SE for massive MIMO systems has been discussed in [12]. The impact of pathloss and spatial correlation with respect to UE interference is given in [12–14]. In this paper, the impact of various factors that affect the uplink spectral efficiency of a massive MIMO system has been studied, and then these factors are used to maximize the spectral efficiency of massive MIMO wireless communication systems.

2 Related Work

The uplink scenario of single cell multi-user massive MIMO system has been considered. The system comprises of BS arranged with M number of antennas and K number of single antenna users. The time frequency resources are divided into frames consisting of T_C s and W_C Hz. This leaves room for $S = T_C W_C$ transmission symbols per frame. Having characterized massive MIMO, we will now characterize the UL system models that are studied further.

2.1 Uplink System Model

The uplink transmission in massive MIMO is represented in Fig. 1. The received UL signal y_j at BS j is displayed as:

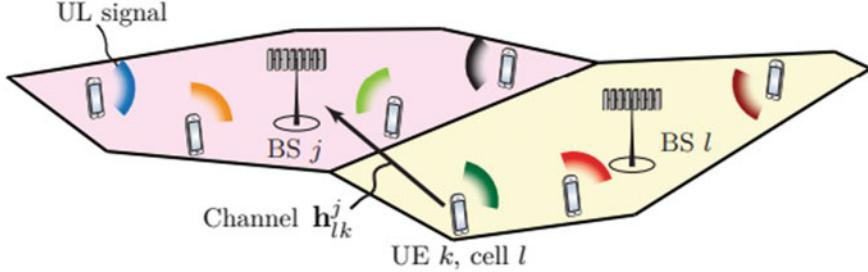


Fig. 1 Uplink massive MIMO transmission in cell j and cell l

$$\mathbf{y}_j = \sum_{k=1}^{K_j} \mathbf{h}_{jk}^j s_{jk} + \sum_{\substack{l=1 \\ l \neq j}}^L \sum_{i=1}^{K_l} \mathbf{h}_{li}^j s_{li} + \mathbf{n}_j \quad (1)$$

where \mathbf{n}_j is independent additive receiver noise with zero mean and variance σ_{UL}^2 . The uplink signal from UE k in cell l is given by s_{lk} and has power $p_{lk} = E\{|s_{lk}|^2\}$. The channel vector between BS j and UE k in cell l is referred as \mathbf{h}_{lk}^j . During data transmission, the BS in cell j selects the receive combining vector \mathbf{v}_{jk} to separate the signal from its k th desired UE from the interference as:

$$\mathbf{v}_{jk}^H \mathbf{y}_j = \mathbf{v}_{jk}^H s_{jk} \mathbf{h}_{jk}^j + \sum_{\substack{i=1 \\ i \neq k}}^{K_j} \mathbf{v}_{jk}^H \mathbf{h}_{ji}^j s_{ji} + \sum_{l=1}^L \sum_{\substack{i=1 \\ l \neq j}}^{K_l} \mathbf{v}_{jk}^H \mathbf{h}_{li}^j s_{li} + \mathbf{v}_{jk}^H \mathbf{n}_j \quad (2)$$

2.2 Uplink Pilot Transmission

τ_p samples are kept for uplink pilot signaling in every coherence block. Each user transmits a pilot sequence that spans these τ_p samples. The pilot sequence of user k in cell j is given by φ_{jk} . It is assumed to have unit-magnitude elements so as to attain a fixed power level, and this gives $\|\varphi_{jk}\|^2 = \varphi_{jk}^H \varphi_{jk} = \tau_p$. The elements of φ_{jk} are scaled by the uplink transmit power as $\sqrt{p_{jk}}$ and then transmitted as the signal s_{jk} in (1) over τ_p UL samples, giving the received UL signal \mathbf{Y}_j^p at BS j . This signal is given by:

$$\mathbf{Y}_j^p = \sum_{k=1}^{K_j} \sqrt{p_{jk}} \mathbf{h}_{jk}^j \varphi_{jk}^T + \sum_{\substack{l=1 \\ l \neq j}}^L \sum_{i=1}^{K_l} \sqrt{p_{li}} \mathbf{h}_{li}^j \varphi_{li}^T + \mathbf{N}_j^p \quad (3)$$

The first term in Eq. (3) represents the desired pilot, second term represents interfering pilots and the last term represents noise. \mathbf{N}_j^p is the independent additive receiver noise with i.i.d. distributed elements. \mathbf{Y}_j^p is the observation that BS j can utilize to estimate the channel responses.

Suppose, for the sake of argument, that BS j wants to estimate the channel \mathbf{h}_{li}^j from an arbitrary user i in cell l . The base station can then multiply/correlate \mathbf{Y}_j^p with the pilot sequence φ_{li} of this UE, leading to the processed received pilot signal y_{jli}^p , given as:

$$y_{jli}^p = \mathbf{Y}_j^p \varphi_{li}^* = \sum_{l'=1}^L \sum_{i'=1}^{K'_l} \sqrt{p_{l'i'}} \mathbf{h}_{l'i'}^j \varphi_{l'i'}^T \varphi_{li}^* + \mathbf{N}_j^p \varphi_{li}^* \quad (4)$$

which has the same dimension as \mathbf{h}_{li}^j . For the k th user in the base station's own cell, (4) can be expressed as:

$$\begin{aligned} y_{jjk}^p &= \mathbf{Y}_j^p \varphi_{jk}^* = \sqrt{p_{jk}} \mathbf{h}_{jk}^j \varphi_{jk}^T \varphi_{jk}^* \\ &\quad + \sum_{\substack{i=1 \\ i \neq k}}^{K_j} \sqrt{p_{ji}} \mathbf{h}_{ji}^j \varphi_{ji}^T \varphi_{jk}^* \\ &\quad + \sum_{\substack{l=1 \\ l \neq j}}^L \sum_{i=1}^{K_l} \sqrt{p_{li}} \mathbf{h}_{li}^j \varphi_{li}^T \varphi_{jk}^* + \mathbf{N}_j^p \varphi_{jk}^* \end{aligned} \quad (5)$$

Ideally, it is desired that all pilot sequences are orthogonal, but since the pilots are τ_p -dimensional vectors, for a given τ_p , we only have a set of at most τ_p mutually orthogonal sequences. It is assumed that the network utilizes a set of τ_p mutually orthogonal pilot sequences. These can be gathered as the columns of the uplink pilot book Φ , which satisfies $\Phi^H \Phi = \tau_p \mathbf{I}_{\tau_p}$. We define the set:

$$P_{jk} = \{(l, i) : \varphi_{li} = \varphi_{jk}, \quad l = 1, \dots, L \text{ and } i = 1, \dots, K_l\} \quad (6)$$

with the indices of all users that use the same pilot sequence as user k in cell j . Hence, $(l, i) \in P_{jk}$ means that UE i in cell l utilizes the same pilot as user k in cell j . Also

$(j, k) \in P_{jk}$ by definition. Using the notation in (6), the expression in (5) gives:

$$y_{jik}^p = \sqrt{p_{jk}} \mathbf{h}_{jk}^j \tau_p + \sum_{(l,i) \in P_{jk}(j,k)} \sqrt{p_{li}} \tau_p \mathbf{h}_{li}^j + \mathbf{N}_j^p \varphi_{jk}^* \quad (7)$$

The first term in Eq. (7) represents the desired pilot, second term represents the interfering pilots and the last term represents the noise.

3 Channel Estimation and Uplink Spectral Efficiency

3.1 Channel Estimation

Minimum mean square error (MMSE) channel estimation will be used. An estimator of channel response \mathbf{h}_{li}^j will be derived, based on the received pilot signal \mathbf{Y}_j^p in (3). The MMSE estimator of \mathbf{h}_{li}^j is the vector $\hat{\mathbf{h}}_{li}^j$ that minimizes the MSE $\mathbf{E}\left\{\|\mathbf{h}_{li}^j - \hat{\mathbf{h}}_{li}^j\|^2\right\}$.

Using a pilot book with mutually orthogonal sequences, the MMSE estimate of the channel \mathbf{h}_{li}^j based on the observation \mathbf{Y}_j^p in (3) is:

$$\hat{\mathbf{h}}_{li}^j = \sqrt{p_{li}} \mathbf{R}_{li}^j \Psi_{li}^j \mathbf{y}_{jli}^p \quad (8)$$

where

$$\Psi_{li}^j = \left(\sum_{(l',i') \in P_{li}} p_{l'i'} \tau_p \mathbf{R}_{l',i'}^j + \sigma_{\text{UL}}^2 \mathbf{I}_{M_j} \right)^{-1} \quad (9)$$

The estimation error $\mathbf{h}_{li}^j - \hat{\mathbf{h}}_{li}^j$ has correlation matrix $\mathbf{C}_{li}^j = \mathbf{E}\left\{\mathbf{h}_{li}^j (\mathbf{h}_{li}^j)^H\right\}$, given by:

$$\mathbf{C}_{li}^j = \mathbf{R}_{li}^j - p_{li} \tau_p \mathbf{R}_{li}^j \Psi_{li}^j \mathbf{R}_{li}^j \quad (10)$$

This explains the procedure to calculate the MMSE estimate of the channel from any user in the system to base station j . The estimation quality is represented by the MSE, which is $\mathbf{E}\left\{\|\mathbf{h}_{li}^j - \hat{\mathbf{h}}_{li}^j\|^2\right\}$ for the MMSE estimator. A small MSE represents a favorable estimation quality.

To estimate \mathbf{h}_{li}^j based on (8), the base station should correlate the received pilot signal with the pilot sequence used by user i in cell l , as $\mathbf{y}_{jli}^p = \mathbf{Y}_j^p \varphi_{li}^*$, and then multiply this observation with the two matrices Ψ_{li}^j and \mathbf{R}_{li}^j . The matrix Ψ_{li}^j is the

inverse of the normalized correlation matrix $E\left\{y_{jli}^p \left(y_{jli}^p\right)^H\right\}/\tau_p$ of the processed received signal while \mathbf{R}_{li}^j is the spatial correlation matrix of the channel to be estimated. These multiplications suppress interference and noise that do not share the same second order statistics as \mathbf{h}_{li}^j . The MMSE estimator in (8) is linear because $\hat{\mathbf{h}}_{li}^j$ is formed by multiplying the processed received signal y_{jli}^p with matrices. The estimator used here is hence mostly referred as the linear MMSE estimator (LMMSE).

3.2 Uplink Spectral Efficiency

If MMSE channel estimation is used, then the UL ergodic channel capacity of user k in cell j is lower bounded by $\text{SE}_{jk}^{\text{UL}}$ [bit/s/Hz] and is determined as:

$$\text{SE}_{jk}^{\text{UL}} = \frac{\tau_u}{\tau_c} E\{\log_2(1 + \text{SINR}_{jk}^{\text{UL}})\} \quad (11)$$

where

$$\begin{aligned} \text{SINR}_{jk}^{\text{UL}} &= \frac{p_{jk} \left| \mathbf{v}_{jk}^H \hat{\mathbf{h}}_{jk}^j \right|^2 \mathbf{M}_j}{\sum_{l=1}^L \sum_{i=1}^{K_l} p_{li} \left| \mathbf{v}_{jk}^H \hat{\mathbf{h}}_{li}^j \right|^2 + \mathbf{v}_{jk}^H \left(\sum_{l=1}^L \sum_{i=1}^{K_l} \mathbf{C}_{li}^j p_{li} + \sigma_{\text{UL}}^2 \mathbf{I}_{M_j} \right) \mathbf{v}_{jk}} \\ &\quad (l, i) \neq (j, k) \end{aligned} \quad (12)$$

Equation (12) represents an achievable SE for the UL. The prelog factor τ_u/τ_c in (11) is the fraction of samples per coherence block that are used for UL data. Since $\tau_u = \tau_c - \tau_p - \tau_d$, the prlog factor increases if we decrease the length τ_p of the pilot sequences (i.e., reduce the pilot overhead) and/or reduce the number of samples τ_d used for downlink data.

The spectral efficiency expression given here holds for any choice of the receive combining vector, under the assumption that the MMSE estimator is used for channel estimation. Maximal ratio (MR) combining with $\mathbf{v}_{jk} = \hat{\mathbf{h}}_{jk}$ or zero-forcing (ZF) combining with $\mathbf{v}_{jk} = \hat{\mathbf{h}}_{jk} \left(\left(\hat{\mathbf{h}}_{jk}^H \right) \hat{\mathbf{h}}_{jk} \right)$ is commonly considered in the massive MIMO.

4 Simulation Results

A square cell with a side of 250 m has been considered. The users are uniformly distributed at distances to the BS of at least 35 m. Pilot reuse factor of $f = \{1, 3, 4\}$ has been selected for simulations. The coherence block length is $\tau_c = 400$, which supports high user mobility.

The uplink SE of ZF and MR receivers is shown in Fig. 2. The average uplink SE as a function of the number of base station antennas has been plotted for universal pilot reuse with $f = 1$. There are $K = 10$ UEs per cell and a coherence block length of $\tau_c = 400$ symbols.

It has been found in [15] that MR provides a lower complexity than that of ZF; however, it can be seen in Fig. 2 that it also provides lower SE. On the other hand, ZF has a computational complexity which is only some tens of percent higher [15] but the SE is double than that of MR, which is the main concern here. Hence, ZF receive combining is preferred over MR and only ZF has been simulated in further results.

Next, the SE is simulated with different pilot reuse factors. The parameters are same as for Fig. 2 except that 3 pilot reuse factors have been considered, i.e., $f = \{1, 3, 4\}$.

It was noticed that different pilot reuse factors are desired for different number of Base Station antennas. A pilot reuse of $f = 3$ is desirable when the number of BS antennas (and hence K) [16] is less, while $f = 1$ is required to decrease the prelog factor $(1 - fK/\tau_c)$ when K is large. By selecting f accordingly, one can always operate on the top curve in Fig. 3 and then Massive MIMO can provide a high SE over a wide range of different number of BS antennas.

In Fig. 4, SE has been simulated against the number of users for $M = 100$ and 500 to find out the optimal number of users for these values of M . ZF combining has been used and the peak numbers that are star marked are the optimal number of

Fig. 2 UL spectral efficiency as a function of the number of BS antennas for ZF and MR combining schemes

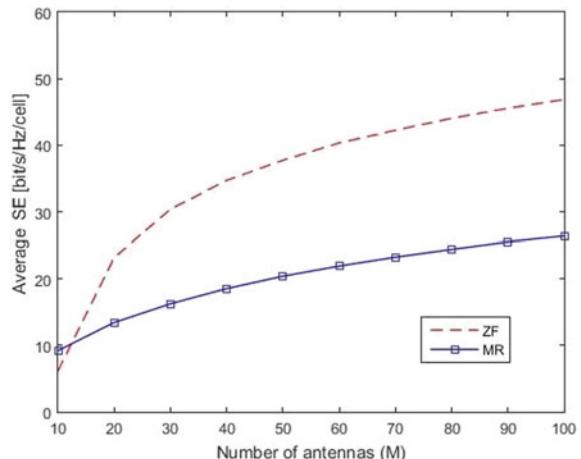


Fig. 3 Spectral efficiency as a function of the number of BS antennas and pilot reuse factors

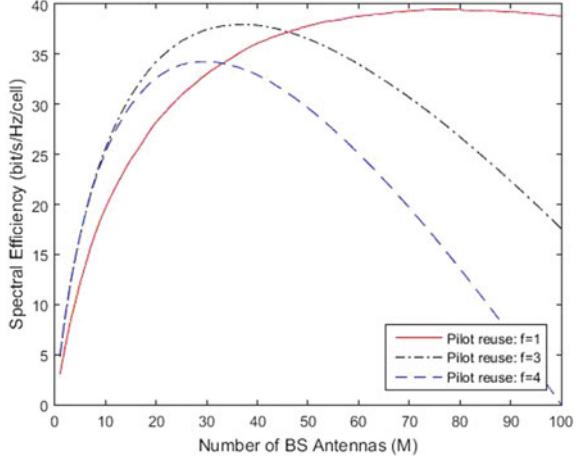
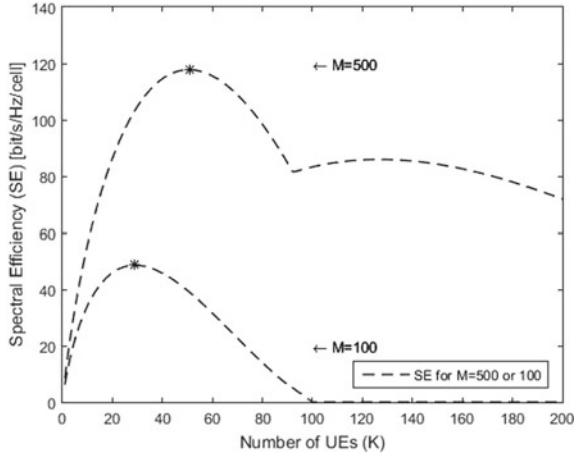


Fig. 4 Spectral efficiency as a function of the number of users



users for the given value of M . $f = 1$ has been selected as for larger M , this value provides better results (see Fig. 3).

Figure clearly shows that for $M = 100$, the optimized number of users are 30, whereas for $M = 500$, the optimized UEs are 50. This shows that as the number of BS antennas M increases, more number of users can be served by the system. Moreover, this also gives the result that for a particular value of M (e.g., 100), if optimal number of users are considered, the highest SE will be achieved. Hence, a number of users can be optimized according to the particular value of M to achieve highest spectral efficiency.

Next, SE is simulated against the coherence block length for $M = 100$ and 500 for ZF combining scheme. $f = 1$ has been selected as for larger M , this value provides better results (see Fig. 3).

Fig. 5 Spectral efficiency as a function of coherence block length

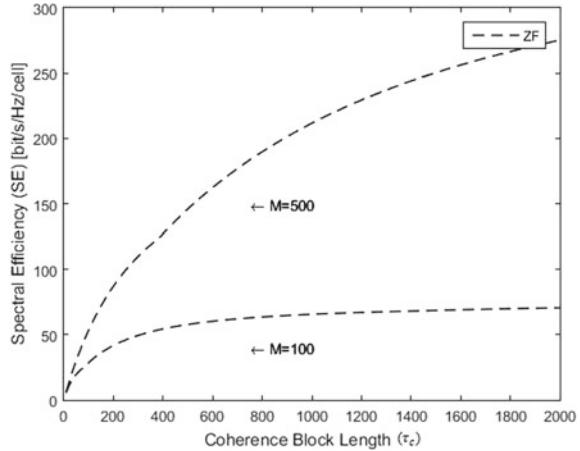
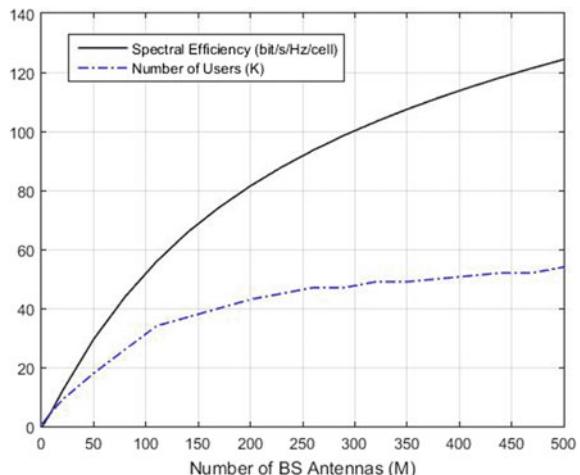


Figure 5 investigates how the coherence block length, τ_c , affects the per cell SE. For $M = 100$ antennas, the achievable gain by increasing τ_c above 500 is relatively small—the system cannot schedule more users since the ratio M/K would then be too small, so the gain mainly comes from reducing the prelog factor $(1 - \tau_p / \tau_c)$. However, in the case of $M = 500$, the system can utilize an increasing τ_c to schedule more UEs and achieve significant improvements in SE. As the number of users increases, the part of the intra-cell interference that cannot be rejected due to imperfect CSI becomes the main limiting factor. Hence, coherence block length cannot be increased above a certain limit.

In Fig. 6, the results of previous figures have been combined. Hence, a pilot reuse factor of $f = 1$ has been selected for large values of M and $f = 3$ for smaller M . The number of users has been optimized alongside M so as to attain the best possible

Fig. 6 Proposed spectral efficiency as a function of the number of BS antennas



spectral efficiency. Also, it was concluded that larger coherence block length (τ_c) was helpful but this value could not be increased above a certain limit. Hence, the value of τ_c was chosen to be 400.

Figure 6 depicts the spectral efficiency as a function of the number of BS antennas M . The number of active UE is optimized for each M to get the highest spectral efficiency and the optimal user numbers are also depicted in the figure. A reasonable performance baseline is the IMT-advanced requirements for 4G networks, provided in [17]. This document specifies spectral efficiencies in the range of 2–3 bit/s/Hz/cell, depending on the simulation scenario. In comparison, the massive MIMO network simulated in Fig. 6 achieves 52 bit/s/Hz/cell using $M = 100$ antennas, which is a 16–25 times improvement over IMT-advanced. This occurs due to a greater number of antennas in massive MIMO systems. With $M = 400$ antennas the Massive MIMO system achieves 113 bit/s/Hz/cell, which is an incredible 37–56 times improvement over IMT-advanced.

Furthermore, compared to the research work in [18], wherein the spectral efficiencies for massive MIMO systems using FBMC-OQAM modulation are found to be around 70 bit/s/Hz/cell for 500 BS antennas, the model simulated in Fig. 6 achieves 123 bit/s/Hz/cell for same number of BS antennas.

5 Conclusion

In the paper, massive MIMO systems have been studied, and it is observed that massive MIMO is the most promising technology for 5G wireless communication systems. It gives high data rates and communication reliability. Firstly, achievable uplink rate using zero-forcing and maximum ratio combining receivers has been evaluated. It was observed that the zero-forcing receiver outperforms the maximum ratio combining receiver in terms of spectral efficiency, and hence, this receiving technique was used for further simulations. After this, the impact of various system parameters like number of base station antennas (M), number of users (K), pilot reuse factor (f), and coherence block length (τ_c) on SE of massive MIMO systems was studied. Then, a model was proposed to maximize the spectral efficiency by using appropriate values of the system parameters mentioned above. It is shown that the proposed massive MIMO model provides a spectral efficiency which is higher than that provided by the IMT-advanced techniques. The simulation results presented in this paper can be helpful in estimating the spectral efficiency of massive MIMO systems employing different values of system parameters and receiving techniques for 5G wireless communications.

References

1. Andrews JG, Buzzi S, Choi W, Hanly SV, Lozano A, Soong ACK, Zhang JC (2014) What will 5 g be? *IEEE J Sel Areas Commun* 32(6):1065–1082
2. Nokia Siemens Networks. 2020: beyond 4G radio evolution for the Gigabit experience. White Paper, Technical Report, 201
3. Marzetta TL (2010) Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Trans Wirel Commun* 9(11):3590–3600
4. Baldemair R, Dahlman E, Fodor G, Mildh G, Parkvall S, Selen Y, Tullberg H, Balachandran K (2013) Evolving wireless communications: addressing the challenges and expectations of the future. *IEEE Veh Technol Mag* 8(1):24–30
5. Boccardi F, Heath R, Lozano A, Marzetta T, Popovski P (2014) Five disruptive technology directions for 5G. *IEEE Commun Mag* 52(2):74–80
6. Larsson EG, Tufvesson F, Edfors O, Marzetta TL (2014) Massive MIMO for next generation wireless systems. *IEEE Commun Mag* 52(2):186–195
7. Hoydis J, ten Brink S, Debbah M (2013) Massive MIMO in the UL/DL of cellular networks: how many antennas do we need? *IEEE J Sel Areas Commun* 31(2):160–171
8. Ngo HQ, Larsson EG, Marzetta TL (2013) Energy and spectral efficiency of very large multiuser MIMO systems. *IEEE Trans Commun* 61(4):1436–1449
9. Ha D, Lee K, Kang J (2013) Energy efficiency analysis with circuit power consumption in massive MIMO systems. In: Proceedings of IEEE international symposium personal, indoor and mobile radio communications
10. Björnson E, Sanguinetti L, Hoydis J, Debbah M (2015) Optimal design of energy-efficient multi-user MIMO systems: is massive MIMO the answer. *IEEE Trans Wirel Commun* 14(6):3059–3075
11. Yang H, Marzetta T (2013) Total energy efficiency of cellular large scale antenna system multiple access mobile networks. In: Proceedings of IEEE online conference on green communications
12. Huh H, Caire G, Papadopoulos H, Ramprashad S (2012) Achieving “massive MIMO” spectral efficiency with a not-so-large number of antennas. *IEEE Trans Wirel Commun* 11(9):3226–3239
13. Li M, Nam Y-H, Ng B, Zhang J (2012) A non-asymptotic throughput for massive MIMO cellular uplink with pilot reuse. In: Proceedings of IEEE global communications conference (GLOBECOM)
14. Müller R, Vehkaperä M, Cottatellucci L (2013) Blind pilot decontamination. In: Proceedings of WSA
15. Björnson E, Hoydis J, Sanguinetti L (2017) Massive MIMO networks: spectral, energy, and hardware efficiency. In: Foundations and trends® in signal processing, vol. 11(3–4), pp 154–655. <https://doi.org/10.1561/2000000093>
16. Cox D (1982) Cochannel interference considerations in frequency reuse small-coverage-area radio systems. *IEEE Trans Commun* 30(1):135–142
17. ITU: Requirements related to technical performance for IMT-advanced radio interface(s). Technical report, ITU-R M.2134
18. Jose FK, Lolis LH, Mafra SB, Ribeiro EP (2018) Spectral efficiency analysis in massive MIMO using FBMC-OQAM modulation. *J Microwaves Optoelectron Electro Magn Appl* 604–618. <https://doi.org/10.1590/2179-10742018v17i41544>

Chapter 2

Deep Learning-Enabled Interference Management in LTE-D2D Communication



Sandeepika Sharma and Brahmjit Singh

1 Introduction

Progression of ubiquitous wireless access technologies and unprecedented growth of networked devices with data-intensive applications has resulted in huge mobile data traffic. This data traffic is anticipated to hit 131 Exabyte/month by 2024. Around 35% of this volume is expected to be carried over 5G networks [1]. Presently, available wireless technologies are not adequate to achieve the key performance indicators set by 3GPP for 5G wireless networks. Innovative solutions are required in order to increase the capacity of existing LTE-A network by 1000 times, to provide data transmission rate of 1 Gbps for users with high mobility and 10 Gbps for stationary or pedestrian users along with end-to-end latency in the order of 1 ms [2].

In order to improve the capacity of the existing cellular networks utilizing low-band frequencies, efficient utilization of available radio resources and co-channel interference management are seen as the critical issues to be resolved. 5G networks using low-band frequencies are termed as new radio (NR) [3]. The device-to-device (D2D) communication underlaying cellular network is one of the key technologies to be adopted for spectrum sharing in 5G NR. It allows direct content transfer among cellular users instead of routing the data through the base station. D2D was adopted as a feature in 3GPP Rel.12 in 2014 for supporting proximity services (Pro-Se) in public safety network or commercial applications [4]. Pro-Se allows two different modes of direct communication. The first mode 1 supports centralized BS controlled communication in which BS or cluster head is responsible for allocation of channels among D2D pairs. The second mode 2 supports distributed control where each D2D

S. Sharma (✉) · B. Singh
ECE Department, NIT Kurukshetra, Kurukshetra 136119, India

B. Singh
e-mail: brahmjit@nitkkr.ac.in

pair can select its frequency channel autonomously with partial or no assistance of BS or cluster head. This paper focuses on interference management in mode 2-type D2D communication.

5G NR will evolve into heterogeneous and ultra-dense network requiring large number of radio resources and much faster data transfer to support smart home, smart city and smart transportation applications. It is evident from the available literature that due to increasing number of socially aware cellular and D2D users using data-hungry applications, efficient radio resource and interference management have become a challenge for successful implementation of in-band D2D communication. In such scenario, the intelligent channel selection is required to provide high reliability and accuracy. Getting motivation from this observation, we propose a deep learning-based autonomous resource selection scheme, which utilizes sequence-to-sequence regression LSTM network and predicts the interference over available channel for next transmission time interval.

The remainder of the paper is structured as follows. Section 2 presents a review of related work on interference management techniques used for D2D communication. Section 3 provides system model and fundamentals of autonomous resource selection procedure in LTE-D2D communication. Section 4 discusses basics of regression LSTM network and presents the proposed scheme. Simulation results are presented and analyzed in Sect. 5. Finally, conclusions are drawn in Sect. 6.

2 Related Work

Distributed power control in D2D system is modeled as a multi-agent learning game in [5]. It is assumed that after channel allocation by the base station, D2D users will utilize Q-learning algorithm to learn the optimal transmit power value so as to generate less interference over other users using the same channel. Authors in [6] have formulated channel selection problem as a non-cooperative game and uses regret matching learning algorithm to choose the channel which maximizes the overall D2D system throughput. Such a strategy does not require much interaction among the players, i.e., D2D users.

The cellular learner automata method is adopted in [7] for simultaneous allocation of channels in underlay D2D communication system. A single-state multi-agent reinforcement learning scheme is proposed in [8] for autonomous channel selection. It assumes cooperation among users who share their Q-values with the neighboring D2D users within the cooperation range. Authors in [9] have applied graph-based approach for non-orthogonal channel allocation in two scenarios, first in which a D2D pair can be allocated at most one channel and second in which a D2D pair can be allocated multiple channels which are shared among other D2D and cellular users. Authors in [10] have proposed a genetic algorithm-based scheme to mitigate interference among D2D users without the knowledge of channel state information and improves the spectral efficiency of D2D system. They have utilized the channel prediction to reduce CSI overhead. Power control, resource allocation and relay node

selection in cooperative D2D heterogeneous network are performed simultaneously in [11] using quantum coral reef optimization algorithm.

Authors in [12] have opted deep neural network-based approach for predicting D2D channel gains for efficient resource allocation in D2D system. It utilizes correlation between D2D and cellular channel gains. The efficiency of the proposed algorithm is measured in terms of Pearson correlation coefficient between true and predicted values of channel gain. Cooperative reinforcement learning-based channel and power control scheme are proposed in [13]. It is shown through simulation results that proposed scheme improves D2D throughput by 15%.

3 System Model

In this work, we focus on sidelink transmission of messages among neighboring cellular users utilizing D2D communication under LTE-A network as shown in Fig. 1. It is assumed that D2D users will share their channels with other cellular users coexisting within the same LTE-A network. Each D2D user will select its channel autonomously by using the installed interference prediction module. D2D receiver and transmitter can be within an area of given radius hereafter termed as D2D radius. D2D radius value will be predefined by the base station, and this will affect the level of interference on D2D as well coexisting cellular users. Effect of D2D radius on performance metrics is discussed in Sect. 5.

When a D2D user has a new message to transmit, it needs to select a group of adjacent channels within a time period less than the maximum allowed latency of the application.

This resource selection process in mode 2 communication involves following steps:

- Step 1: Power Sensing

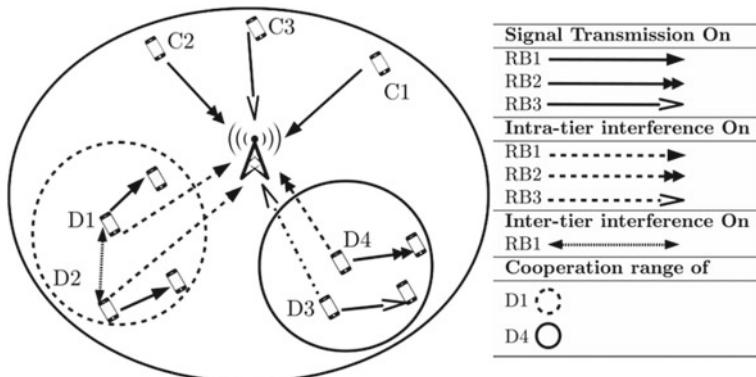


Fig. 1 Channel sharing among D2D users in LTE-D2D mode 2-type communication system

The D2D transmitter measures the received signal power over all the channels which it can reserve for next transmission. Large value of received signal power signifies high interference level over that particular channel.

- Step 2: Channel Categorization

A candidate channel list is prepared on the basis of intensity of interference obtained over the available channels. A channel is excluded from the candidate list if received signal power over it is greater than the threshold value. If number of channels in the candidate list is less than 20% of the total channels available, then the threshold value is incremented by 3 dB. This step is repeated until the candidate list is populated by the required number of channels.

- Step 3: Channel Selection

D2D transmitter ranks the channels in candidate list in ascending order based on the received signal power over them. Then, it selects a channel randomly for its next transmission.

4 The LSTM Network

Artificial neural network (ANN)-based solutions have shown state-of-the-art performances in classification and sequence prediction tasks. ANNs can be of several types, but in this work, we have used long short-term memories (LSTM) architecture. LSTM is a recurrent neural network (RNN) which deals with continuous and time-dependent information. LSTM overcomes the issue of vanishing/exploding gradient which occurs in RNN. Each memory cell has an input gate, forget gate and an output gate. These gates are used to update the information stored within the LSTM memory cell. It is assumed that $x(t)$ denotes the input sequence at time t , h_{t-1} denotes the output of the LSTM cell at $t - 1$, h_t is the output of the cell at time t , and C_t represents the state of the cell at time t . Output of the LSTM cell is computed using the following equations:

$$F_t = \sigma(W_F^i \cdot x(t) + W_F^h \cdot h_{t-1} + b_f) \quad (1)$$

$$I_t = \sigma(W_I^i \cdot x(t) + W_I^h \cdot h_{t-1} + b_I) \quad (2)$$

$$D_t = \tanh(W_D^i \cdot x(t) + W_D^h \cdot h_{t-1} + b_D) \quad (3)$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot D_t \quad (4)$$

$$O_t = \sigma(W_O^i \cdot x(t) + W_O^h \cdot h_{t-1} + b_O) \quad (5)$$

$$h_t = O \cdot \tanh(C_t) \quad (6)$$

Here W_F^i , W_I^i , W_D^i and W_O^i denote weight matrices connecting $x(t)$ to forget gate, input gate, cell input and output gate, respectively. W_F^h , W_I^h , W_D^h and W_O^h represent weight matrices connecting h_{t-1} to three gates and cell input. These weights are learned during the training process. b_f , b_i , b_c and b_o are the bias vectors. σ and \tanh represent sigmoid and hyperbolic tangent activation function, respectively. Hadamard product of F_t with C_{t-1} in Eq. (4) decides which information from previous cell state needs to be retained or deleted. Similarly, product of I_t with D_t decides which information from current cell state needs to be deleted or retained in C_t . \tanh transforms the information in C_t between -1 and 1 . The Hadamard product in Eq. (6) decides on the information from current cell state C_t to be present in the output.

LSTM has been used for sequence-to-sequence prediction problems. Each problem is solved by learning a mapping function which takes an input sequence and internal state values to predict an output sequence. The problem is solved by learning a mapping function which takes an input sequence and internal state values to predict an output sequence. The mapping function can be one-to-one which generates single output value for single input value; one-to-many which generates multiple output values for single input value; many-to-one which generates single output value for multiple input values and many-to-many which provides multiple output values for multiple input values. In this work, we present the strategy to predict interference value for next transmission time based on past values. For doing so, we have utilized many-to-one prediction model. Further, in the proposed scheme, a single interference value is predicted by the averaging of values in LSTM output sequence. The weighted exponential averaging is utilized for the purpose, and its performance is compared with linear averaging.

5 Numerical Results and Their Analysis

In this section, we present the comparative performance evaluation of the proposed resource selection scheme through MATLAB computing platform. We consider a single microcell of radius 1000 m with the base station located at its center. Table 1 gives values of the system parameters. It is assumed that the number of channels available in the resource pool is equal to the number of cellular users in the cell. A set of 50 channels is available for sharing among cellular and D2D users. Performance of our proposed algorithm LSTM + WEA is compared with LSTM + LA and LSTM-based resource selection schemes. LSTM + WEA refers to the weighted exponential averaging over LSTM output sequence, LSTM + LA refers to linear averaging over LSTM output sequence, and LSTM refers to predicting single time step value without averaging. For evaluation, normalized mean square error (NMSE), system reliability and packet error rate are used as the performance metrics.

Table 1 Simulation parameters

Parameter	Value
Cell radius	1000 m
Channel bandwidth	180 kHz
D2D radius	100 m
eNodeBTx power	46 dBm
CUE Tx power	26 dBm
DUE Tx power	13 dBm
Noise power	-174 dBm/Hz
Path loss (D2D Link)	$148 + 40 \log_{10}(d/1000)$
P_{th}	-5 dBm

Figure 2 illustrates NMSE as a function of awareness message size. NMSE depicts divergence between true and predicted value of interference level on available channels. It is observed that NMSE increases with increasing message size in case of LSTM + WEA ($w = 0.4$), LSTM + WEA ($w = 0.8$) and LSTM + LA. But it is constant in case of LSTM and LSTM + WEA ($w = 0.1$). “ w ” here denotes the weighing factor used in weighted exponential averaging. LSTM + WEA ($w = 0.4$) reduces RMSE by 62.5 and 88% as compared to LSTM + LA and LSTM, respectively. The setting of $w = 0.4$ provides the least error in prediction as compared to $w = 0.1$ or $w = 0.8$. It is due to the fact that it gives equal weight to recent as well as delayed interference values.

Figure 3 depicts packet error rate as a function of awareness message size. Packet error rate signifies number of erroneous packets among the total received packets by a D2D receiver. It is inferred from the results that packet error increases with the

Fig. 2 Normalized mean square error versus message size

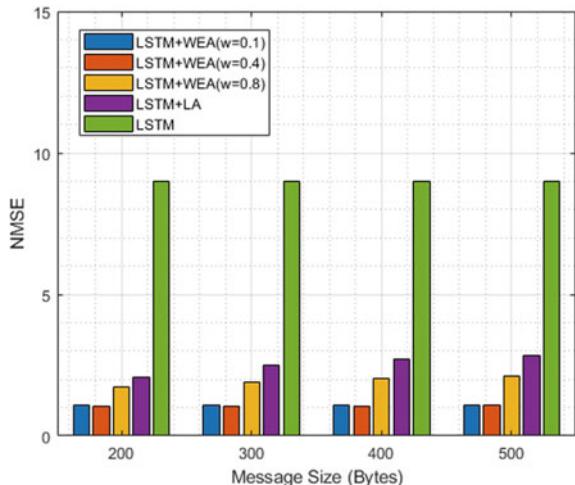
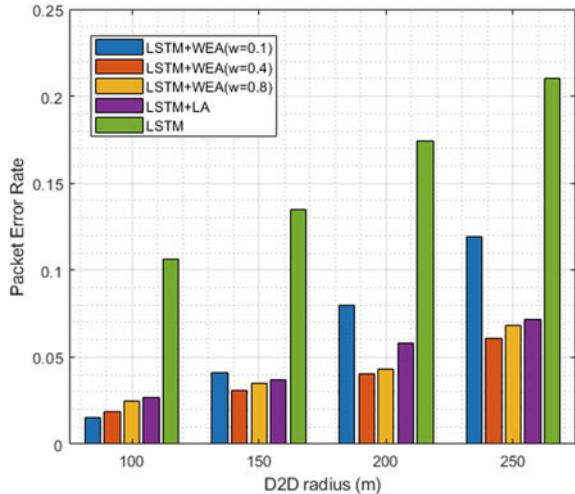


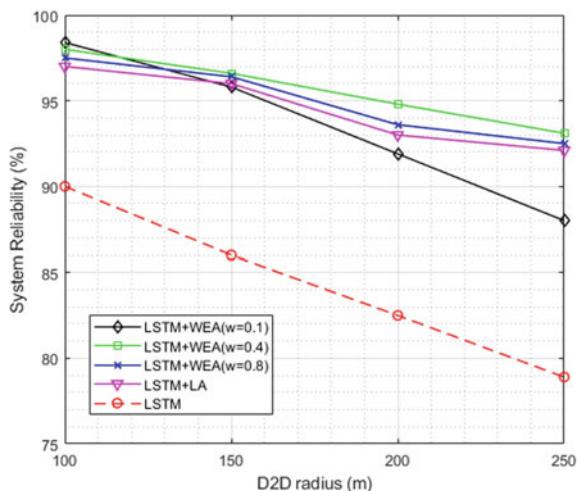
Fig. 3 Packet error rate as a function of D2D radius



increase in message size. This happens because large packet size requires reserving large number of resources.

This increases co-channel interference among the users, thereby reducing the D2D channel gain and average D2D user throughput leading to more number of errors. LSTM + WEA ($w = 0.4$) reduces packet error rate by 15.2 and 70.9% as compared to LSTM + LA and LSTM, respectively. Variation in system reliability with increasing D2D radius is shown in Fig. 4. The system reliability signifies percentage of packets received successfully among the total number of packet exchange between D2D pairs. Large separation distance between D2D transmitter and receiver reduces channel gain, thereby reducing D2D throughput. Therefore, system reliability decreases with

Fig. 4 System reliability as a function of D2D radius



increase in D2D radius for each resource selection scheme. LSTM + WEA ($w = 0.4$) reduces packet error rate by 4.6 and 15% as compared to LSTM + LA and LSTM, respectively.

6 Conclusion

In this research work, we have introduced an autonomous resource selection scheme for mode 2 D2D communication utilizing LTE-A network. The proposed LSTM-based strategy is based on time series forecasting enabled with deep learning and weighted exponential averaging. The interference prediction module installed at each D2D user predicts the interference level in order to select the best available channel. The efficacy of the proposed solution is established through numerical results in terms of system reliability, packet error rate and normalized mean square error. The system reliability is improved by 4.6% while reducing the packet error rate by 15.2% for 62.5% less prediction errors. The LSTM in conjunction with the weighted exponential averaging as such offers a reliable and accurate autonomous resource reselection strategy for the underlay D2D communication.

References

1. <https://www.ericsson.com/en/mobility-report/reports/june-2019>
2. Sharma S, Singh B (2016) 5G networks: The next gen evolution. International conference on signal processing and communication. IEEE, pp. 55–60
3. Iqbal AR, Chrysostomou C, Hassan SA et al (2017) 5G D2D networks: techniques, challenges, and future prospects. *IEEE Syst J* 12(4):3970–3984
4. Xingjin L, Andrews JG, Ghosh A, Ratasuk R (2014) An overview of 3GPP device-to-device proximity services. *IEEE Commun Mag* 52(4):40–48
5. Setareh M, Stańczak S (2015) Hybrid centralized-distributed resource allocation for device-to-device communication underlaying cellular networks. *IEEE Trans Veh Technol* 65(4):2481–2495
6. Lhazmir S, Kobbane A, Ben-Othman J (2018) Channel assignment for D2D communication: a regret matching based approach. In: International wireless communications and mobile computing conference (IWCMC), Limassol, Cyprus, pp 322–327
7. Farzanegan S, Jabbari A (2019) A novel method for assigning joint power spectrum and power selection in device to device networks to improve performance. *Majlesi J Telecommun Dev* 8(3)
8. Sharma S, Singh B (2019) Weighted cooperative reinforcement learning-based energy-efficient autonomous resource selection strategy for underlay D2D communication. *IET Commun* 13(14):2078–2087
9. Tseng Y-L (2015) LTE-advanced enhancement for vehicular communication. *IEEE Wirel Commun* 22(6):4–7
10. Hengameh T, Doğan G, Arslan H (2018) Joint optimization of device to device resource and power allocation based on genetic algorithm. *IEEE Access* 6:21173–21183
11. Hongyuan G, Zhang S, Su Y, Diao M (2019) Joint resource allocation and power control algorithm for cooperative D2D heterogeneous networks. *IEEE Access* 7:20632–20643

12. Mehyar N, Mach ZP, Gesbert D (2019) Predicting device-to-device channels from cellular channel measurements: a learning approach. [arXiv:1911.07191](https://arxiv.org/abs/1911.07191)
13. Sharma S, Singh B (2019) Cooperative reinforcement learning based adaptive resource allocation in V2V communication. International conference on signal processing and integrated networks (SPIN). IEEE, pp 489–494

Chapter 3

Traffic Violations Prediction System on the Basis of Human Behaviour



Deepti Goel, Rajesh Bhatia, and Kashish Bhatia

1 Introduction

Traffic violation is major cause of road accident in India. Traffic violations happen due to physical condition of road, driving behaviour of drivers, psychology of drivers, mental illness of drivers and the presence of passenger in car. However, most of violations happen due to psychological behaviour of drivers.

467,044 road accident of India has been reported in 2018 [1, 2]. Out of them, 469,418 people were injured in that road accidents happened in 2018, and 151,417 people were dead [1, 2]. Most of the time major cause is over speeding for road accidents [1, 2]. After over speeding, driving in wrong lane causes 5.8% of total causalities in 2018. Driving after consumption of alcohol and drugs, red lights jumping and usage of mobile phones leaded to 6.5% of total accidents. These violations also leaded to 6.2% of total deaths [3]. As compared to 2017, 0.46% road accidents have been increased in 2018 [4]. According to world road statistics, rank 1 is holding by India in road accidents' death over 199 countries [4].

Human factors that can cause traffic violation are as follows [5]:

- Socio-demographic characteristics—Age, income, gender, education, nationality, country, area etc.
- Personality—This can be measured by big five personality questionnaire which include five personalities, i.e. openness, conscientiousness, extraversion, agreeableness and neuroticism.
- Aggression.
- Impulsiveness.

D. Goel (✉) · R. Bhatia
PEC University of Technology, Chandigarh, India

K. Bhatia
UCOE, Punjabi University, Patiala, India

- Usage of alcohol while driving.
- Drug addiction.
- Psychological problems like PTSD.
- Stress or anxiety.

Traffic violation/road accident and stress/anxiety are highly related. According to research, one of five victims in road accidents are suffering from stress, and one of four are suffering from psychological problem like post-traumatic stress disorder [6]. Another major cause of road accident and traffic violation is road rage. A road rage refers to violent incidents arising out of stress and various psychological factors while driving on roadways or high-traffic areas. According to National Highway Traffic Safety Administration (NHTSA), 94% accidents are happened due to driving error, and 33% of them can be linked to behaviour of driver like road rage [7].

Traffic psychology has been introduced to raise awareness in drivers. Traffic psychology is quite new scientific discipline. Traffic psychologists are hired by government agencies and organisations working in traffic to reduce traffic risk.

We need a system that predicts ability of drivers to violate traffic rules. On the basis of their behaviour and tendency to violate traffic rules, they are referred to the counselling (traffic psychologist) to reduce traffic danger. In this study, traffic violation prediction system is developed by using machine learning like logistic regression, decision tree and artificial neural network algorithm to predict the ability of drivers to commit traffic violations on the basis of human behaviour and data analysis is done by using graphs, means and medians to find what traits are more common in the participants.

2 Related Work

There are many factors which can affect driving behaviour like demographic information, personality trait, mental illness and alcohol addiction. There are lot of psychology questionnaire to find out above parameters like personality trait and driving behaviour.

2.1 Factors Affecting Traffic Violations

Traffic violations or road accidents depend on many factors like culture, age, gender, physical condition of roads, education, marital status, personality, mental illness, addiction [8]. Human factors that have impact on traffic violations are fatigue, drowsiness, mental disorder, disability, driving style, lack of training for traffic safety and inappropriate method to obtain driving licence [3].

Factors like driving behaviour, personality trait, mental illness (PSTD, mental retardation and dissociative disorder), age group, driving skills, driving experience,

marital status, education, income, aggression, usage of mobile phone, wearing seat belts, anger and passenger presence were used in traffic violations' research [3, 5, 9].

2.2 Tools for Assessment

Different questionnaires, direct interview and simulators were used to assess personality trait, driving behaviour, drinking addiction and mental illness like NEO personality inventory for assessing personality trait, MDBQ for driving behaviour and SADS for mental illness [3].

2.3 Technology

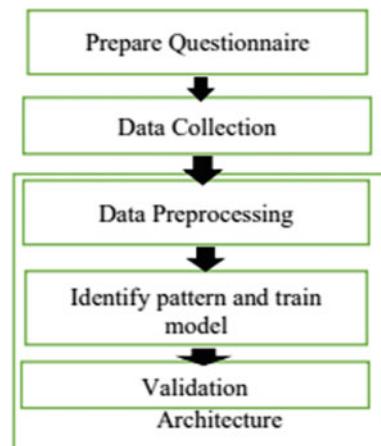
In most researches, SPSS software is come into play for data analysis. Techniques like t-test, chi-square, logistic regression, odd ratio, mediation analysis, bivariate correlation, anova model, descriptive statistics, multi-regression and tobit model were used for data analysis like inferential analysis and predictive analysis [3, 4, 6, 10–13].

3 Proposed Methodology

Steps of the proposed methodology are (as shown in Fig. 1):

1. To prepare questionnaire that gather demographic, personality and traffic behaviour information of participants.

Fig. 1 Proposed methodology of the research



2. To collect data that include demographic, personality and traffic behaviour information through Google form.
3. To prepare data by using data pre-processing techniques.
4. To identify patterns and train the model by using machine learning techniques (logistic regression).
5. Validation of result can be done with the help of collective data.

3.1 Questionnaire Preparation

There is IoT of questionnaires which can be used for assessing personality of human and driving behaviour of participants. However, in this study, two questionnaires have been referred for the preparation of traffic survey questionnaire. Firstly, short 15-item big five inventory (BFI-S) questionnaire is used to assess five personality traits: openness, extraversion, conscientiousness, neuroticism and agreeableness [14]. Secondly, driving behaviour survey is used to assess anxiety-based performance deficits, exaggerated safety/caution behaviour and hostile/aggressive behaviours of participants that can be faced while driving [15]. Thirdly, traffic questionnaire includes demographic information like education qualification, age, driving experience, gender, marital status and number of violations in last 6 months. Thus, the traffic survey is used to record demographical, personality and driving behaviour of participants.

3.2 Data Collection

Data are collected through Google form. Google form contains questionnaire which contains questions regarding demographic information, personality trait and driving behaviour. Google form is circulated through Whatsapp. Demographic information contains age, marital status, education, gender, driving experience and number of violations in last 3 months. Personality information identifies traits that make up whole personality of individual. Traffic behaviour information tells about the behaviour driver possess while driving. These behaviour can be hostile behaviour, cautious behaviour and causing error due to anxiety.

3.2.1 Dataset

Data are being collected through Google form. The survey has been conducted on random group. Total 239 records have been collected. Out of them, 136 participants have violated traffic rules, and 103 participants have not violated traffic rules. In this study, 140 male and 95 female participants have been participated, and remaining 4 records are faulty. It also contains 68 participants whose educational qualification is diploma or less than diploma and 171 participants whose education qualification

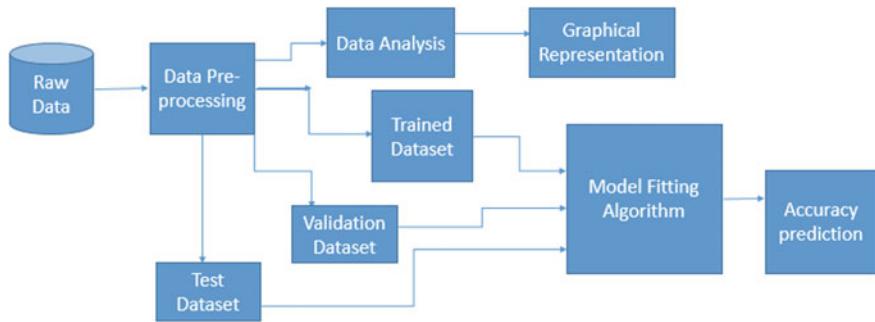


Fig. 2 Architecture of traffic violation prediction system

is graduation or higher than that. In this study, 165 participants are unmarried, and 70 are married. Remaining 4 records are faulty records. Out of 239 records, 119 records have 2–10 years driving experience, 65 records have 1 or less years driving experience, 36 records have 11–20 years driving experience, 31 records belong to 21–30 years driving experience, and remaining belong to 31 or more year driving experience.

3.3 *Architecture*

After collection of data, pre-processing of data is done. It includes few steps like finding missing values, finding outliers and conversion of categorical data in dummy variable. After pre-processing of data, data analysis is done. In this step, analysis of individual feature is done by finding its mean value and plotting graph for each variable. After data analysis, training of model is done with machine learning technique. Logistic regression, decision tree and ANN are used for training model. Testing and validation of model are done by using collective information. Before these steps, splitting of data is happened in ratio of 70:15:15. At last, comparison of these models is done on the basis of accuracy. Evaluation of model is done by calculating its accuracy and confusion matrix.

3.3.1 *Technology*

In this study, Psytoolkit has been referred for building questionnaire. For data pre-processing, model fitting and model evaluation, anaconda and Jupiter platform is used. Language used for this is Python.

3.3.2 Data Pre-processing

Data pre-processing is that step in which the data is changed, or *Encoded*, to bring it to such a form that the machine learning algorithm can easily implement it.

Steps of data pre-processing for this research are as follows:

- Row elimination.
- Checking for missing value.
- Checking for numerical data.
- Checking for categorical value.
- Checking for outlier.
- Analysis of data.
- Create dummy variable.
- Feature selection.
- Data splitting (70, 15, 15%).

Row Elimination

Row elimination is a process in which row is eliminated if it contains false information. In dataset, there are four rows which have faulty information. Hence, these rows have to be deleted to make the data suitable for data modelling.

Checking for Missing Value

Missing values mean that some values are not present for some attributes. In this research, median imputation is used to handle missing values. Any missing value in a given column is replaced by median of that column. There are three variables which have missing values.

Checking for Numerical Data

Data can be of two type that is as follows: (1) numerical data and (2) categorical data. Numerical data is further divided into two. One is discrete, and another is continuous. In this, there is only one discrete variable, and others are continuous.

Checking for Categorical Variables

In this, there are five categorical variable that is ‘Gender’, ‘Age’, ‘Education’, ‘Marital status’ and ‘Driving Experience’.

Checking for Outliers

Outlier is a point which is significantly different from other data points. Sometimes, outliers are error occurred in the dataset, or they are true outliers. If outlier is erroneous or not a part of population, then we can remove outliers. But, if they are part of population, then outliers should not be removed. Outlier is a point which is significantly different from other data points. Outliers for continuous variables are found through boxplot. In this research, dataset is small, and outliers are part of the population. Thus, they are not removed from dataset.

Analysis of Data

Data analysis is a process for identifying patterns in dataset. Analysis process can be done by doing statistical test or by visualisation like bar graph, histogram and scatter plot. In this research, graph and percentage are used to analyse individual features like finding maximum number of people possesses traits like openness, agreeableness, extraversion, conscientiousness and emotional stability or maximum number of people possesses driving behaviour like anxiety-based performance deficits, cautious behaviour and aggressive behaviour. The relationship between violations and different variables is found by percentage analysis as shown in Eqs. 1 and 2.

$$x_1 = \frac{y_1}{y_1 + z_1} \quad (1)$$

$$x_2 = \frac{z_1}{y_1 + z_1} \quad (2)$$

where x_1 represents % of participants having particular value for feature violates traffic rules, x_2 represents % of participants having particular value for feature do not violate traffic rules, y_1 represents number of participants having particular value for feature violates traffic rules, and z_1 represents number of participants having particular value for feature do not violate traffic rules.

Creation of Dummy Variable

Dummy variable is numeric variable which is used to represent categorical variable like gender, age, driving experience, marital status and education qualification.

Feature Selection

It is a process of reducing input variables to reduce cost or improve performance. In this research, recursive feature elimination (RFE) method is used for feature selection.

3.3.3 Modelling Process

Modelling is a process that can train model and model helps in predicting labels, so that it will satisfy business need and validate it on testing data [3]. In this research, decision tree, logistic regression and artificial neural network are used for model fitting. In ANN, it has 1 input layer containing 128 neurones, 1 output layer having 1 neurone and 3 hidden layers containing 256 neurons each. Relu function is used as activation function for all layers. Input layer has 128 artificial neurons, all hidden layers have 256 artificial neurons, and output layer has 1 neuron.

3.3.4 Model Validation

Model validation is the process of evaluating and measuring the performance of a trained model on test dataset. In the research, dataset is split into three parts that are 70, 15 and 15%. 70% dataset is used for training model, 15% is a test set, and 15% is used for validation set. Except this, accuracy, precision, F1 score, confusion matrix and ROC curve are used for measuring models performance.

4 Result and Discussion

In this study, most of the participants possess aggressive behaviour, caution behaviour, extraversion trait, emotional stability trait and agreeableness traits in range 2.5–3.5 like 46.81% participants possess aggressive behaviour, 39.57% possess caution behaviour, 65.96% possess extraversion trait, 58.30% participants possess, and 50.64% participants possess emotional stability in range 2.5–3.5. 45.53% participants possess anxiety-based performance deficits behaviour. That means, most of the participants do not make errors due to anxiety while driving.

Finding of this research also shows that 86.95% participants of age group 26–30 years violate traffic violations, and 80% of age group above 51 years do not violate traffic rules. Similarly, 61.61% having diploma or less education, 67.22% having 2–10 years of driving experience violate traffic rules. On the other hand, 69.23% participants having 21–30 years and 75% participants having 31 or more years driving experience do not violate rules.

Three models have been used for model fitting. Decision tree, logistic regression and artificial neural network give 59%, 60% and 73%, respectively, as shown in Figs. 3, 4 and 5. Therefore, artificial neural network is used for model fitting (Fig. 5).

Fig. 3 ROC graph of logistic regression

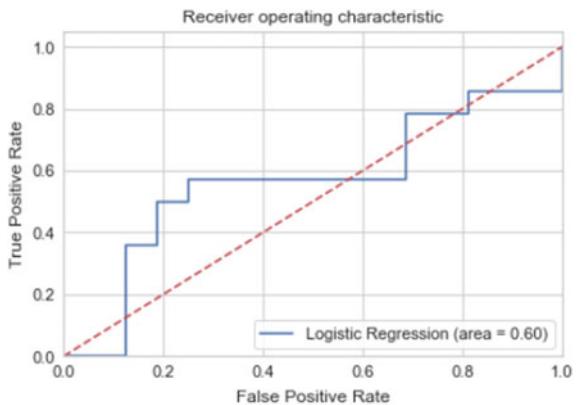


Fig. 4 ROC graph of decision tree

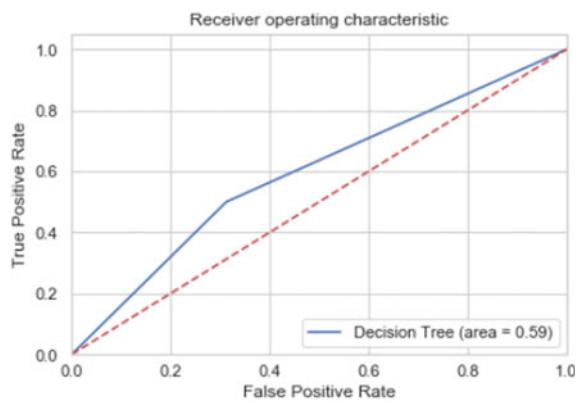
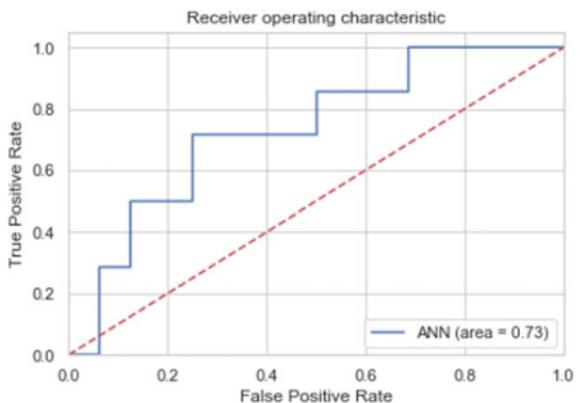


Fig. 5 ROC graph of ANN



5 Conclusion and Future Work

The aim study is to predict traffic violations on the basis of human behaviour. In this study, it is shown that participants show more aggressive and cautious behaviour while driving. However, there is less number of participants which causes error due to anxiety. Most of the participants are high or medium on all five personalities (emotional stability, extraversion, openness, agreeableness and conscientiousness trait). Less number of the participants is low on these traits. That means, most of the participants are more social, have empathy, open minded, creative, stable and balanced. In the study, participant with 2–10 years of driving experience or of age group 26–30 years causes more traffic violations than others. Artificial neural network is used for implementing model which is used to predict traffic violations on the selected features. Its performance is 73%. This model can be used in driving schools, by traffic police, etc.

The future work for this is to include more parameters (impulsiveness, mental illness, etc.) and gather large dataset. In current study, we just predict occurrence of traffic violations or not. But, in future work, prediction of specific traffic violations can be done and identification of more frequent traffic violator.

References

1. Alavi SS, Mohammadi MR, Souri H, Kalhorri SM, Jannatifard F, Sepahbodi G (2017) Personality, driving behavior and mental disorders factors as predictors of road traffic accidents based on logistic regression. *Iran J Med Sci* 42(1):24
2. Balasubramanian V, Sivasankaran SK (2019) Analysis of factors associated with exceeding lawful speed traffic violations in Indian metropolitan city. *J Transp Saf Sec* 1–17
3. Koehrsen W (2018) Modeling: teaching a machine learning algorithm to deliver business value. [Online]. Available <https://towardsdatascience.com/modeling-teaching-a-machine-learning-algorithm-to-deliver-business-value-ad0205ca4c86>. Accessed 22 June 2020
4. Yang J, Du F, Qu W, Gong Z, Sun X (2013) Effects of personality on risky driving behavior and accident involvement for Chinese drivers. *Traffic Inj Prev* 14(6):565–571
5. Sani SRH, Tabibi Z, Fadardi JS, Stavrinou D (2017) Aggression, emotional self-regulation, attentional bias, and cognitive inhibition predict risky driving behavior. *Accid Anal Prev* 109:78–88
6. Yoh K, Okamoto T, Inoi H, Doi K (2017) Comparative study on foreign drivers' characteristics using traffic violation and accident statistics in Japan. *IATSS Res* 41(2):94–105
7. Davis S (2019) Road rage: what it is, how to avoid it. [Online]. Available <https://www.webmd.com/mental-health/features/road-rage-what-it-is-how-to-avoid-it#1>. Accessed 15 June 2020
8. Porter BE (ed) (2011) Handbook of traffic psychology. Academic Press
9. Zhang M, Han N, Lobo BJ (2019) Understanding and predicting drivers' seatbelt usage in crashes in Virginia. In: 2019 systems and information engineering design symposium (SIEDS). IEEE, pp 1–6
10. Găianu PA, Giosan C, Sârbescu P (2020) From trait anger to aggressive violations in road traffic. *Transp Res Part F Traffic Psychol Behav* 70:15–24
11. Hezaveh AM, Cherry CR (2019) Neighborhood-level factors affecting seat belt use. *Accid Anal Prev* 122:153–161

12. Shahar A (2009) Self-reported driving behaviors as a function of trait anxiety. *Accid Anal Prev* 41(2):241–245
13. Statista (2020) Road accidents in India—statistics and facts. [Online]. Available <https://www.statista.com/topics/5982/road-accidents-in-india/>. Accessed 20 June 2020
14. Lang FR, John D, Ludtke O, Schupp J, Wagner GG (2011) Short assessment of the Big Five: robust accros survey methods except telephone interviewing. *Behav Res Methods* 43:548–567
15. Clapp JD, Olsen SA, Beck JG, Palyo SA, Grant DM, Gudmundsdottir B, Marques L (2011) The driving behavior survey: scale construction and validation. *J Anxiety Disord* 25(1):96–105

Chapter 4

Recent Developments and Challenges in Intelligent Transportation Systems (ITS)—A Survey



Vishal Sharma, Love Kumar, and Sergey Sergeyev

1 Introduction

The traffic volume of today's era in many nations is escalating at a rapid rate and imposing severe issues like the long delays, huge fuel consumption, high CO₂ emission, causalities, huge traffic jams, and moreover, the lower quality life. According to Texas Transportation Institute (TTI), the commuters of the US spend \approx 8.8 billion hours per year in traffic jams with 3.3 billion gallons of fuel consumption which costs an overall nationwide expenditure of \approx \$166 billion [1]. Meanwhile, about \approx 1.24 million people around the world lost their life due to road accidents, and \approx 20–50 million people survive with critical damages. If this current tendency persists, it is predicted that road accidents may increase by \approx 65% and turn out to be the fifth major reason for fatality by 2030 [2]. The direct estimated costs due to road-accident injuries have been \approx 1, \approx 1.5, and \approx 2% of the total revenue of the underdeveloped, developing, and well-developed countries, respectively, [3]. Moreover, as reported by United Nation Population Fund [4] and Population Reference Bureau [5], the population is growing drastically, and several people are migrating to the urban vicinities, and therefore, the transportation problems will aggravate in near future. To combat the critical setbacks of the growing traffic, intelligent transportation systems are a superlative alternative to traditional transportation. To reduce road accidents and to increase the safety of the drivers, some considerable attention has been remunerated to the automobile protection systems. In this direction, the automotive safety system is being implemented by proposing active safety and passive safety systems [6]. The passive safety system including the seatbelts, airbags, and crumple zones has

V. Sharma (✉) · S. Sergeyev

Aston Institute of Photonics and Technologies (AiPT), Aston University, Birmingham, UK
e-mail: v.vishal@aston.ac.uk

L. Kumar

Department of Electronics and Communication, DAVIET, Jalandhar, Punjab, India

been widely employed for many years and almost reached its full potential. Alternatively, the active safety system which is the main ingredient of ITS environment comprises of a driver assistance system (DAS) which further includes several subsystems, like the automotive collision avoidance system (ACAS), lane departure warning system (LDWS), and brake assistance system (BAS) to update the drivers about the road conditions and to take appropriate prompt action accordingly. Besides these autonomous systems, the sensing platform becomes a crucial part of the ITS system, broadly classified into two categories: intra-vehicular and urban sensing platforms [7]. The intra-vehicular sensing platform is accountable for the vehicle condition; meanwhile, the urban sensing platforms devise the traffic conditions. Recently, light detection and ranging (LiDAR)-based object detection and range estimation technology are on the way to revalorizing the ITS industry and is playing a vital role in the development of autonomous vehicles. This technology uses beams of laser light to detect and profiling the object from a distance. Therefore, colossal research activities are being carried out in the implementation of LiDAR-based ITS environments.

Keeping in view of the recent developments in the ITS industry, firstly, the authors cover the integration of several emerging sensor-based technologies along with their optimal deployment to realize a sustainable, responsive, and secure ITS environment. Secondly, the challenges that need to be addressed in the future to develop a fully operational and co-operative ITS atmosphere along with an illustration of the future research directions for realizing the next-generation ITS system.

2 Developments of Intelligent Transport Systems

Initially, the major developments in the ITS industry are accomplished in Europe, US, and Japan through three phases, i.e. preparation (1930–1980), feasibility study (1980–1995), and the product development (1995–present) [8, 9]. During the preparation phase, the development of the microprocessor and GPS system positioned the initial groundwork of today's ITS environments. During the feasibility phase, the joint efforts of various industries and academicians developed a Program for European Traffic with Efficiency and Unprecedented Safety (PROMETHEUS) [9] that installed multiple forward looking TV cameras and processor to realize the automatic lane and road track. US Department of Transportation (USDOT) formed an Intelligent Transportation Society of America (ITS America) in 1994 that worked tremendously and developed an automated highway system (AHS) [10]. The preparation-and feasibility phase have created the technical foundation to develop the high-end ITS environment, which is being used in the product development phase and at the initial stage (Table 1).

During the past decade, several technologies have been developed for vehicle navigation and positioning. Some computer-based systems are reported; however, they experience some issues including a narrow field of view, intensity variations, and low accuracy in depth information. At the same time, the sensor technology comes out as an attractive approach. It becomes mandatory to equip the self-driving vehicles

Table 1 Various development phases of ITS and technologies introduced

ITS phase	Year	Technology
Preparation	1930–1980	Development of the microprocessor and GPS system
Feasibility study	1980–1995	European traffic with efficiency and unprecedented safety program introduced, installation of multiple forward looking TV cameras, and processor, realization of the automatic lane and road track, automated highway system (AHS) developed
Product development	1995–present	Sensor technology introduced, LiDAR equipped automotive vehicles introduced, implementation of rear-view visibility, tire pressure measurements like functions, self-control, traffic congestion, and execution of intelligent transportation system

with sensors to offer a variety of applications, for instance, rear-view visibility, tire pressure measurements like functions, self-control, traffic congestion, and execution of intelligent transportation system in a more realistic way. Today's autonomous vehicle (AV) industry is offering smart vehicles equipped with 60–100 sensors and maybe augmented to 200 in the near future as per the prediction of the ITS industry [11]. Conversely, LiDAR is also gaining popularity in AV industries in the last few years to provide a wide field of view and high accuracy with efficient data processing capability. As per the current market scenario, LiDAR will touch $\approx \$4.5$ billion by 2022 at a CAGR of $\approx 24.0\%$. Therefore, many LiDAR sensors are being developed ranging from planer to multi-planer and two dimensions to three dimensions. The most commonly used LiDAR sensor is the SICK LMS 2xx series [12] operating up to a distance of 80 m, having a range resolution of 5 cm with azimuth accuracy of 0.5° . Another, well-liked single-planar LiDAR sensor in ITS systems is HOKUYO UXM-30LN with a detection range of 60 m, a field of view of 190° with a range resolution from 30 to 50 mm [13]. On the other hand, a widely used 2D multi-planar LiDAR sensor is ALASCA-XT having the ability to split a laser beam into four vertical layers [14] and to detect an object up to 200 m with a field of view of 240° . Although, Velodyne HDL-64E is a 3D LiDAR sensor, specifically designed for driverless vehicles with a detection range of 100 m and resolution of < 5 cm. Velodyne provides 3D information of neighbouring environment with 360° horizontal field of view and 0.09° angular resolutions [15]. All measurements made by LiDAR sensors are based on a set of points where each point is a collection of polar coordinates and are known as LiDAR cloud points. LiDAR cloud point is gaining attention due to its features of object detection, object identification, and remote navigation applications. However, LiDAR performance depends on the climate conditions as well and is vulnerable to the atmospheric fluctuations [16].

3 Challenges of Intelligent Transportation Systems

The ITS system is an ad hoc network-based technology, and its security becomes a pre-requisite for its deployment. Moreover, as the speed and density of the vehicle are dynamic in nature, therefore, a fixed network topology cannot be adopted. The situation becomes more critical when the numbers of vehicles are closely spaced. Therefore, more smart and efficient secure approaches are needed to tackle the dense traffic scenarios [17]. Furthermore, a wider bandwidth is needed in a dense environment to reduce the communication inference, delay, and delivery ratio [17, 18]. Conversely, some of these challenges have been resolved by introducing LiDAR technology and is being deployed in AVs efficiently. The malicious attacker is another challenge as they may inject false information to mislead the vehicles [19]. To tackles these security and privacy issues, many research attempts have been reported. However, many of these issues still persist for real-time ITS deployment as the existing work is either application specific or depends upon the nature of the attacker [20–23]. Additionally, the other emerging information and communication technologies (ICT), such as the Internet of Things (IoT), cloud computing, 5G network, artificial intelligence (AI), big data, and Internet of vehicles (IoV) have remarkably revolutionized the ITS [24]. Recently, the software-defined networking (SDN) technology is being adopted to monitor the network status globally to estimate and predict the traffic congestion along with the spatiotemporal features [24]. Besides these technologies, the exact prediction of traffic congestion in every part of the transportation network is still facing major challenges. For example, the global abstract view of the entire transportation network is one of the critical defy. Although, the current ITS systems provide us the instant path determination and navigation; however, it needs to be revised and re-recommended to achieve an optimal route discovery solution [24–30]. Besides all of these revolutions in ICT-based technologies, their interoperability is still a big confront as the client agencies are not authorized to share the data and to develop a common data format. Even if the data system interoperability and standardization are achieved, the addition of the raw data at large scale to obtain the required results will also be cost effective [31, 32]. Therefore, the vehicle manufacturers, government policy-makers, global standardization deeds, and the business models are required to play essential roles collectively for the successful deployment of ITS systems. The flowchart for the ITS in based on road condition is illustrated in Fig. 1.

4 Futuristic Possibilities for Intelligent Transportation Systems

The development, challenges, and approaches in ITS industries depict that its future falls in multi-layers and cross-layers, i.e. the cyber, social, and physical environments. In near future, the task of ITS will not be limited to gather data of instant

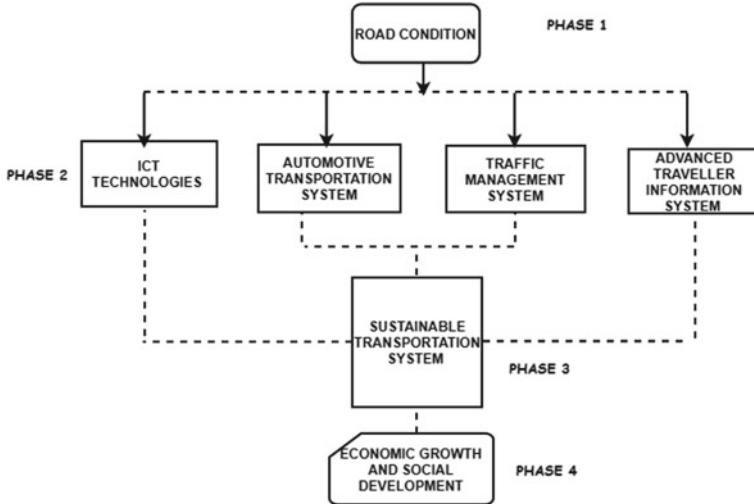


Fig. 1 Flowchart for intelligent transportation system [31]

road conditions, but extend to the prediction of future traffic condition, and to allow agencies to change the plan and strategies accordingly as per a press release of IBM [33]. Apart from physical traffic congestion data, generally congregated by various sensors mounted on autonomous vehicles, the collection of public attributes and perception from cyberspace is an upcoming opportunity in the ITS domain. These public attribute can be congregated from social networking sites like Twitter, natural language processing (NLP) algorithms [34–36], and predefined semantic structures. Further, the fusion of collected data by CPS is an emerging domain for futuristic intelligent transportations. In future, a hierarchical traffic network model that will integrate the physical, semantic, logical, and perceptual networks in the digital reconstruction of CSP spaces may be developed [30]. Additionally, the cross-domain data fusion strategies to identify and define the type and amount of information data can be implemented by incorporating the statistical and NLP approaches to realize state-of-the-art ITS environment efficiently. Moreover, the behaviour and characteristics of AVs are different from regular vehicles; therefore, the flow models underconnected environment are another future direction for smart and intelligent transportation. Additionally, an open-source platform for the ITS environment is an immediate need of the future for smoothly integrated pathways, intending to provide hassle-free upgradation of the earlier developed ITS techniques. The potential strategies, practical issues, and future directions for the ITS systems are depicted in Fig. 2.

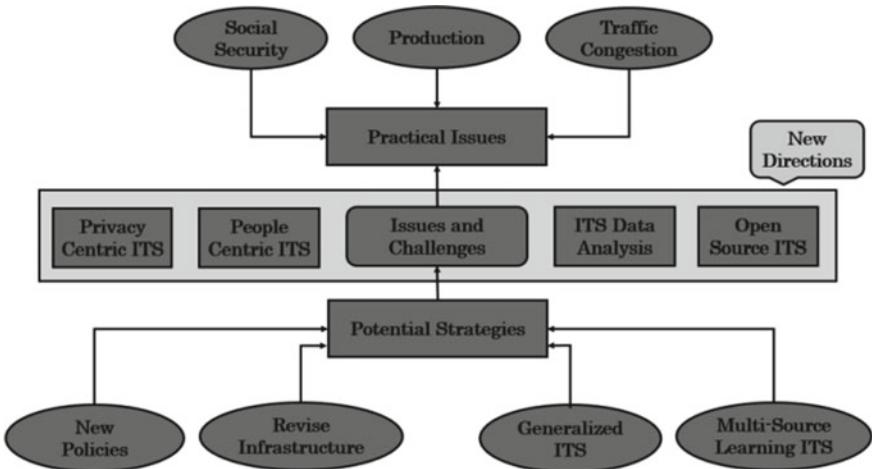


Fig. 2 Potential strategies, practical issues, and future direction for ITS

5 Conclusion

In this manuscript, the recent developments in the area of ITSs are highlighted along with the challenging aspects of the conventional transportation systems to combat the growing population and traffic scenarios. Moreover, the upcoming and under consideration strategies, policies, and developments to realize advanced, secure, and economical intelligent transportation networks are also outlined.

Acknowledgements This work is carried out in Aston Institute of Photonic Technologies, School of Engineering and Applied Science, Aston University, Birmingham, UK, and is supported by European Union-sponsored H2020-MSCA-IF-EF-ST project no: 840267.

References

1. Texas A&M Transportation Institute. Urban Mobility Scorecard, INRIX. Technical Report 2019. <https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf>. Last accessed 2020/06/12
2. Mukhtar A et al (2015) Vehicle detection techniques for collision avoidance systems: a review. *IEEE Trans Intell Transp Syst* 16(5):2318–2338
3. Peden M (2004) World report on road traffic injury prevention: summary. World Health Organization (WHO), Geneva, Switzerland. Last accessed 2020/06/12
4. United Nations Population Fund (UNFPA) (2011) State of world population 2011: people and possibilities in a world of 7 Billion. Technical Report, USA. Last accessed 2020/06/12
5. Population Reference Bureau (2016) 2016 world population datasheet, inform empower advance. Available online <http://www.prb.org/pdf16/prb-wpds2016-web-2016.pdf>. Last accessed 2020/06/12

6. Zhou H, Cao P, Chen S (2016) A novel waveform design for multi-target detection in automotive FMCW radar. In: IEEE radar conference (RadarConf). <https://doi.org/10.1109/radar.2016.7485315>
7. Guerrero-Ibáñez J, Zeadally S, Contreras-Castillo J (2018) Sensor technologies for intelligent transportation systems. Sensors 18(4):1212. <https://doi.org/10.3390/s18041212>
8. Masaki I (1998) Machine-vision systems for intelligent transportation systems. IEEE Intell Syst 13(6):24–31. <https://doi.org/10.1109/5254.735999>
9. Figueiredo L, Jesus I, Machado JAT, Ferreira JR, Martins de Carvalho JL (n.d.) Towards the development of intelligent transportation systems. In: 2001 IEEE intelligent transportation systems (ITSC 2001). Proceedings (Cat. No.01TH8585). <https://doi.org/10.1109/itsc.2001.948835>
10. Koshi M (1989) Development of the advanced vehicle-road information system in Japan—the' CACS project and after. In: Proceedings of JSK international symposium—technological innovations for tomorrow's automobile traffic and driving information systems, pp 9–19
11. Yilmaz Y, Uludag S, Dilek E, Ayozen YE (2016) A preliminary work on predicting travel times and optimal routes using Istanbul's real traffic data. In: 9th transit transport congress and exhibition
12. SICK U.S.A. see <http://www.sick.com/us/en-us/home/Pages/Homepage1.aspx>. Last accessed 2020/06/12
13. Online http://www.hokuyoaut.jp/02sensor/07scanner/uxm_30ln.html. Last accessed 2020/06/12
14. The laser scanner product overview. see <http://www.ibeoas.com/english/products.asp>
15. Velodyne HDL-64E LIDAR. <http://www.hizook.com/blog/2009/01/04/velodyne-hdl-64e-laser-rangefinder-lidar-pseudo-disassembled>
16. Sharma V, Sergeyev S (2020) Range detection assessment of photonic radar under adverse weather perceptions. Opt Commun 472:
17. Zhang J (2011) A survey on trust management for vanets. In: Proceedings of the 2011 IEEE international conference on advanced information networking and applications (AINA), Singapore, 22–25 March 2011, pp 105–112
18. Shen X, Cheng X, Yang L, Zhang R, Jiao B (2014) Data dissemination in Vanets: a scheduling approach. IEEE Trans Intell Transp Syst 15:2213–2223
19. Bouassida MS (2011) Authentication versus privacy within vehicular ad hoc networks. Int J Netw Secur 13:121–134
20. Lin J et al (2017) A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J 4:1125–1142
21. Andrea I et al (2015) Internet of Things: security vulnerabilities and challenges. In: IEEE symposium on computers and communication (ISCC), pp 180–187
22. Niu J, Jin Y, Lee AJ, Sandhu R, Xu W, Zhang X (2016) Panel security and privacy in the age of Internet of Things: opportunities and challenges. In: Proceedings 21st ACM on symposium on access control models and technologies, Shanghai, China, 6–8 June 2016, pp 49–50
23. Qu F, Wu Z, Wang F, Cho W (2015) A security and privacy review of VANETs. IEEE Trans Intell Transp Syst 16(6):2985–2996. <https://doi.org/10.1109/its.2015.2439292>
24. Engoulou RG, Bellaïche M, Pierre S, Quintero A (2014) VANET security surveys. Comput Commun 44:1–13
25. Petit J, Schaub F, Feiri M, Kargl F (2015) Pseudonym schemes in vehicular networks: a survey. IEEE Commun Surv Tutor 17:228–255
26. Boualouache A, Senouci S-M, Moussaoui S (2017) A survey on pseudonym changing strategies for vehicular ad-hoc networks. IEEE Commun Surv Tutor 2017(20):770–790
27. Lin C, Han G, Du J, Xu T, Shu L, Lv Z (2020) Spatio-temporal congestion-aware path planning towards intelligent transportation systems in software-defined smart city. IEEE Internet Things J Early Access
28. Goto Y, Masuyama H, Ng B, Seah WKG, Takahashi Y (2016) Queueing analysis of software defined network with realistic OpenFlow-based switch model. In: 2016 IEEE 24th international symposium on modeling, analysis and simulation of computer and telecommunication systems (MASCOTS). <https://doi.org/10.1109/mascots.2016.30>

29. Zou D, Li S, Kong X, Ouyang H, Li Z (2018) Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling. *Energy* 147(8):59–80
30. Sumalee A, Ho HW (2018) Smarter and more connected: future intelligent transportation system. *IATSS Res* 42(2):67–71
31. Qureshi KN, Abdullah AH (2013) A survey on intelligent transportation systems. *Middle-East J Sci Res* 15(5):629–642
32. Wang W, Krishnan R, Diehl A Advances and challenges in intelligent transportation: the evolution of ICT to address transport challenges in developing countries. <https://www.worldbank.org/en/topic/transport/brief/connections-note-26>. Last accessed 2020/06/14
33. IBM and Texas Transportation Institute to Collaborate on Intelligent Transportation Projects. Available online <https://www-03.ibm.com/press/us/en/pressrelease/30809.wss>
34. Hasselmann JT Machine intelligence in the travel and transportation industry. <https://towardsdatascience.com/machine-intelligence-in-the-travel-transportation-industry-e63606cd45f1>. Last accessed 2020/06/14
35. Hürriyetoğlu A, Oostdijk N, van den Bosch A (2017) Estimating time to event of future events based on linguistic cues on Twitter. *Stud Comput Intell* 67–97. https://doi.org/10.1007/978-3-319-67056-0_5
36. Dabiri S (2019) Application of deep learning in intelligent transportation systems. Virginia Polytechnic Institute and State University

Chapter 5

A Modified YOLO Model for On-Road Vehicle Detection in Varying Weather Conditions



Rajib Ghosh

1 Introduction

Investigations on developing computer vision-based automated systems for detection and tracking of on-road vehicles have gained momentum in the recent past due to the vast applications of these systems in this digital era. The main goal of these systems is to detect various on-road vehicles of varying size from still images or video data collected from the road. These systems provide innovative support to transport and traffic management in various ways. Vehicle detection systems provide information in vehicle counting and classification at toll plaza [1], vehicle number plate recognition [2], vehicle speed measurement, tracking traffic accidents, etc. These systems also provide assistance to the drivers during driving by tracking front and rear vehicles in the same as well as in different lanes, especially in heavy rain, snowfall or foggy weather. With the continuous development of urban roads, increase in vehicle buying capacity of common people and as a consequence more number of on-road vehicles, the importance of automated vehicle detection systems is gradually increasing. With the advent of more developed computer vision techniques, more developed systems are being enabled in this domain.

Several computer vision-based vehicle tracking systems using different machine learning (ML) methods are available in the literature. Researchers have mostly relied upon three different shallow ML techniques to detect vehicles—support vector machine (SVM) [3–6], AdaBoost classifier [5, 7, 8] and artificial neural network (ANN) [9, 10]. Although shallow convolutional neural network (CNN) has been successfully used in various studies for object detection in natural scene images, a little number of studies [12] have used shallow CNN to locate on-road vehicles in natural scene images. Even the use of faster R-CNN, a deep learning network has also been reported in the literature [13] for on-road vehicle detection. But, no study has

R. Ghosh (✉)

National Institute of Technology Patna, Patna 800005, India

e-mail: rajib.ghosh@nitp.ac.in

been found in the literature on using you only look once (YOLO)-based deep learning model to detect on-road vehicles from the images. YOLO, a recently developed deep learning model, was introduced by Redmon et al. [14] in 2016 to detect various objects in natural scene images. Faster R-CNN network generates detecting bounding boxes through region proposal network (RPN) and then trains a classifier to learn these bounding boxes. Detecting bounding boxes are then generated on test images based on this training model. The classification phase is followed by post-processing to eliminate incorrect bounding boxes. This entire process is time consuming and makes the system slow. The present article proposes a new model for on-road vehicle detection in varying weather conditions by modifying the existing model of YOLO. In YOLO model, a single CNN predicts the detecting bounding boxes and the class probabilities for those boxes together in one evaluation from the full images. In this model, detection of vehicles is represented as a regression problem and due to which it does not need to implement a complex process like faster R-CNN. This makes the present system extremely faster in comparison with faster R-CNN-based systems. The proposed system runs at 43 frames per second (fps) on the Alienware aurora GPU. The vehicles have been detected from the images collected in good (sunny) as well as bad weather conditions like blizzard, snowfall and wet snow weather. Few sample on-road vehicle images in varying weather conditions from CDNet 2014 [15] and LISA 2010 [16] public data sets are shown in Fig. 1.

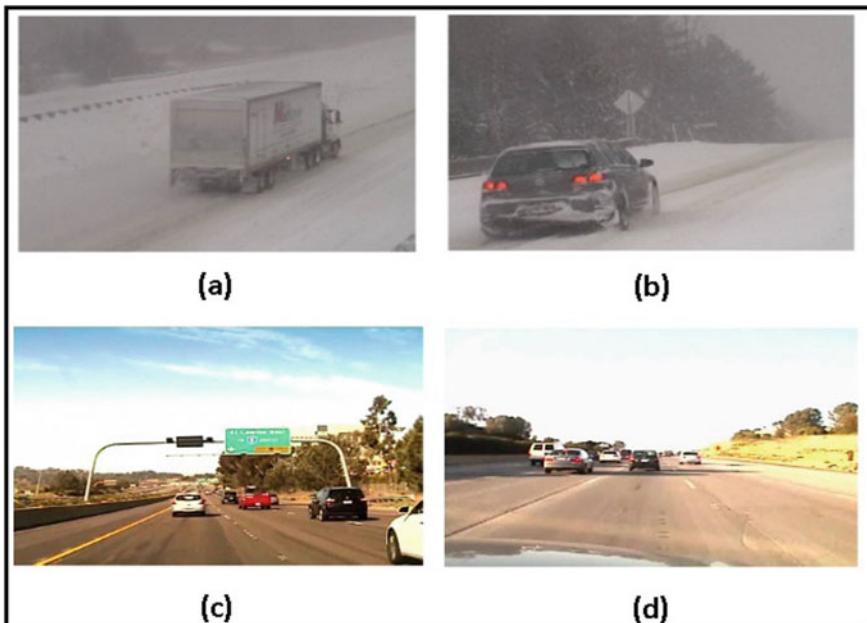


Fig. 1 Few sample on-road vehicle images in varying weather conditions: **a** In blizzard (CDNet 2014 dataset), **b** In snowfall (CDNet 2014 dataset), **c** In sunny (LISA 2010 dataset) and **d** In dense (LISA 2010 dataset)

The rest of the paper is organized as follows. Related works are discussed in Sect. 2. Section 3 details the proposed methodology. Section 4 analyses the vehicle detection results of the present system. Finally, Sect. 5 concludes the paper with a direction for future research.

2 Literature Survey

Several studies on computer vision-based vehicle detection are available in the literature. Most of the studies have relied on different shallow ML techniques to detect vehicles, whereas few studies have used non-ML techniques as well. Some of those related studies are discussed below in brief.

Sun et al. [3] presented one computer vision-based vehicle detection system utilizing the popular ML technique, SVM. In this study, two different classes were created, one for vehicle and the other for non-vehicle, and the classification was performed using SVM after extracting various features from the samples of each class. Cheon et al. [4] proposed another vision-based vehicle detection method where the features were extracted through histogram of oriented gradients (HOG), and the classification of vehicle and non-vehicle was performed using SVM. In another study [5] on on-road vehicle detection, two different classifiers were applied, SVM and AdaBoost, to detect the vehicles. Feature vectors generated through HOG features were classified using SVM, and the vectors generated through Haar-like features were classified using AdaBoost classifier. Khairdoost et al. [6] presented an on-road vehicle detection method for both front and rear vehicles. Feature vectors were generated through pyramid histogram of oriented gradients (PHOG) features and classified using linear SVM. Yan et al. [7] reported one investigation outcome in this regard using AdaBoost classifier. Two types of HOG features were extracted from vehicle and non-vehicle samples, and the feature vectors were classified using AdaBoost classifier in one of the two classes, vehicle or non-vehicle. The use of Adaboost classifier is found in another study [8] as well, where Haar-like features were extracted from vehicle and non-vehicle classes samples, and the classification of these feature vectors was performed using AdaBoost classifier. Few studies have relied upon ANN as well to detect on-road vehicles. Ming et al. [9] used ANN to detect vehicles based on the information obtained from vehicle tail light. In another study on the use ANN [10], Haar-like features were extracted from the samples of vehicle and non-vehicle classes, and the feature vectors were classified using ANN. Matthews et al. [11] presented a two-stage vehicle recognition method by combining image processing techniques with ANN. ROIs were generated using various image processing techniques which were then fed to ANN for vehicle recognition. Zhou et al. [12] proposed an end-to-end vehicle detection system using shallow CNN. Fan et al. [13] presented a deep learning-based on-road vehicle detection system using faster R-CNN. Sivaraman et al. [16] presented an active-learning framework to track on-road vehicles. The system was trained using a supervised learning technique. Ghosh et al. [1] presented a system for detection and classification of vehicles in toll

plaza using image processing and ML techniques. Chan et al. [17] presented a non-ML-based system to detect the preceding vehicles during driving in varying light and weather conditions. Four different vehicular structure-related cues were considered, and a particle filter, combined with these cues, was used to detect the preceding vehicles. Another non-ML-based vehicle tracking system was proposed by Chellappa et al. [18] using acoustic and video sensors. Both video and acoustic sensors were fused to develop the system. Bertozzi et al. [19] proposed another non-ML-based vehicle detection system where vehicles were located through detecting its corners as in general vehicles have a rectangular shape with four corners. One template was considered corresponding to each corner to locate all of the four corners of any vehicle. In another non-ML-based vehicle detection study [20], a multi-sensor correlation method was proposed utilizing magnetic wireless sensor network. Haselhoff et al. [21] used Haar-like and triangle features to detect on-road vehicles. Triangle filters were computed for feature extraction using four integral images.

So, the survey shows that no study has been found on on-road vehicle detection using YOLO-based model.

3 Proposed Methodology

The present investigation proposes an end-to-end method of on-road vehicle detection and tracking by proposing a modified structure of the YOLO model. As YOLO model was proposed in 2016 to detect various objects from scene images, so it consisted of 24 convolutional layers followed by two fully connected layers [14]. The present investigation has modified this structure of YOLO by including 16 convolutional layers followed by two fully connected layers as it focuses on detection of vehicles only from scene images. The proposed modified architecture of YOLO is shown in Fig. 2.

3.1 Detection Method

In the proposed method, the entire input image is divided into several grids. If any one of the grids contains the centre of the vehicle, then that grid plays the role of

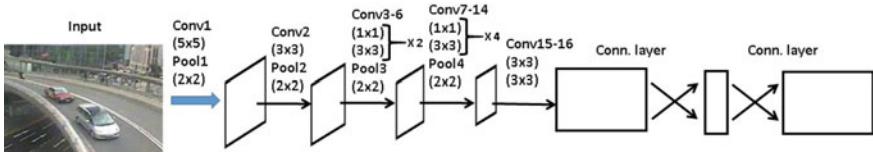


Fig. 2 Proposed modified architecture of YOLO

detection of the vehicle. Each grid predicts few bounding boxes and the confidence score of each bounding box. If there is no object in any grid, then the confidence score is zero for each box. Otherwise, confidence score should be equal to the intersection over union (IoU) value between the bounding box and ground-truth. IoU measures the area of overlapping between the predicted box and the ground-truth. If the IoU value is greater than 0.8 for any predicted box, then it has been considered as the correct predicted box.

3.2 Network Architecture

The proposed model has been implemented using a convolutional neural network. The various convolutional layers extract the features from the input image, whereas the fully connected layers predict the detecting bounding boxes and the class probabilities for those boxes. In this investigation, two classes are considered—vehicle and non-vehicle. The proposed network architecture has 16 convolutional layers followed by two fully connected layers. Instead of 24 layers of YOLO, 16 layers have been used in this work as it focuses only on vehicle detection from images; whereas, YOLO model focuses on detection of any object in scene images. A 5×5 dimensional filter has been used in convolutional layer 1; whereas, convolutional layer 2 uses a 3×3 dimensional filter. Maxpooling operation has been performed after each of these two convolutional layers. Two varying dimensional filters of size 1×1 and 3×3 have been used in third and fourth convolutional layers, respectively. This combination of filters has been repeated in fifth and sixth convolutional layers, respectively, also. A maxpooling operation is performed after the sixth convolutional layer. 1×1 and 3×3 dimensional filters have been repeated from seventh to 14th convolutional layers also. A final maxpooling operation is followed by the 14th convolutional layer. Finally, a 3×3 dimensional filter has been used in each of the 15th and 16th convolutional layers.

3.3 Training Scheme

The convolutional layers of the proposed model have been pretrained on the CDNet 2014 public data set. The pretraining has been performed using the first 14 convolutional and associated pooling layers shown in Fig. 2 followed by a fully connected layer. The model has then been converted to perform the vehicle detection. To improve the detection performance, both convolutional and fully connected layers have been added to the pretrained network [14]. In this work, two convolutional and two fully connected layers have been added to the pretrained network.

4 Experimental Results and Analysis

The performance of the proposed system and the data sets used to evaluate this performance is discussed in this section.

4.1 Data set Description

In comparison with other research fields in computer vision, on-road vehicle detection field has fewer publicly available data sets. The present investigation has used CDNet 2014 public data set to train the proposed model. This data set contains a realistic, camera-captured, diverse set of videos of various on-road vehicles in good (sunny) as well as in challenging weather conditions like blizzard, snowfall and wet snow under the category “bad weather” and due to which this data set has been chosen in the present investigation to train the proposed model as it was planned to evaluate the performance of the proposed system in varying weather conditions. Videos were acquired in both day and nighttime conditions in CDNet 2014 data set. The testing phase has been performed using both CDNet 2014 and LISA 2010 public datasets. After keeping aside most of the samples of CDNet 2014 data set for training purpose, remaining samples from this data set have been used for the testing purpose. The LISA 2010 data set contains three video sequences, in dense condition, sunny weather and on urban roads, captured from an on-board camera. All the samples from this data set have been used to test the system performance.

4.2 Vehicle Detection Results

The performance of detecting on-road vehicles of the present system has been measured through various metrics, namely accuracy, precision, recall and F1-score. The values of these metrics have been computed by matching the detected bounding boxes with the ground-truth bounding boxes. The performance has been evaluated in varying weather conditions using CDNet 2014 data set, whereas only in sunny weather condition using LISA 2010 data set as it does contain video sequences only in sunny weather. The vehicle detection results of the present system in terms of accuracy and precision are presented in Table 1. Recall and F1-score of the proposed system are presented in Table 2. Table 3 presents the processing speed of the proposed system in fps for both of the testing data sets. Figures 3 and 4 illustrate the correct detection of on-road vehicles on few test images from both the testing data sets. Detected vehicles have been enclosed within green-coloured boxes in this figure.

Table 1 Accuracy and precision of the proposed vehicle detection system

Dataset	Weather condition	Accuracy (%)	Precision (%)
CDNet 2014	Sunny	92.84	92.80
CDNet 2014	Blizzard	91.02	90.96
CDNet 2014	Snowfall	89.76	89.72
CDNet 2014	Wet snow	89.72	89.68
LISA 2010	Sunny	93.17	93.08

Table 2 Recall and F1-score of the proposed vehicle detection system

Dataset	Weather condition	Recall (%)	F1-Score (%)
CDNet 2014	Sunny	92.61	92.71
CDNet 2014	Blizzard	90.82	90.87
CDNet 2014	Snowfall	89.54	89.61
CDNet 2014	Wet snow	89.51	89.58
LISA 2010	Sunny	92.97	93.02

Table 3 Processing speed of the proposed vehicle detection system

Dataset	Processing speed (in fps)
CDNet 2014	43
LISA 2010	43

4.3 Error Analysis

For few test images, false positive results have been obtained. The present system could not detect the vehicles in these images properly; the detecting boxes have enclosed some other objects in these images, but not the vehicles. Few erroneous detections have been obtained in bad weather conditions like wet snow or blizzard environments, and few have occurred in normal weather condition also where some other objects have obstructed the vision of vehicle in the image. Few erroneous outcomes are shown in Fig. 5.

4.4 Comparative Performance Analysis

Most of the research explorations in on-road vehicle detection field have not used publicly available data sets to evaluate the system performance, rather they relied upon self-generated non-public data sets due to the specific designed algorithms. So, the performance of the proposed system cannot be compared with the existing

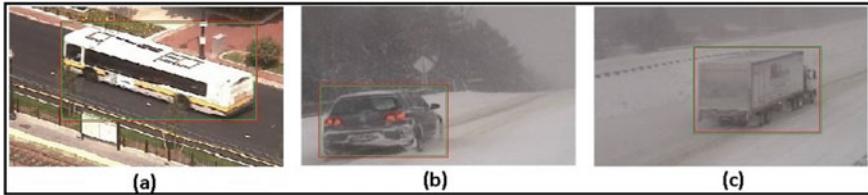


Fig. 3 Correct detection of on-road vehicles on few test images from CDNet 2014 data set in **a** Sunny, **b** Snow fall and **c** Blizzard conditions. The red bounding box indicates the ground-truth, and the green one indicates the detecting box



Fig. 4 Correct detection of on-road vehicles on few test images in **a** Sunny (LISA 2010 dataset) and **b** Wet snow (CDNet 2014 dataset) conditions. The red bounding box indicates the ground-truth, and the green one indicates the detecting box



Fig. 5 Few instances of incorrect detection of vehicles in **a** Wet snow, **b** Normal weather and **c** Blizzard conditions. The red bounding box indicates the ground-truth, and the green one indicates the detecting box

studies relied upon non-public data sets. The performance comparison of the present system with few existing studies relied on CDNet 2014 and LISA 2010 public data sets is presented in Table 4.

5 Conclusion and Future Scope

Detecting and tracking on-road vehicles from natural scene images in bad weather conditions is a challenging task because of the poor visibility of the target object. The present investigation proposes a new method of on-road vehicle detection by

Table 4 Comparative performance analysis with few existing studies

Dataset	Reference	Method	Precision (%)	Processing speed (in fps)
CDNet 2014	Hadi et al. [22]	Adaptive search window	90.49	NA
CDNet 2014	Proposed method	Modified YOLO	90.79 (overall)	43
LISA 2010	Zhou et al. [12]	Shallow CNN	79.41	4
LISA 2010	Proposed method	Modified YOLO	93.08	43

modifying the structure of YOLO model. The proposed strategy is much faster than the conventional faster R-CNN method in detecting varying-sized vehicles in good as well as bad weather conditions because a single CNN predicts the detecting bounding boxes and the class probabilities for those boxes together in one evaluation in the proposed strategy. The results also demonstrate that the processing speed of the proposed detection strategy is more than the other existing methods on LISA 2010 data set due to the same reason as mentioned above, and the proposed vehicle detection strategy outperforms the existing systems in this regard utilizing the CDNet 2014 and LISA 2010 public data sets. There is still room for the improvement of the performance of the present system, and the future work will be devoted to achieve this goal. The attempt will also be made in future to further increase the processing speed of the system.

References

1. Singh V, Srivastava A, Kumar S, Ghosh R (2019) A structural feature based automatic vehicle classification system at toll plaza. In: Proceedings of the 4th international conference on internet of things and connected technologies, Jaipur, India, pp 1–10
2. Ghosh R, Thakre S, Kumar P (2018) A vehicle number plate recognition system using region-of-interest based filtering method. In: Proceedings of the 2018 conference on information and communication technology, Jabalpur, India, pp 1–6
3. Sun Z, Bebis G, Miller R (2006) Monocular precrash vehicle detection: features and classifiers. *IEEE Trans Image Process* 15(7):2019–2034
4. Cheon M, Lee W, Yoon C, Park M (2012) Vision-based vehicle detection system with consideration of the detecting location. *IEEE Trans Intell Transp Syst* 13(3):1243–1252
5. Sivaraman S, Trivedi MM (2014) Active learning for on-road vehicle detection: a comparative study. *Mach Vis Appl* 25(3):599–611
6. Khairdoost N, Monadjemi SA, Jamshidi K (2013) Front and rear vehicle detection using hypothesis generation and verification. *Sig Image Process* 4(4):31–50
7. Yan G, Yu M, Yu Y, Fan L (2016) Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification. *Int J Light Electron Opt* 127(19):7941–7951
8. Wen X, Shao L, Fang W, Xue Y (2015) Efficient feature selection and classification for vehicle detection. *IEEE Trans Circ Syst Video Technol* 25(3):508–517
9. Ming Q, Jo KH (2011) Vehicle detection using tail light segmentation. In: Proceedings of the 6th international forum on strategic technology, Harbin, China, pp 729–732

10. Mohamed A, Issam A, Mohamed B, Abdellatif B (2015) Real-time detection of vehicles using the Haar-like features and artificial neuron networks. *Procedia Comput. Sci.* 73:24–31
11. Matthews N, An P, Charnley D, Harris C (1996) Vehicle detection and recognition in greyscale imagery. *Control Eng. Pract.* 4(4):473–479
12. Zhou Y, Liu L, Shao L, Mellor M (2016) DAVE: a unified framework for fast vehicle detection and annotation. In: Proceedings of the European conference on computer vision, Amsterdam, Netherlands, pp 278–293
13. Fan Q, Brown L, Smith J (2016) A closer look at faster R-CNN for vehicle detection. In: Proceedings of the IEEE intelligent vehicles symposium, pp 124–129
14. Redmon J, Divvala SK, Girshick RB, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition
15. Wang Y, Jodoin PM, Porikli F, Konrad J, Benezeth Y, Ishwar P (2014) CDnet 2014:an expanded change detection benchmark dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 387–394
16. Sivaraman S, Trivedi M (2010) A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Trans Intell Transp Syst* 11(2):267–276
17. Chan Y, Huang S, Fu L, Hsiao P (2007) Vehicle detection under various lighting conditions by incorporating particle filter. In: Proceedings of the IEEE intelligent transportation systems conference, Seattle, USA, pp 534–539
18. Chellappa R, Qian G, Zheng Q (2004) Vehicle detection and tracking using acoustic and video sensors. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, Montreal, Canada, pp 793–796
19. Bertozzi M, Broggi A, Castelluccio S (1997) A real-time oriented system for vehicle detection. *J Syst Arch* 43(1–5):317–325
20. Tian Y, Dong H, Jia L, Li S (2014) A vehicle re-identification algorithm based on multi-sensor correlation. *J Zhejiang Univ Sci C* 15(5):372–382
21. Haselhoff A, Kummert A (2009) An evolutionary optimized vehicle tracker in collaboration with a detection system. In: Proceedings of the IEEE intelligent transportation systems conference, St. Louis, USA
22. Hadi RA, George LE, Mohammed MJ (2017) A computationally economic novel approach for real-time moving multi-vehicle detection and tracking toward efficient traffic surveillance. *Arab J Sci Eng* 42:817–831

Chapter 6

Different Techniques Used in Smart Traffic Light Management System Using CCTV



Riddhika Rawat, ShrutiKA Singh, and Sumit Kumar

1 Introduction

In smart traffic lights system using CCTV cameras, the CCTVs are placed at the junctions of the traffic lights for capturing images. The primary step is image acquisition which includes capturing and converting an image to grayscale or binary format, for smooth processing of data. After this, the image is enhanced and segmented by applying different techniques and algorithms.

In this paper, we have reviewed the recent researches done in this particular field. We have discussed the new approaches, techniques, and algorithms used in the implementation of real-time-based systems. All the limitations of the researches and mechanisms to overcome those limitations are also been discussed in the paper to help in future researches. The paper is discussed as follows: Sect. 2 contains the methodology used for a smart traffic light, Sect. 3 contains a detailed study of the cited papers along with the table analysis, and Sect. 4 contains the conclusion part.

2 Methodology

There are different techniques used for smart management of traffic light but before acknowledging them, here are some of the image processing steps used in every referenced model (Fig. 1):

- i. *Image acquisition:* Acquisition of an image is the primary step in a smart traffic managing system. The pictures are captured on a real-time basis with the help of different rotatory CCTV cameras installed at the junction of the traffic light.

R. Rawat (✉) · S. Singh · S. Kumar

Department of Computer Engineering, Women Institute of Technology, Dehradun, India

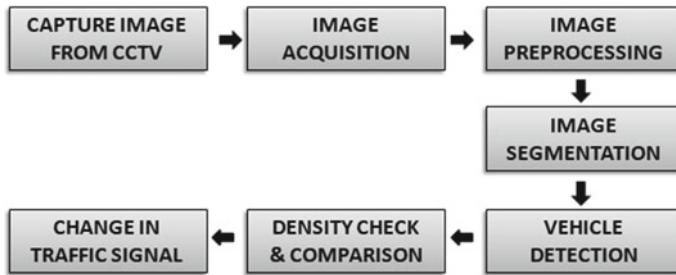


Fig. 1 Block diagram of methodologies used in smart traffic light using CCTV

- ii. *Image pre-processing*: Image preprocessing includes steps like image enhancement, conversion of images to grayscale, binary conversion, and brightening of image. Refining of images is processed here. It is used for having a lucid image of detected vehicles' compactness. The principle of image preprocessing is to get rid of the effect of interferences and haul out an improved forefront image
- iii. *Segmentation*: It comprises of techniques like edge detection and threshold method. In this process, the image is divided into several segments for simplifying and changing the appearance of the image. Segmentation is used for analyzing of the image easily. The image is partitioned, based on image features like pixel intensity value, color, texture, etc. Segmentation is used to detect objects and boundaries in the images.
- iv. *Vehicle detection*: Since the aim is to detect vehicles, vehicles in the image are detected here, and all the unwanted objects are removed or abstracted in the background. Detection involves edge detection and line detection. This helps to differentiate between vehicle and road.
- v. *Density check and image matching*: In this, background frames are converted to grayscale again and then images are compared after which they are subtracted to get hold of objects on the road. System [7] works on the green traffic light signal by comparing the images at the four-way junctions during busy hours and non-busy hours. Density is checked and the count of vehicles is made and stored in the database. Based on the results, the system makes changes in traffic light signals.

3 Techniques Used for Smart Traffic Light System Using CCTV

3.1 Smart Traffic Lights Using Image Processing Algorithms

The system is designed in two modes [1]: Smart mode and manual mode (With and without wireless connection). Arduino Mega is used as the primary controller, whereas Arduino Uno is used for wireless and Bluetooth model connection. The

model is sub-categorized into four categories: Conventional model, smart intelligent model, WiFi model, and standby model [1].

In the conventional model, the system works like a normal timer-based traffic control system.

In the smart intelligent model, the system uses image processing techniques to identify the density and decides the time proportionality of the number of cars to pass.

In WiFi model, the traffic light can be changed to a wireless connection with the help of an android application which will be provided to the traffic warden only. This system will help in emergency conditions.

In standby model, the yellow light of all the lanes will be turned on which is an indication that there is no traffic.

3.2 Smart Traffic Lights Using Image Processing Algorithms

The implemented system checks the traffic only at daylight and provides three kinds of results. The result is given in the form of notification, whether the traffic condition is light, heavy, or medium. Step-by-step processes are implemented; the first masking technique [2] is used for deleting the unnecessary parts which are captured using CCTV cameras. Then, thresholding of the image followed by subtraction is done for finding the cars. Contour technique [2] is used for edge detection. In this paper, researchers have used a formula to find the number of cars:

$$\text{Range} = \text{Contour Size}/\text{Size of one car in range (in pixels form)}$$

Now, the average value and ratio are taken out using the masking technique. The ratio is categorized into three cases:

Case 1: If the ratio is 1, then that is considered as a traffic jam, i.e., heavy traffic.

Case 2: If the ratio is between 0.4 and 0.7, then that is considered as medium traffic.

Case 3: If the ratio is less than 0.4, then that is considered light traffic.

According to the above cases, the notification is provided and traffic data is stored in a database.

3.3 Implementation of Image Processing in Real-Time Traffic Light Control

The traffic system is formed with the idea of keeping urban area traffic congestion in mind. The implemented system comprises hardware, software, and interfacing

model. Parallel port drivers are used for interfacing. In this implemented system, gamma correction is done for image enhancement. Prewitt edge detection [3] operator is used for edge detection. After edge detection, the reference image and real-time image matching is done and following are the result:

- If matching is in the range of 0–10%, then green light is ‘ON’ for 90 s.
- If matching is in the range of 10–50%, then green light is ‘ON’ for 60 s.
- If matching is in the range of 50–70%, then green light is ‘ON’ for 30 s.
- If matching is in the range of 70–90%, then green light is ‘ON’ for 20 s.
- If matching is in the range of 90–100%, then red light is ‘ON’ for 60 s.

3.4 Real-Time Area-Based Traffic Density Estimation by Image Processing for Traffic Signal Control System: Bangladesh Perspective

A descriptive case study of detecting vehicles using the Canny algorithm [5] is presented. Canny uses two thresholding [13]: upper and lower thresholding. If the gradient is above the upper threshold, the pixel is accepted, and if below the threshold, then rejected. If gradient values are between thresholds then accepted, only if connected to a pixel. Based on edge detection of objects, the vehicles are counted, and change in traffic light takes place according to the vehicle count.

3.5 Smart Traffic Congestion Control System

Congestion control system uses Otsu method [7] for calculating the threshold, Gaussian filter [7] for removing noise followed by Weiner filter, blob analysis is done, and final count of the vehicles is calculated. Otsu’s principle [7] is the main principle used for counting the automobiles. Here, two images are taken as input, image 1 contains an empty road and image 2 contains a road full of vehicles. Then, these two images are subtracted and converted into binary. The density of the vehicles is calculated by the formula

$$\text{Density of vehicle} = (\text{Number of objects}) / (\text{Total size of image})$$

Based on this density, the traffic light is adjusted for the peak and non-peak hours, and the result obtained is as follows:

Green signal timings for peak hours:

- If density is between 0 and 15%, then 60 s red
- If density is between 15 and 25%, then 20 s green
- If density is between 25 and 45% then, 50 s green
- If density is between 45 and 60%, then 60 s green

If density is greater than 60%, then 70 s green.

Green signal timings for non-peak hours:

If density is between 0 and 15%, then 60 s red

If density is between 15 and 25%, then 20 s green

If density is between 25 and 45%, then 50 s green

If density is between 35 and 45%, then 50 s green

If density is between 45 and 80%, then 60 s green

If density is greater than 80%, then 70 s green.

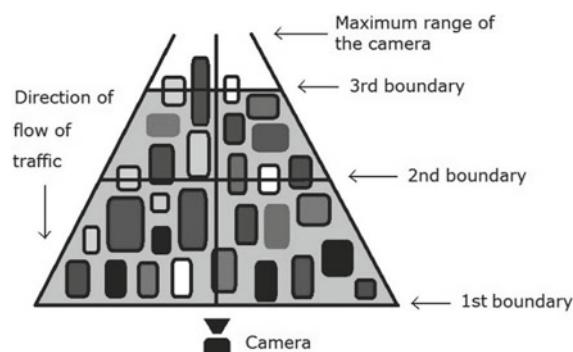
This system gives the approximate result of 90–93% in calculating the density and the controlling of green lights 3.2 Naive Bayes.

3.6 Automatic Control of Road Traffic Using Video Processing

High definition camera is connected and interfaced with the Raspberry Pi through an onboard HDMI port. The Raspberry Pi is connected to the server containing MATLAB. MATLAB communicates with Raspberry Pi for capturing video from peripherals and process on it. The contour tracking algorithm is implemented as it detects the object's boundary continuously in every frame. The researcher introduces an imaginary trapezium [8] by dividing the road into four imaginary quadrants (Fig. 2).

Now, when vehicles cover the trapezium [8] completely or reach the 3rd boundary, MATLAB reads it and sends signals to Raspberry Pi to turn on the green light for 30 s for that road. After 30 s if the threshold still occupies vehicles completely, then it continues with a green light for another 30 s, otherwise turns the red light on and check for other lanes.

Fig. 2 (See Fig. 4 in Ashwin S et al. 2017) Imaginary trapezium, vehicles at the traffic signals waiting, captured by the camera



3.7 CCTV Traffic Congestion Analysis at Pejompongan Using Case-Based Reasoning

This paper proposes an approach by considering the traffic conditions in Jakarta. The machine learning approach is applied using case-based reasoning (CBR) [5]. Stages used are retrieve, reuse, revise, and retain. It uses the case base maintenance (CBM) for the effective and accurate management of data storage. CBR algorithms are used to show the condition of roads empty (low), smooth (medium), or solid (high). Images acquired by the CCTV camera via website lewatmane.com, Jakarta, and then converted to binary. The binary pattern is matched to determine the condition of traffic on the road. The results are as follows

- If low traffic pattern, then there will be no delay in the duration of traffic light
- If medium traffic pattern, then green light for 90 s
- If high traffic pattern, then green light for 30 s.

The binary is compared with the pattern on the CBM. The compared result shows the difference in lines. The less row line for the low traffic and more row line for the more congested traffic.

3.8 Smart Traffic Control System with Application of Image Processing Techniques

The technique used in this model is that the traffic density is calculated by the number of pixels instead of counting the number of vehicles. Traffic control algorithm [12] is implemented in the traffic system. The traffic cycle is used for controlling the traffic light, i.e., denser the traffic, the longer will be the traffic cycle. Similarly, the traffic cycle will be short during less traffic. The total area covered by the vehicles is considered as traffic density. A variable traffic cycle is maintained based on total traffic density. Weight is also determined for the roads by traffic density. Weight is a time allocation used for passing traffic with higher density. It is determined for every road, and then the traffic cycle is weighted (Table 1).

$$W_i = \frac{TD_i}{\sum_{k=1}^n TD_k}$$

4 Conclusion

With the growth and advancement of the smart city programs, smart traffic lights using CCTV cameras are becoming a part of development. They are comparatively

Table 1 Summary of the analysis

Table	Context/year	Hardware/software/methods used by systems	Case study	Limitations
[1]	Smart traffic lights using image processing algorithms/2019	OpenCV library along with C++, MATLAB environment. Android Programming Arduino Mega (Primary controller), Arduino Uno, H-05 Bluetooth model	The system has four sub-models, which comprises of a conventional system, intelligent system, WiFi system, and standby system. These systems are there for changing the traffic light according to the situation. Useful in emergencies	Construction of a centralized monitoring station for the working of this model.
[2]	The traffic congestion investigating control system using CCTV cameras/2017	Masking technique, thresholding, median filtration, dilation and erosion operation, contour-based method	The system checks traffic on a day when there is static light density and is used for transportation path planning	The implemented system cannot work in rainy conditions
[3]	Image processing in real-time traffic light control/2011	Power law transformation(gamma correction), Prewitt edge detection, MATLAB version 7.8	Takes empty road as a referenced image (visualizes image) and matching is done with real-time image. The percentage of matching, control traffic light duration	Need improvement for 100% accurate result
[4]	Automated traffic monitoring using image vision/2018	Kernel-based edge detection, algorithm to detect the perimeter of close figures, ML, round robin algorithm, longest remaining job first algorithm	Vehicles are counted by detecting the no. of windshields. Computational time and starvation time are also low	The algorithm can be updated for better output

(continued)

Table 1 (continued)

Table	Context/year	Hardware/software/methods used by systems	Case study	Limitations
[5]	Area-based traffic density estimation with the help of image processing for traffic signal control system: Bangladesh Perspective/2015	Canny algorithm for edge detection	Detect traffic density according to the edges of the vehicle	Beneficial only in some urban areas
[6]	Adaptive and robust traffic surveillance system for urban intersections on embedded platform/2014	Motion detection approach, finite state machine	The count of a vehicle is done using motion detection and the waiting time of vehicles' is optimized according to a finite state machine	For faster processing image, processing should be done on an embedded platform
[7]	Smart traffic congestion control system/2019	MATLAB, Wiener filter, Otsu principle, Gaussian density function, blob analysis	Taking an empty road image as a reference image. Based on the density and count of vehicles, the time slot is adjusted	For storing a large amount of data, cloud computing can be used
[8]	Video processing for automatic control of road traffic/2017	Video processing, Raspberry Pi, MATLAB, threshold, contour tracking algorithm	During bad weather conditions, thermal cameras are used. The pedestrian call button is used to stop the traffic flow for 10 s	The system is expensive and the processing time is more. Instead of manually adjusting the green light for the ambulance, a camera can be used to detect the ambulance
[9]	Smart control of traffic light signal using image processing/2017	Four webcams, two Arduino Uno, MATLAB, morphological operations, thresholding, Wiener filter	A Wiener filter is used to reduce noise. Traffic density and duration of green light are calculated	More accuracy is needed

(continued)

Table 1 (continued)

Table	Context/year	Hardware/software/methods used by systems	Case study	Limitations
[10]	Density-based traffic control system using image processing/2018	Otsu principle, Gaussian, Wiener filter, blob analysis	Seven segment display screen to display the count of vehicles and also to display the time allotted	Weather conditions like fog and rain are not taken into consideration
[11]	CCTV traffic congestion analysis at Pejompongan using case-based reasoning/2018	Machine learning approach using case-based reasoning, case-based management algorithm, case-based reasoning algorithm	Traffic light control system using a CBR approach. Converting the images to binary and matching and comparing the binary patterns	Weather conditions and night time glow is not taken into consideration, integration with traffic light system
[12]	Smart traffic control system with application of image processing techniques/2014	Webcam, MATLAB, ATMEGA8 microcontroller Universal Synchronous Asynchronous Receiver Transmitter (USART) module, Sobel edge detection	The total no. of pixels of the images is taken into consideration for determining the traffic density. The area occupied by the vehicles is taken as the traffic density	The model can be extended to include a large number of interconnected traffic junctions

fast, accurate, time-saving, and economical in comparison to timer-based and sensor-based traffic systems. Various image processing techniques [5, 10], automatic and manual ideas discussed, are very convenient for controlling the traffic intelligently and reducing manpower. The limitation that was observed in every paper was that the systems developed are only able to work accurately during the day time when the image capturing is easy, whereas at night time and during rain/fog condition [10, 11], the system does not provide efficient result and is facing problems. For overcoming this situation, night object detection algorithms can be used in future researches.

Some limitations are illustrated below:

- Construction of a centralized monitoring station.
- The implemented preexisting systems cannot work in rainy conditions.
- Night time glow is not taken into consideration, integration with traffic light system.

Problems to be worked on are:

- Inefficient working of image processing techniques during harsh climatic conditions like rain, fog, and night time.

By reviewing the cited papers, the following techniques and methods are interpreted as the suitable ones:

- Implementation of a system with the help of MATLAB provides fast and desirable results [7, 12], whereas the Raspberry Pi system is less cost-efficient [8].
- The machine learning approach using CBR proves to be beneficial for congestion control.
- Canny algorithm along with Gaussian filtering is suited for edge detection [5, 7], which helps provide approximated vehicle count.

References

1. Elkhatab MM, Alsamna Electrical AS, Adwan AI, Abu-Hudrouss AM Smart traffic lights using image processing. 978-1-5386-6291-5/19/\$31.00 ©2019 IEEE
2. Eamthanakul B, Ketcham M, Chumuang N The traffic congestion investigating system by image processing from CCTV camera. 978-1-5090-5210-3/17/\$31.00 ©2017 IEEE
3. Choudekar P, Banerjee S, Muju MK Implementation of image processing in real-time traffic light control. 978-1-4244-8679-3/11/\$26.00 ©2011 IEEE
4. Krishnamoorthy R, Manickam S Automated traffic monitoring using image vision. In: Proceedings of the 2nd international conference on inventive communication and computational technologies (ICICCT 2018). IEEE Xplore Compliant—Part Number: CFP18BAC-ART; ISBN 978-1-5386-1974-2
5. Uddin S, Das AK, Taleb MdA (2015) Real-time area based traffic density estimation by image processing for traffic signal control system: Bangladesh perspective. In: 2nd international conference on electrical engineering and information and communication technology (ICEEICT). Jahangirnagar University, Dhaka-1342, Bangladesh, 21–23 May 2015
6. Bharade AD, Gaopande SS (2014) Robust and adaptive traffic surveillance system for urban intersections on embedded platform. In: 2014 annual IEEE India conference (INDICON)
7. Balu S, Priyadharsini C Smart traffic congestion control system. In: Proceedings of the third international conference on computing methodologies and communication (ICCMC 2019). 978-1-5386-7808-4/19/\$31.00 ©2019 IEEE
8. Ashwin S, Vasist RA, Hiremath SS, Lakshmi HR Automatic control of road traffic using video processing. 978-1-5386-0569-1\$31.00©2017 IEEE
9. Khushi Smart control of traffic light signal using image processing. In: International conference on current trends in computer, electrical, electronics and communication (ICCTCEEC-2017). 978-1-5386-3243-7/17/\$31.00 ©2017 IEEE
10. Prakash UE, Vishnupriya KT, Thankappan A, Balakrishnan AA (2018) Density based traffic control system using image processing. In: Proceedings of 2018 international conference on emerging trends and innovations in engineering and technological research (ICETIETR). 978-1-5386-5744-7/18/\$31.00 ©2018 IEEE
11. Surjandy, Soeparno H, Anindra F, Napitupulu TA CCTV traffic congestion analysis at Pejompongan using case based reasoning. 978-1-5386-0954-5/18/\$31.00 ©2018 IEEE

12. Munir Hasan Md, Saha G, Hoque A, Majumder MdB (2014) Smart traffic control system with application of image processing techniques. In: 3rd international conference on informatics, electronics and vision 2014. 978-1-4799-5180-2/14/\$31.00 ©2014 IEEE
13. Rong W, Li Z, Zhang W, Sun L (2014) An improved canny edge detection algorithm. In: Proceedings of 2014 IEEE international conference on mechatronics and automation, 3–6 August, Tianjin, China

Chapter 7

A Novel Rule-Based Expert System for Penalty Prediction for Two-Wheeler's Traffic Rules Violation in India



Anoop Sahani, Niranjan Panigrahi, Jagadish Mohanty, Anita Moharana,
and Diptimayee Sahoo

1 Introduction

India is one of the most quickly developing economies on the planet, and furthermore, it is a country with the second biggest street arrangement. With the expansion in populace, changes in the occupational structure living space, and necessity of relaxation, the demand for passenger transportation services increase. In this way, to travel peacefully, individuals are picking driving in their own vehicles. Thus, more vehicles are going ahead streets, and lastly, traffic framework turns into a problem that is begging to be addressed. Nowdays, it is a massive challenge for traffic administration to provide state-of-the-art traffic framework, because of increment in personal vehicles and over-loaded streets in the country. There are numerous issues emerging in the traffic framework as:

- Absence of legitimate usage of traffic rules.
- Violation of rules and guidelines of the traffic.
- Regional inequality in economic and financial development.

Due to the above problems, genuine traffic offenses are expanding steadily, and it is making substantial misfortune to the human society. So for that the new amendment laws are made which are tight and hard for the individuals who disrupting traffic guidelines [1]. Sometimes, people are becoming prey to these traffic violations due to lack of awareness about new amendments. This necessitates an automated, centralized dissemination and support system for the common citizen to get knowledge about new traffic rules and their violation penalties. This will not only act as information providing platform but also increases awareness and thus, prevent traffic violation.

A. Sahani (✉) · N. Panigrahi · J. Mohanty · A. Moharana · D. Sahoo
Department of CSE, Parala Maharaja Engineering College, Berhampur, Odisha, India

In recent years, AI-based tools and approaches are mostly used in designing such decision support system for traffic violations [2]. One such tool in AI is expert system which acts basically on KB and inference engine [3]. In this paper, an expert system is proposed to compute all the penalties for a two-wheeler's traffic rules violator who abuses the principles and guideline of the traffic. The proposed system will be helpful in two ways: (i) as a decision support system for traffic officers to compute penalties in an automated way with less time, (ii) as an awareness system for general public about traffic rules violation and make the citizenship mindful and safe. The significant contributions of this paper are:

- The proposed system can compute the penalties for a two-wheeler's traffic rules violator as per Motor Vehicle Act 2019 of India.
- A KB is proposed and decision tree-based methodology is adopted for inference.
- A thorough testing is performed on ES-builder, a web-based expert system shell, to show its viability.

2 Related Work

Recently, numbers of works related to traffic management using AI methods are reported since it is a major evolving issue [2, 4, 5]. In India, the standards and guidelines are made so strict yet some way or another resident violets it. Likewise, the populace and regional inequality is one of the significant reasons for the traffic issue. So this segment features some of the relevant work in the traffic framework. In [3], Mohammad *et al.* have proposed an expert system for traffic signal controlling based on belief theory. The major contribution lies in handling uncertainty in traffic signal. In [6], Zhang *et al.* have proposed an expert system for unmanned vehicles using automatic driving traffic rules. It represents the development of a model knowledge-based expert system (KBES) for choosing appropriate traffic control procedures and management techniques around highway work zones. In [7], a decision tree-based approach is proposed to analyze speeding violation behavior in Wujiang city of China. Different factors like number plate, season, location and rainfall are taken into consideration for prediction. A spatio-temporal pattern of traffic violation is studied in [8] using kernel density estimation at 69 traffic intersections of Wujiang city of China. In [9], ML methods are adapted to predict risky and aggressive spatio-temporal behavior of driving by taxi drivers.

3 Problem Statement

The objective of our proposed problem domain is to design an expert system which can figure out all possible penalties of the defaulter who abuses the standards and rules of the traffic while driving two-wheelers which are coming under motor cycle with gear (MCWG) category. The problem statement can be formally stated as.

Using set representation, let us, $T = \{T_1, T_2, \dots, T_n\}$, represents set of traffic rules for MCWG category. The proposed system will predict the output which can be represented as a set $P = \{P_1, P_2, \dots, P_n\}$.

Where P = set of penalties calculated for violation of traffic rules for MCWG as per Government of India guidelines.

The design stage includes making a knowledgebase (KB) which consists of all rules related to traffic violation and developing an inference engine (IE) to derive the output P . The knowledgebase is made by focusing on the traffic rules and guidelines made by the Government under the Motor Vehicle Act. The rules thus collected as per Motor Vehicle Act are stored as knowledge in KB by using rule-based knowledge representation methods.

4 Proposed Expert System

Based on the above depicted objective, the following sections present the proposed architectural design in Fig. 1, the decision tree-based inference mechanism given in Fig. 2, and a few rules out of proposed 360 rules in the KB is shown in Fig. 3.

4.1 System Architecture

The knowledgebase (KB) of the penalties for abusing the standards and guidelines of the traffic expert system contains set of rules to restrict the violations and to make the

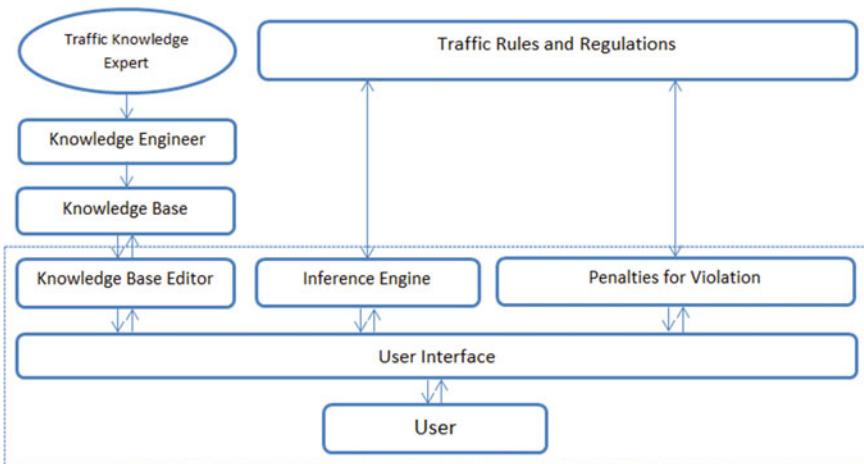


Fig. 1 Architecture of the proposed expert system

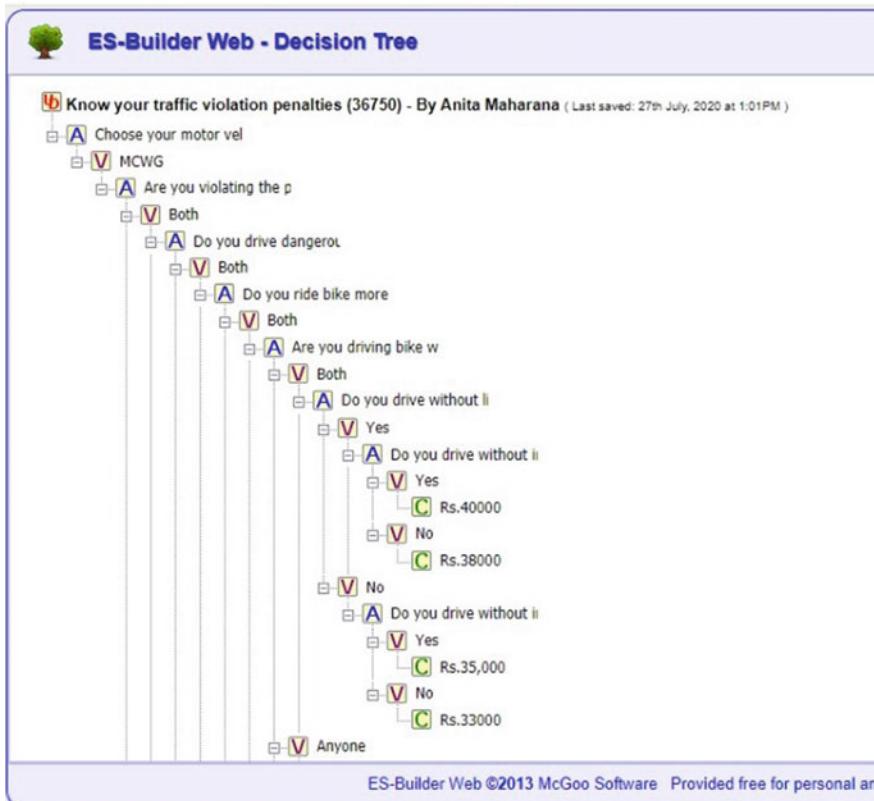


Fig. 2 Snapshot of part of the proposed decision tree

citizenship careful and safe. The client or the defaulter enters the data like absence of vehicle documentation or disregarding traffic through the user interface. At that point, the system generates the penalties for abusing relating rules, utilizing inference engine (IE), which the client or defaulter must need to comply with the standards for their wellbeing.

4.2 Inference Engine (IE)

The proposed IE utilizes decision tree mechanism with forward chaining procedure to reach to a conclusion. Here, inference engine helps in matching the individual's selected information with the rules contained in the knowledgebase to show the possible output as shown in Fig. 2. At the point when the client will get to the expert system, the system will ask a few questions with respect to the traffic rules and

Knowledge Base Rules	
#	Rule
1	IF choose your motor vehicle type? mCWG AND are you violating the principles and guidelines of the road or any other general offenses? both AND do you drive dangerously or use mobile phones while driving? both AND do you ride bike more than two riders or driving without helmet? both AND are you driving bike while drunk or without vehicle pollution? both AND do you drive without licence? yes AND do you drive without insurance? yes THEN the Penalties is Rs.40000.
2	IF choose your motor vehicle type? mCWG AND are you violating the principles and guidelines of the road or any other general offenses? both AND do you drive dangerously or use mobile phones while driving? both AND do you ride bike more than two riders or driving without helmet? both AND are you driving bike while drunk or without vehicle pollution? both AND do you drive without licence? yes AND do you drive without insurance? no THEN the Penalties is Rs.38000.
3	IF choose your motor vehicle type? mCWG AND are you violating the principles and guidelines of the road or any other general offenses? both AND do you drive dangerously or use mobile phones while driving? both AND do you ride bike more than two riders or driving without helmet? both AND are you driving bike while drunk or without vehicle pollution? both AND do you drive without licence? no AND do you drive without insurance? yes THEN the Penalties is Rs.35,000.
4	IF choose your motor vehicle type? mCWG AND are you violating the principles and guidelines of the road or any other general offenses? both AND do you drive dangerously or use mobile phones while driving? both AND do you ride bike more than two riders or driving without helmet? both AND are you driving bike while drunk or without vehicle pollution? both AND do you drive without licence? no AND do you drive without insurance? no THEN the Penalties is Rs.33000.

Fig. 3 Snapshot of example rule set for the system

guidelines. According to the users selected response, the system will show absolute penalty as an output by using forward chaining on the proposed decision tree.

4.3 Knowledgebase (KB)

The knowledgebase consists of 360 rules. It has used IF-THEN—rule method of knowledge representation, also known as production rule method. A sample rule set is shown in Fig. 3.

5 Implementation and Analysis

This section shows the implementation of the proposed system and performance analysis by considering the system as multi-label classification.

5.1 *Implementation*

The open-source web-based expert system shell, and ES-builder [10] is used to implement the proposed system. A possible testing path of the system is shown in Fig. 4. To verify all possible paths, users can visit our proposed system at: <https://www.mcgoo.com.au/esbuilder/viewer/viewES.php?es=afd042f8e5ce19933e948f33589ccd42>.

5.2 *Analysis*

To analyze the performance of the proposed system, the problem is mapped with multi-label classification problem. The queries asked by the system, i.e., traffic violation categories, are mapped to attributes and penalty amounts are mapped to labels of the multi-label classification problem. Using Scikit learn [11], the performance is evaluated using problem relevance and standard parameters settings as given in Table 1. The accuracy score is evaluated as 75%.

6 Conclusion and Future Work

The proposed expert system is based on the penalties for two-wheelers for mishandling rules and guidelines of the traffic violation in India. It can undoubtedly be accessed by anybody, anytime, and anywhere. By utilizing this expert system, it helps the people to know the current penalty and the resident will self-get cautious by paying penalty, and furthermore, it helps to limit further violation. It could infer 360 possible conclusions which can only compute the penalty for disregarding the rules. In future,, the system can be additionally upgraded with more number of rules and can be intended for other motor vehicles like LMV, HMV and that could be inserted in knowledgebase. It is planned to migrate the proposed system on cloud-mobile-based technology for more flexible and cost-optimal adaptation in real-world application.

<p>The Traffic Rules and guidelines in India are set according to the New Motor Vehicle Act 2016. Under the modified amendment, the penalties for criminal traffic offenses have been redefined entirely and have turned more severe than any time in recent memory. This expert system is proposed to compute the penalties of the vehicles as per the standards and rules of the state.</p> <p>Note: There are 250 possible conditions in the expert system.</p> <p>Know your penalties for traffic rule violation</p>	<p>Choose your motor vehicle type?</p> <p>All types of two-wheeler vehicle which are categorized under motor cycle with or without gear (MCG).</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> MCG <input type="radio"/> Two-wheeler without gear
<p>Are you violating the principles and guidelines of the road or any other general offenses?</p> <p>Under the new motor vehicle act of 2016 and 2017, for the risk of General offense and Violating the principle and guidelines of the road.</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> None <input type="radio"/> Some <input type="radio"/> Both <p>Please select[None] if you are not violate any risk.</p> <p>Please select[Some] if you are violate above anyone risk.</p> <p>Please select[Both] if you are violate above both risk.</p> <p>Decision Making:</p> <p>Choose your motor vehicle type? <input checked="" type="radio"/> MCG Is your violating the principles and guidelines of the road or any other general offense? <input checked="" type="radio"/> Both Please click on the checkbox below to refer to that decision.</p>	<p>Do you drive dangerously or use mobile phones while driving?</p> <p>Under the new motor vehicle act 2016 and 2017, for the risk of dangerous driving and using mobile phone while driving.</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> None <input type="radio"/> Some <input type="radio"/> Both <p>Please select[None] if you are not violate any risk.</p> <p>Please select[Some] if you are violate above anyone risk.</p> <p>Please select[Both] if you are violate above both risk.</p> <p>Decision Making:</p> <p>Choose your motor vehicle type? <input checked="" type="radio"/> MCG Is your violating the principles and guidelines of the road or any other general offense? <input checked="" type="radio"/> Both Please click on the checkbox below to refer to that decision.</p>
<p>Are you driving biker while drunk or without helmet protection?</p> <p>Under the new motor vehicle act 2016 and 2017, for the risk of酒醉驾驶 and 骑行不戴头盔。</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> None <input type="radio"/> Some <input type="radio"/> Both <p>Please select[None] if you are not violate any risk.</p> <p>Please select[Some] if you are violate above anyone risk.</p> <p>Please select[Both] if you are violate above both risk.</p> <p>Decision Making:</p> <p>Choose your motor vehicle type? <input checked="" type="radio"/> MCG Is your violating the principles and guidelines of the road or any other general offense? <input checked="" type="radio"/> Both Do you drive dangerously or use mobile phones while driving? <input checked="" type="radio"/> Both Do you drink more than two glasses of alcohol before driving? <input checked="" type="radio"/> Both Please click on the checkbox below to refer to that decision.</p>	<p>Do you ride biker more than two riders or driving without helmet?</p> <p>Under the new motor vehicle act 2016 and 2017, for the risk of 骑行超载 and 骑行不戴头盔.</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> None <input type="radio"/> Both <p>Please select[None] if you are not violate any risk.</p> <p>Please select[Both] if you are violate above both risk.</p> <p>Please select[Both] if you are violate above anyone risk.</p> <p>Decision Making:</p> <p>Choose your motor vehicle type? <input checked="" type="radio"/> MCG Is your violating the principles and guidelines of the road or any other general offense? <input checked="" type="radio"/> Both Do you drive dangerously or use mobile phones while driving? <input checked="" type="radio"/> Both Do you drink more than two glasses of alcohol before driving? <input checked="" type="radio"/> Both Please click on the checkbox below to refer to that decision.</p>

Fig. 4 Snapshots of testing of a possible path of the system in ES-builder

ES-Builder Web

Expert System - Know your traffic violation penalties by Anita Maharana

Do you drive without license?

[Under the new motor vehicle act 2013 for the sake of driving without license.]

- * No
- * Yes

Decision History:

Choose your motor vehicle type?	MCWG
Are you violating the principles and guidelines of the road or any other general offenses?	both
Do you drive dangerously or use mobile phones while driving?	both
Do you ride bike more than two riders or driving without helmet?	both
Are you driving bike while drunk or without vehicle pollution?	both
Do you drive without license?	yes
Do you drive without insurance?	yes

Top: Click on an option will allow to return to the decision.

[Return to the Know your traffic violation penalties Expert System Title Page](#)

ES-Builder Web

Expert System - Know your traffic violation penalties by Anita Maharana

Based on the responses you have made:
The Penalties is Rs.40000

Conclusion Notes:

Expert System Rule:

If choose your motor vehicle type? mcwg
AND Are you violating the principles and guidelines of the road or any other general offenses? both
AND Do you drive dangerously or use mobile phones while driving? both
AND Do you ride bike more than two riders or driving without helmet? both
AND Are you driving bike while drunk or without vehicle pollution? both
AND Do you drive without license? yes
AND Do you drive without insurance? yes
THEN The Penalties is Rs.40000.

[Return to the Know your traffic violation penalties Expert System Title Page](#)

ES-Builder Web ©2013 McGoo Software Provided free for personal and academic use. **ES-Builder Web Help** **Privacy Policy**

Fig. 4 (continued)**Table 1** Parameters for analysis of the proposed system

Parameter	Value
n_labels	20
Problem transform	BinaryRelevance
Classifier	DecisionTreeClassifier
Test_size	0.2

References

1. <http://egazette.nic.in/WriteReadData/2019/210413.pdf>
2. Li J, Cheng H, Guo H (2018) Survey on artificial intelligence for vehicles. *Automot Innov* 1:2–14. <https://doi.org/10.1007/s42154-018-0009-9>
3. Jackson P (1999) Introduction to expert system. Addison-Wesley Publishing
4. Fang A, Qiu C, Zhao L, Jin Y (2018) Driver risk assessment using traffic violation and accident data by machine learning approaches. In: 3rd IEEE international conference on intelligent transportation engineering (ICITE), pp 291–295. <https://doi.org/10.1109/icite.2018.8492665>
5. Zahid M, Chen Y, Jamal A et al (2020) Adopting machine learning and spatial analysis techniques for driver risk assessment: insights from a case study. *Int J Environ Res Pub Health* 17(5193)
6. Hossain MS, Sinha H, Mustafa R (2015) A belief rule based expert system to control traffic signals under uncertainty. In: 2015 international conference on computer and information engineering, ICCIE-2015, pp 83–86. <https://doi.org/10.1109/ccie.2015.7399323>
7. Zhang Z, Jiang Q, Sun B et al (2018) Researches on expert system for automatic driving traffic rules of unmanned vehicle. *J Phys* 1069:12–16
8. Zahid M, Chen Y, Khan S et al (2020) Predicting risky and aggressive driving behavior among taxi drivers: do spatio-temporal attributes matter? *Int J Environ Res Pub Health* 17(3937)
9. Li Y, Aty MA, Yuan J et al (2020) Analyzing traffic violation behavior at urban intersections: a spatio-temporal kernel density estimation approach using automated enforcement system data. *Acc Anal Prev* 141
10. <https://www.mcgoo.com.au/esbuilder/index.php>
11. <https://scikit-learn.org/stable>

Chapter 8

Hybrid Material-Based Dual-Band Yagi-Uda Antenna with Enhanced Gain for the Ku-Band Applications



Rajesh Yadav, Shailza Gotra, V. S. Pandey, and Brahmjit Singh

1 Introduction

In recent times, wireless communication technologies have recorded phenomenal growth [1]. In this growth trajectory, antenna designing has played a major role with miniaturization being one of the impactful factors. The miniaturization enables the antenna relevant for use on different applications such as GPS antenna [2], digital RF antenna [3], IC Packages [4], embedded-based antenna [5], VHF and UHF antenna [6], third generation mobile antenna [7]. In addition, it also requires a new class of materials for designing purpose. Many advanced materials are reported including smart material, EM material, anisotropic, metamaterial and graphene material. Intensive research is going on improvement of the properties and applications of material-based antennas. The lead (Pb), silicon (Si) and copper (Cu) have gained attention in the designing of material-based antennas. Lead has served as the suitable transition metal enabling change in the electromagnetic properties. It can serve the advantage of acquiring novel structures without adding new doped material [8]. Its specific properties include conductivity (4.8×10^6 S/m), Poisson's ratio (0.42), Young's modulus (14KN/MM²) and diffusivity (2.35×10^{-5} m²/s).

Various types of antennas have been reported for end-to-end communication. These include tapered slot leaky wave antennas [9, 10], log periodic antenna [11] and conformal CPW-fed slot antenna [12] providing end-fire radiation pattern. The Yagi-Uda due to its end-fire radiation pattern gained lot of attention[13]. It mainly consists of the three-driven elements, i.e., dipole, reflector and director. Previous designs are

R. Yadav (✉) · S. Gotra
Department of ECE, NIT Delhi, Delhi 110040, India

V. S. Pandey
Department of Applied Sciences, NIT Delhi, Delhi 110040, India

B. Singh
Department of ECE, NIT Kurukshetra, Kurukshetra 136119, India

bulky at lower frequency and faces hindrance due to wind loading. To reduce its size and weight, microstrip Yagi-Uda array antenna come up as a suitable substitution. The array units are the combination of microstrip and Yagi arrays [14]. Nowadays, it has been used as the substitute of wire dipole antenna. The key factors of Yagi-Uda antenna are its bandwidth, gain, directivity and front-to-back ratio. Normally, it provides less bandwidth and high gain due to inherent characteristic of microstrip sheet. Bandwidth can be enhanced using aperture coupled [15], stacked configuration [16] and multi-resonator [17] techniques. The gain can be enhanced using multiple director elements in Yagi-Uda antenna. Front-to-back ratio may be increased by using larger size of the reflector in comparison with the dipole element. A compact dual-band printed Yagi-Uda antenna was reported for GNSS and CMMB applications [18]. A compact microstrip fed planer dual-dipole antenna introduced for broadband applications may be used for Ku-band frequency range [19]. Implementation of Yagi patch antenna with dual-band response and pattern-reconfigurable characteristics is reported in [20]. This provides the beam-configurability through etching of different slots. A dual-band compact Yagi-Uda antenna was designed over an EBG ground plane [21]. A novel dual-band Yagi-Uda antenna was reported in [22]. The major limitations of these proposals are the difficulty level in obtaining the high gain and front-to-back ratio (FBR).

In this paper, we propose a hybrid material–lead–silicon–copper-based Yagi-Uda antenna. The Yagi-Uda elements are placed above the substrate in such a way that the conductivity of these elements is arranged in decreasing order. The dipole is excited via microstrip feedline connected by a conducting vias. The proposed antenna provides dual-band frequency response. It is operated with excitation of higher-order TM_{16} and TM_{26} modes offering good impedance matching with higher gain. Parametric analysis has been carried out to analyse its performance.

2 Antenna Design

Its consists of hybrid structure; lead–silicon–copper material is shown in Fig. 1. The geometry is composed of two symmetrical substrates of silicon dioxide (SiO_2) having relative permittivity, $\epsilon_r = 3.9$. The substrate-1 of dimensions $l \times w \times h_{s1}$ is placed above the ground plane. The $50\ \Omega$ microstrip feedline is sandwiched between two layers of substrates having dimensions $l_f \times w_f$. The driven and parasitic elements are placed above second substrate layer of dimensions $a \times b \times h_{s2}$. The driven elements of the antenna consist of reflector, dipole and director. These elements are arranged in the decreasing order of the conductivity varying from reflector to director. The element of reflector, dipole and director is made up of copper, lead and silicon, respectively, having thickness 0.035 mm and width w_E . The length of the reflector, dipole and director is l_r , l_{dp} and l_{dr} , respectively, as shown in Fig. 1a. The distance between two elements of the antenna is 18 mm. The dipole is connected to the PEC feedline through a copper vias using substrate-2 as shown in Fig. 1b. This vias mechanism improves the impedance matching of the antenna with direct contact

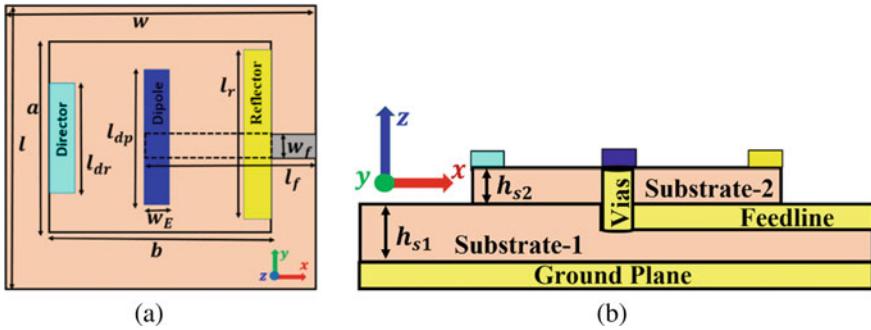


Fig. 1 Geometry of the proposed antenna **a** top view and **b** side view

Table 1 Dimensions of the proposed antenna geometry

Parameters	w	l	a	b	h_{s1}	h_{s2}	l_r	l_{dp}	l_{dr}	l_f	w_f	w_E
Dimension (mm)	81	65	45	45	1	0.25	33	31	28	42	3	3

between the feedline and lead-based dipole element. The dimensions of the proposed antenna are summarized in Table 1. The proposed antenna has been designed and numerically analysed using the CST software.

3 Results and Discussion

Frequency response of S_{11} -parameter and voltage standing wave ratio (VSWR) is shown in Fig. 2. Impedance matching is obtained at 15.5 and 16.1 GHz resonant frequencies that confirm the dual-band response. It provides -10 dB impedance bandwidth of 1.03%(15.6 – 15.44) and 2.05%(15.91 – 16.24) in the lower and upper bands, respectively, as shown in Fig. 2a, b shows the VSWR plot of the proposed antenna. VSWR response of dual-band feature shows perfect matching.

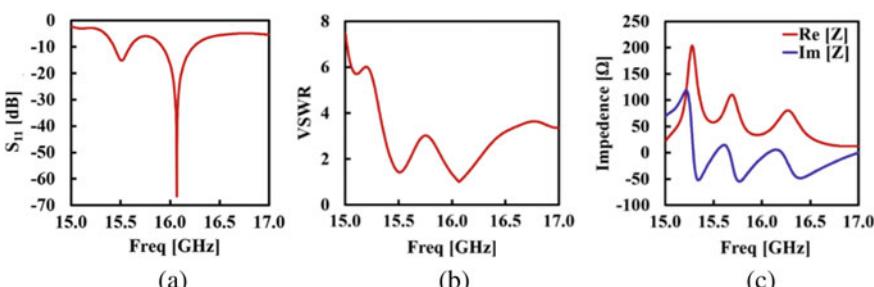


Fig. 2 Frequency response of **a** S_{11} parameter, **b** VSWR and **c** impedance plot of the antenna

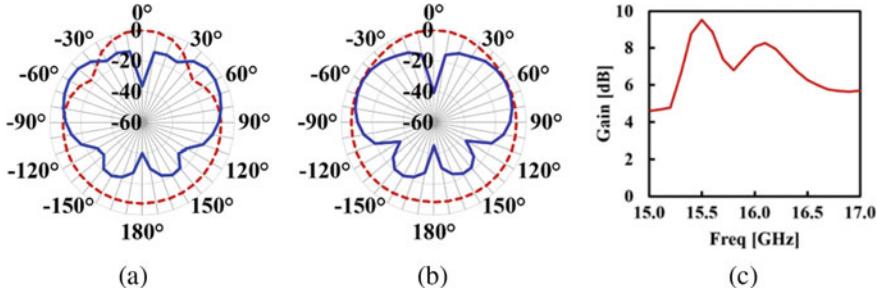


Fig. 3 Radiation pattern of the proposed antenna at **a** 15.5 GHz and **b** 16.1 GHz resonant frequency **c** gain versus frequency plot. Dotted line: co-polarized component, solid line: cross-polarization

Figure 2c shows the frequency response of the impedance plot with real and imaginary part. The imaginary part is approximately zero showing that the power is not stored in near-field region.

The radiation pattern of the proposed antenna was analysed in the far-field region. Figure 3a, b shows the 2D radiation pattern at 15.5 and 16.1 GHz resonant frequencies of the lower and upper operating bands. The proposed antenna provides the radiation pattern in the xz -plane at the resonant frequencies. The radiation pattern shows the co- and cross-polarization components having more than 20 dB difference. The frequency response of the realized gain has been studied in Fig. 3c. The proposed antenna provides the 9.52 dBi and 8.1 dBi in the lower and upper bands of the antenna, respectively.

The mode analysis of the proposed Yagi-Uda antenna based on hybrid material has been analysed by near-field region. Figure 4 shows the E-field distribution inside the driven element. The field distribution confirms the presence of horizontal magnetic dipole. Therefore, the proposed antenna excites the transverse magnetic (TM) mode. The nomenclature of the modes has been done in accordance with the variation in the field of different plane. Figure 4a, b shows the mode analysis at 15.5 GHz resonant frequency of the lower operating passband of the antenna. The field distribution shows the one variation along x -axis and six variations in y -axis confirming the TM_{16} mode. Similarly, Fig. 5a, b shows the mode analysis at 16.1 GHz resonant frequency of the upper operating passband. The field mechanism shows two variation along x -axis and six variations along y -axis. Hence, it confirms existence of TM_{26} mode. Figure 4c and Fig. 5c show the absolute E-field distribution in the xy -plane of the antenna.

4 Parametric Analysis

The antenna performance has been studied by changing its physical parameter like height of substrate and length of the Yagi-Uda element. The main parameter of this proposed antenna design is h_{s1} , h_{s2} , l_r , l_{dp} and l_{dr} . For parametric analysis, only one

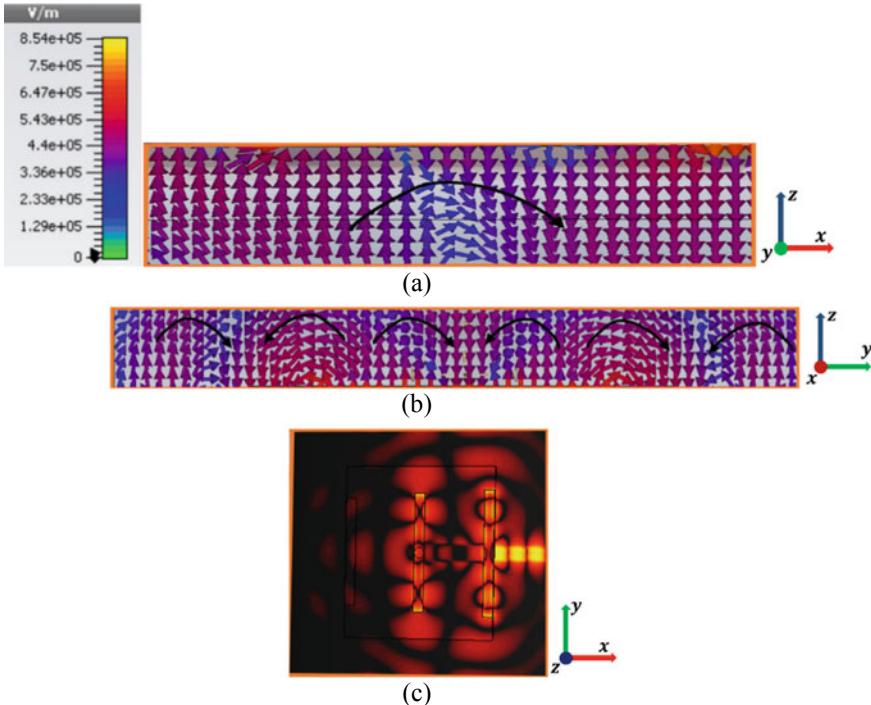


Fig. 4 Electric field distribution in **a** xz plane, **b** yz plane and **c** xy plane at 15.5 GHz

parameter has been changed at a time while retaining other parameters constant. The variables h_{s1} and l_{dr} are the responsible factor for providing impedance matching of the antenna while h_{s2} , l_r and l_{dp} have been used for the tuning of frequency response of the antenna. Figure 6a shows the S_{11} parameter at different dimensions of h_{s1} . As the height of substrate-1 increases from 0.6 to 1.4 mm, the return loss of the antenna decreases for all the dimensions of h_{s1} except at 1 mm. It signifies that impedance matching of the proposed antenna has been poorer excluding the chosen height for the proposed antenna. The height of substrate-1 ($h_{s1} = 1$ mm) is chosen for the desired dual-band frequency response. As the height of substrate-2 (h_{s2}) have been increased from 0.05 to 45 mm, the resonant frequency shifted towards higher range of frequency as shown in Fig. 6b. Shifting of frequency response confirms the tuning of resonant peak. The height of substrate-2 has been considered as 0.25 mm for the better analysis.

The length of driver helps to tune the frequency response. As the length of dipole (l_{dp}) varies from 27 to 35 mm with the difference of 2 mm, the resonating frequency of the peak response decreases but tuned at different frequencies as shown in Fig. 7a.

The optimized dipole length of the proposed antenna is obtained at 31 mm. The length of the reflector element should be greater than the dipole for better performance in Yagi-Uda antenna. Taking this into account, the parametric analysis has been done

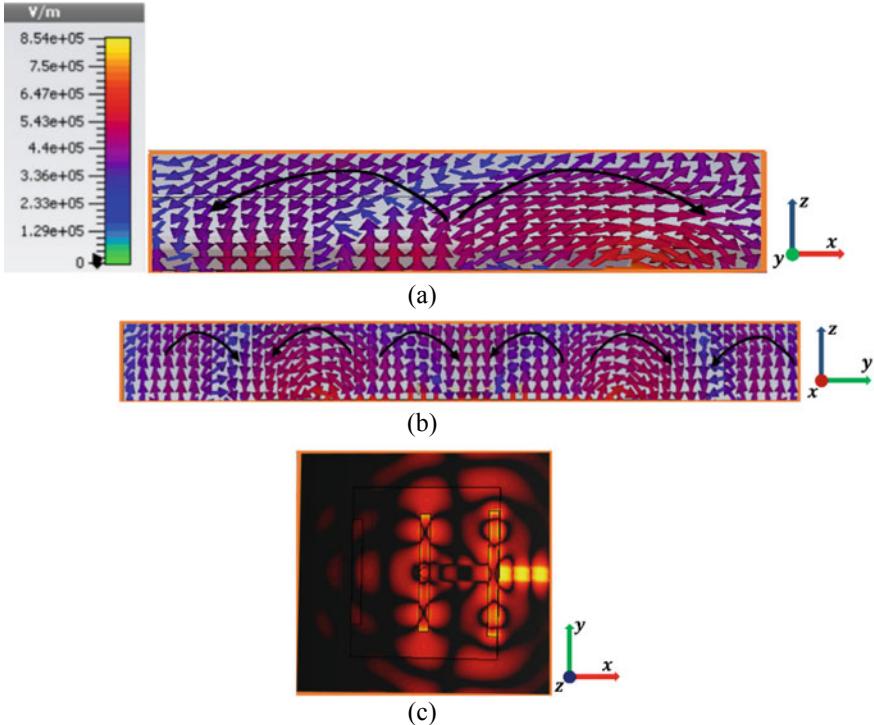


Fig. 5 Electric field distribution **a** xz plane, **b** yz plane and **c** xy plane at 16.1 GHz

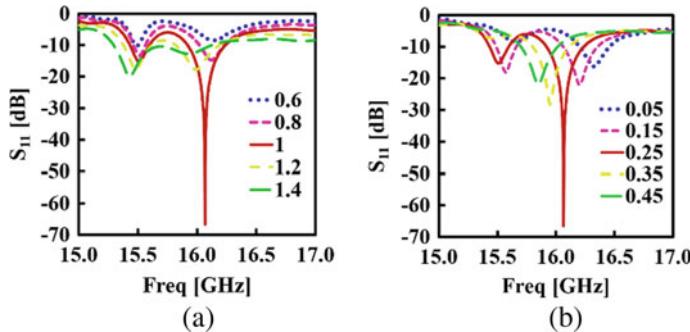


Fig. 6 Frequency response of S_{11} parameter by varying **a** h_{s1} and **b** h_{s2}

for reflector element as shown in Fig. 7b. The length of reflector is changed from 29 to 37 mm. Frequency response of the proposed antenna shifted towards upper frequency ranges and the impedance matching of antenna degrades for entire range of frequency except $l_r = 33$ mm. The higher band is shifted at $l_r > 17$ GHz. The variation of the director length has been studied to analyse the effect on the antenna

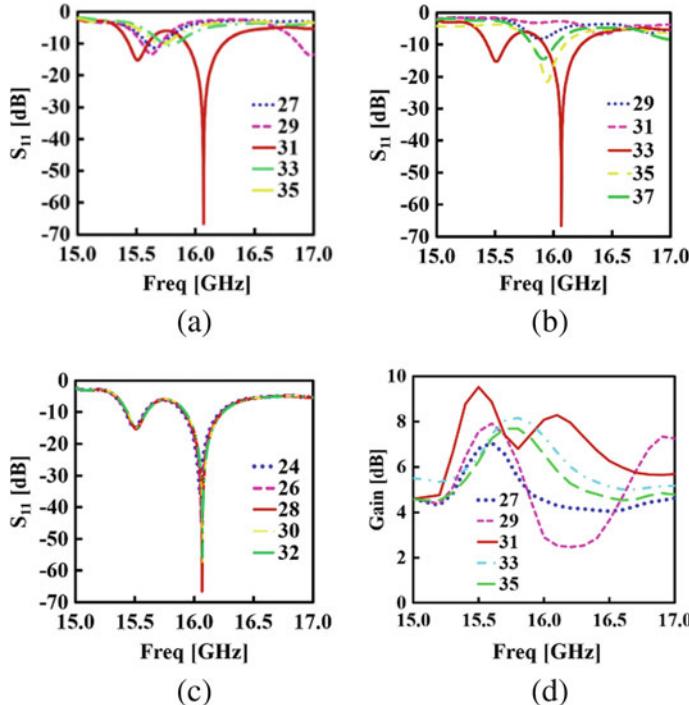


Fig. 7 Frequency response of S parameter by varying **a** l_{dp} , **b** l_r , **c** l_{dr} and **d** frequency response of the gain by varying l_{dp}

performance. Figure 7c shows the frequency response of return loss (S_{11}) at different values of l_{dr} . The resonant frequency of the antenna does not change with director length from 24 to 32 mm. The antenna provides good impedance matching at $l_{dr} \geq 28$ and $l_{dr} < 28$ mm. Furthermore, the variation of l_{dp} has been analysed on the frequency response of the antenna as shown in Fig. 7d. It has been observed that maximum gain has been obtained at the optimized dimension of the dipole in comparison with the other dimensions. Also, the maximum peak has been obtained for both the lower and upper passbands of the antenna.

5 Conclusion

A hybrid material-based Yagi-Uda antenna was proposed for Ku-band applications. The Yagi-Uda elements are made of different materials having different conductivities arranged in decreasing order. The proposed structure provides dual-band response with resonant frequencies at 15.5 and 16.1 GHz. The realized gain are 9.52 dBi and 8.1 dBi at lower and higher bands, respectively. The dipole of the

antenna is excited the higher-order TM_{16} and TM_{26} mode. The proposed antenna structure with the optimized dimensions may have promising utilization for Ku-band applications.

References

1. Wang C, Chen S, Yang Y, Hu F, Liu F, Wu J (2018) Literature review on wireless sensing-Wi-Fi signal-based recognition of human activities. *Tsinghua Sci Technol* 23:203–22. <https://doi.org/10.23919/TST.2018.8329114>
2. Chen M, Chen CC (2013) A compact dual-band GPS antenna design. *IEEE Antennas Wirel Propag Lett* 12:245–248. <https://doi.org/10.1109/LAWP.2013.2247972>
3. Zhang J, Wu W, Fang DG (2011) Single RF channel digital beamforming multibeam antenna array based on time sequence phase weighting. *IEEE Antennas Wirel Propag Lett* 10:514–516. <https://doi.org/10.1109/LAWP.2011.2157073>
4. Ito T, Kasami H (2015) External millimeter-wave antenna using spatial coupling for antenna in IC package. In: IEEE international symposium on antennas and propagation & USNC/URSI national radio science meeting 2015, pp 2033–2034. <https://doi.org/10.1109/APS.2015.7305406>
5. Lu J, Ireland D, Schlub R (2005) Dielectric embedded ESPAR (DE-ESPAR) antenna array for wireless communications. *IEEE Trans Antennas Propag* 53:2437–2443. <https://doi.org/10.1109/TAP.2005.852517>
6. Cho K, Hong S (2012) Design of a VHF/UHF/L-band low-power active antenna for mobile handsets. *IEEE Antennas Wirel Propag Lett* 11:45–48. <https://doi.org/10.1109/LAWP.2011.2181149>
7. Dai X, Wang Z, Liang C, Chen X, Wang Li, et al (2013) Multiband and dual-polarized omnidirectional antenna for 2G/3G/LTE application. *IEEE Antennas Wirel Propag Lett* 12:1492–1495. <https://doi.org/10.1109/LAWP.2013.2289743>
8. Todosiciuc A, Nicorici A, Condrea E, Warchulska J (2012) Electrical properties of lead telluride single crystals doped with Gd. In: Proceedings on international semiconductor conference CAS, vol 2, pp 269–72. <https://doi.org/10.1109/SMICND.2012.6400788>
9. Wu JW, Wang CJ, Jou CF (2009) Method of suppressing the side lobe of a tapered short leaky wave antenna. *IEEE Antennas Wirel Propag Lett* 8:1146–1149. <https://doi.org/10.1109/LAWP.2009.2034474>
10. Alhalabi RA, Rebeiz GM (2010) Differentially-fed millimeter-wave yagi-uda antennas with folded dipole feed. *IEEE Trans Antennas Propag* 58:966–969. <https://doi.org/10.1109/TAP.2009.2039320>
11. Zhai G (2015) Gain enhancement of printed log-periodic dipole array antenna using an elliptical patch. In: Proceedings on 2015 IEEE 4th Asia-Pacific conference antennas propagation, APCAP 2015, vol 62, pp 54–55. <https://doi.org/10.1109/APCAP.2015.7374268>
12. Basit MA, Wen G, Ping W (2016) Wide-band CPW-fed slot antenna with parasitic directors for end-fire radiation. *IET Microwaves Antennas Propag* 10:1734–1739. <https://doi.org/10.1049/iet-map.2016.0346>
13. Rodriguez-Ulibarri P, Bertuch T (2016) Microstrip-fed complementary Yagi-Uda antenna. *IET Microwaves Antennas Propag* 10:926–931. <https://doi.org/10.1049/iet-map.2015.0734>
14. Honma N, Seki T, Nishikawa K (2008) Compact planar four-sector antenna comprising microstrip Yagi-Uda arrays in a square configuration. *IEEE Antennas Wirel Propag Lett* 7:596–598. <https://doi.org/10.1109/LAWP.2008.2000874>
15. Perez-garrido C, Prieto M (2004) Modified aperture coupled microstrip antenna. *IEEE Trans Antennas Propag* 52:3397–3401

16. Thiel DV (1993) An experimental investigation behaviour of the staked antenna for surface electrical field measurement. In: Proceedings of IEEE antennas and propagation society international symposium, pp 1312–1315. <https://doi.org/10.1109/APS.1993.385435>
17. Hu W, Yin YZ, Yang X, Fei P (2013) Compact multiresonator-loaded planar antenna for multiband operation. *IEEE Trans Antennas Propag* 61:2838–2841. <https://doi.org/10.1109/TAP.2013.2242819>
18. Huang HC, Lu JC, Hsu P (2015) A compact dual-band printed Yagi-Uda antenna for GNSS and CMMB applications. *IEEE Trans Antennas Propag* 63:2342–2348. <https://doi.org/10.1109/TAP.2015.2406914>
19. Tan BK, Withington S, Yassin G (2016) A Compact microstrip-fed planar dual-dipole antenna for broadband applications. *IEEE Antennas Wirel Propag Lett* 15:593–596. <https://doi.org/10.1109/LAWP.2015.2462114>
20. Yang XS, Wang BZ, Wu W, Xiao S (2007) Yagi patch antenna with dual-band and pattern reconfigurable characteristics. *IEEE Antennas Wirel Propag Lett* 6:168–171. <https://doi.org/10.1109/LAWP.2007.895292>
21. Lim S, Iskander MF (2009) Design of a dual-band, compact yagi antenna over an EBG ground plane. *IEEE Antennas Wirel Propag Lett* 8:88–91. <https://doi.org/10.1109/LAWP.2008.2011502>
22. Xin Q, Zhang F, Sun B, Zou Y, Liu Q (2010) A novel dual-band Yagi-Uda antenna for wireless communications. In: 2010 9th international symposium on antennas, propagation EM theory, ISAPE 2010, pp 289–92. <https://doi.org/10.1109/ISAPE.2010.5696456>

Chapter 9

Broadband Electromagnetic Performance Analysis of Radome Structures Realized Using Hybrid Equilibrium Optimization Strategy



Anindya Midya Chowdhury , Ravi Yadav , Varun Chaudhary , and Ravi Panwar

1 Introduction

The radome is a crucial and necessary aspect of shielding the antenna from physical and environmental factors such as air, water, ice, wind, and birds [1]. It has been classified according to their application such as ground radomes which are utilized in the telecommunication sector, military surveillance or intelligence, and weather forecasts, etc. [2]. Airborne radomes are very widely used in aircraft, missile, satellite, and maritime STATCOM applications, etc. [2]. Radome should be physically robust in such a way that electrical parameters could not be affected [3]. The selection of material is a crucial task in the development of a radome structure. Researchers have explored several materials in this direction, which include polymers, ceramics, and resins, etc. [4, 5]. Many nanocomposite materials are also explored for their potential application in the development of radome walls [6]. Furthermore, Sunil et al. have determined the wall thickness for monolithic half-wave radomes using

A. Midya Chowdhury
Mechatronics, Indian Institute of Information Technology, Design and Manufacturing Jabalpur,
Madhya Pradesh, India
e-mail: 1815004@iiitdmj.ac.in

R. Yadav · V. Chaudhary · R. Panwar
Electronics and Communication Engineering, Indian Institute of Information Technology, Design
and Manufacturing Jabalpur, Madhya Pradesh, India
e-mail: rpanwar@iiitdmj.ac.in

R. Yadav
e-mail: 1822607@iiitdmj.ac.in

V. Chaudhary
e-mail: 1822610@iiitdmj.ac.in

their modified expression [7], but this type of radomes has very limited applications due to their narrow bandwidth. In an attempt to address this problem, the researchers have suggested various designs and techniques in the development of efficient radome structures [8–14].

Advanced EM structures for radome applications are reported to boost their transmitting characteristics, but manufacturing difficulty has not yet been resolved [15–17]. Although researchers have also explored the optimization techniques like genetic algorithm and particle swarm optimization approaches for reducing the complexity which still lacks in precision in optimizing radome parameters [18, 19]. So, it is still a very challenging task to design a cost effective, thin, and simple radome structure with the strongest transmission characteristics. So, here, the methodology is demarcated using a hybrid ABCD method enabled hybrid equilibrium optimization (EO) strategy to design efficient radome wall configurations. This theoretical model is suitable for the optimum design of distinct radome structures presented in the literature, and the best result is verified using the full-wave simulations.

2 Theoretical Background and Analytical Approach Utilized

Radome can be manufactured with various shapes, different wall structures, and material composition according to its application. Some well-known radome walls include monolithic, sandwiched, and layered structures [20]. A schematic of distinct types of radome wall structures is shown in Fig. 1. Radomes, made with single-layer

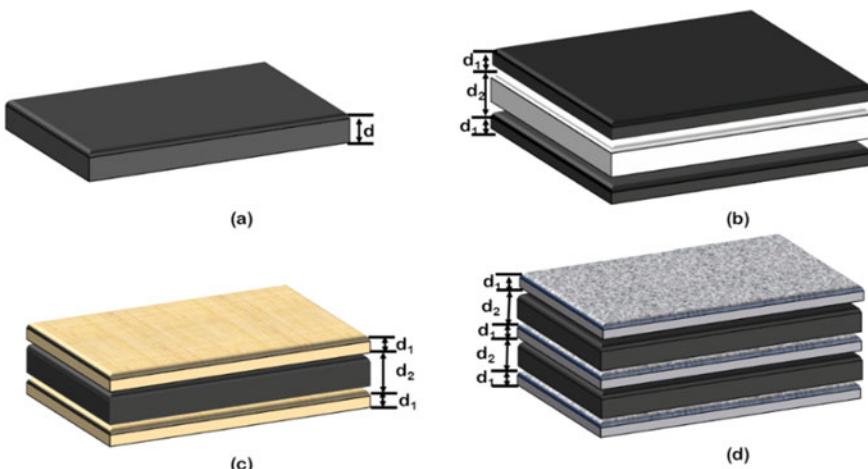


Fig. 1 Various types of radome configurations **a** monolithic radome, **b** A-sandwich radome, **c** B-sandwich radome, and **d** C-sandwich radome

dielectric material, are known as a monolithic radome as shown in Fig. 1a. The properties of various materials used to develop this type of radome are provided in [21]. A-sandwich is a three-layer radome structure, where the skin layer is thinner than the core layers, but the dielectric constant is more for the skin layer. Figure 1b represents a sketch of the A-sandwich radome wall configuration. B-sandwich radome is also a three-layered radome structure as shown in Fig. 1c. The dielectric constant of the skin layer is less than the core layer [2]. C-sandwich radome consists of two A-sandwich cascaded over each other as shown in Fig. 1d [20].

The analysis and optimization of these radomes are very crucial, due to the involvement of various geometrical design variables. However, there is a possibility to club multi-objective optimization techniques with traditional EM formulations to design optimal radome structures with good transmission characteristics. Multi-objective metaheuristic optimization algorithms have been widely explored by scientists for various EM applications [22, 23]. Consecutively, a novel algorithm named equilibrium optimization (EO) is utilized in the optimal design of efficient radome wall configuration. EO algorithm is inspired by a physics-based dynamic source and sink models used to estimate equilibrium states. The detailed optimization process has been discussed in [24]. A fitness function is used to get the best transmission as follows:

$$\text{Max_Cost.} T^{\text{TE/TM}} = \text{Maximize} \left[\sum_{\substack{\text{Freq}=\text{f max} \\ \text{Freq}=\text{f min}}} T^{\text{TE/TM}}(d, E, \theta) \right] \quad (1)$$

where T represents the total power transmission efficiency, d denotes the thickness of each layer, θ is the incident angle, and E is the radome parameter which includes permittivity and loss tangent of the radome materials. Further, ABCD-parameters are adapted to analyze the greater number of microwave networks [25].

3 Results and Discussion

The upper bound of dielectric constant, loss tangent, and thickness of skin and core layers is 10, 0.009, 8 mm, respectively. Similarly, the lower bound of the dielectric constant of skin layers and the core layer is 2 and 1, respectively. Loss tangent and thickness lower bound are the same for both skin, and core layers are 0.001 and 1 mm, respectively. The objective of the optimization process is to maximize the transmission, minimize the total thickness (d) of the radome structure, and maximize the bandwidth. The variables include dielectric constant (ϵ), loss tangent (δ), and thickness (d). The design variable with the population vector and the number of iterations is displayed in Table 1. It can be seen that the total number of iterations is 100, and the particle number is 10 for all the optimization. To calculate the power transmission efficiency of the proposed radome configurations, optimal geometrical

Table 1 Design variables with population vector and iteration

Variables	Population vector	No. of iterations
ε, δ, d	10	100

design variables are optimized by the aid of the ABCD method embedded EO technique. Based on the optimized results, a critical analysis of EM characteristics is carried out for all types of radome configurations.

3.1 Monolithic or Single-Layer Radome Structure

To achieve the best EM characteristics of a single-layer radome, the electrical properties of the material and corresponding layer thickness are optimized. The optimal geometrical design variables for monolithic radome configuration are layer thickness (d) = 5.79 mm, dielectric constant (ε) = 3, and loss tangent (δ) = 0.001. The power transmission efficiency is calculated at normal incidence using the ABCD method driven by EO. It can be observed that the power transmission efficiency is more than 75% over the broadband frequency. The minimum transmission is noticed around 7.0 GHz. The highest power transmission efficiency of more than 99% is found at about 15 GHz. The proposed radome structure has the best performance (i.e., the power transmission efficiency of more than 85%) in Ku-band. The passband (above 90%) can be seen around 13–18 GHz, and the bandwidth is calculated as 5.0 GHz. This narrow bandwidth is a significant limitation for such monolithic half-wave radome wall configurations. Therefore, further investigation is carried out on the EM characteristics of sandwiched structures.

3.2 A-sandwich Radome Structure

A-sandwich radome structure consists of three layers, i.e., one core layer and two thin skin layers. The optimized dielectric constant of skin (ε_1) and core (ε_2) layers is 3.0 and 2.1, respectively, as estimated by the EO strategy. Similarly, the optimal values of dielectric loss in the skin and core layers are determined as $\delta_1 = 0.001$ and $\delta_2 = 0.0035$, respectively. The optimized thickness of the skin layer (d_1) is 1.75 mm; on the other hand, core layer thickness (d_2) is 2.01 mm. The frequency-dependent power transmission efficiency of the proposed structure is obtained at a normal incidence angle for the perpendicular polarization over 1–18 GHz. The total thickness of the proposed radome structure is 5.51 mm.

The power transmission efficiency is more than 95% at the starting frequency and gradually decreases to 8.0 GHz. After 8.0 GHz, the power transmission efficiency increases and maximized around 15 GHz. All over the power transmission efficiency is more than 80% in the range of 1.0–18 GHz. Overall average transmission can be

obtained as 90%. The best value of the power transmission efficiency is observed in Ku-band. One can also see that the total thickness of A-sandwich is less than the proposed monolithic radome, and the power transmission efficiency is also improved in the case of A-sandwich. The passband in the case of A-sandwich is obtained from 11.8 to 18 GHz; as a result, the bandwidth also improved concerning the monolithic radome. So, it could be said that the objective of increasing bandwidth is getting fulfilled. But 6.2 GHz is a narrow bandwidth that is the reason to further study toward B-sandwich and C-sandwich radome structures.

3.3 *B-sandwich Radome Structure*

The variables ε_1, δ_1 , and d_1 are represented by the dielectric constant, dielectric loss, and thickness of the skin layer, respectively for the B-sandwich structure. Similarly, ε_2, δ_2 , and d_2 are parameters designated for the core layer. The optimization of these parameters is carried out using EO in the same manner as explained in the previous cases. The optimal values for B-sandwich radome are $\varepsilon_1 = 2$, $\delta_1 = 0.015$, $d_1 = 2.77$ mm and $\varepsilon_2 = 3$, $\delta_2 = 0.002$, $d_2 = 4.75$ mm. That means the total thickness of the proposed B-sandwich radome is 10.29 mm. More than 90% of the power transmission efficiency over the range of 9.0–18 GHz at zero incidence angle for perpendicular polarization is noticed for the proposed configuration as shown in Fig. 2a. It is found to possess a minimum efficiency of 77% at C-band. In this case, there is a considerable improvement in bandwidth that can be seen as the passband obtained across 7.5–18 GHz. There are some physical limitations due to less density of skin layers, and for further improvement in bandwidth, C-sandwich radome configuration is explored.

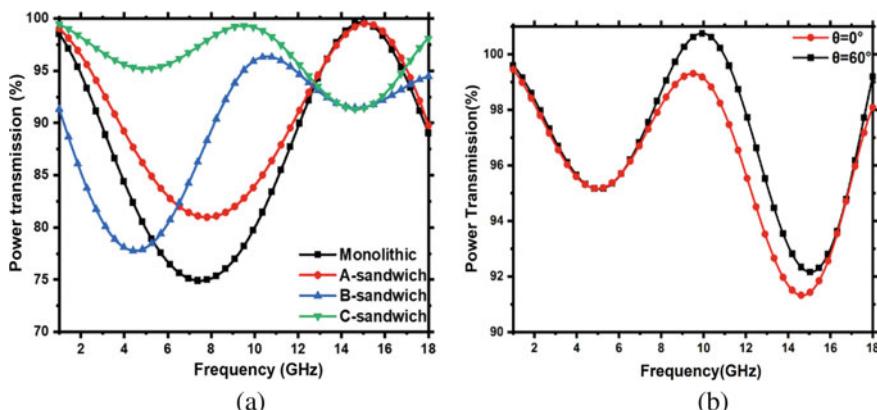


Fig. 2 Power transmission efficiency **a** of various types of radome configurations at a normal incidence angle over 1–18 GHz for perpendicular polarization, **b** of optimal radome at normal and high incidence angle over 1–18 GHz for parallel polarization

3.4 C-sandwich Radome Structure

The optimized dielectric constant, dielectric loss, and thickness of the first, third, and fifth layers are 3, 0.001 and 1.001 mm, respectively. Similarly, in the case of the second and fourth layers, the dielectric constant is 1.08, dielectric loss is 0.005, and the layer thickness is 3.37 mm. One can easily calculate the total thickness of the proposed C-sandwich radome structure is 9.7 mm. Figure 2a depicts the power transmission efficiency of the proposed C-sandwich radome structure. It can be observed that power transmission efficiency is outstanding (i.e., more than 90%) in all over frequency 1 to 18 GHz, and the maximum and minimum transmission can be observed at 10 GHz and 15 GHz, respectively. The passband is obtained in the case of C-sandwich radome across the broadband frequency.

The power transmission efficiency of the C-sandwich radome at a normal incidence angle and high incidence angle 60° for parallel polarization is shown in Fig. 2b. It can be observed that transmission is above 90% throughout the frequency and best at X-band (more than 95%) for both incidence angles. The maximum and minimum transmission is seen across 10 GHz and 15 GHz, respectively. So, one can say that these optimized values are very much valid, and the model is very suitable for practical applications. In the case of parallel polarization, the bandwidth of C-sandwich is obtained across the broadband frequency as the total transmission is more than 90%.

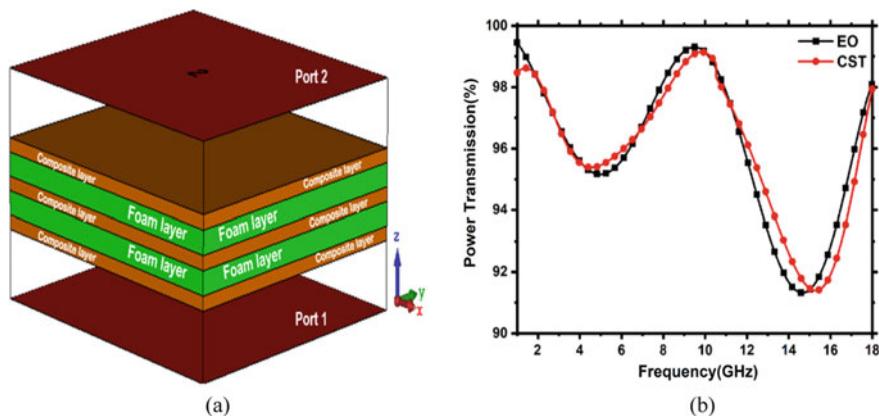
4 Validation of the Adopted Approach

A comparative study has been done of optimization results of the power transmission efficiency for proposed radome configurations in Table 2. The bandwidth (over 90% transmission) of monolithic, A, B, and C-sandwich, is 5 GHz, 6.2 GHz, 10.5 GHz, and 17 GHz, respectively. The proposed monolithic and A-sandwich radomes are thinner but the main focus of this work was to get the maximum transmission and wide bandwidth. So, it is easily perceived that the performance of the C-sandwich radome is the best as compared to other radome configurations. So, an optimal radome (i.e., C-sandwiched) designed using EO is validated using full-wave simulation.

A 3D model (see Fig. 3a) is created according to the optimized values obtained from EO and compared with the EO results, as shown in Fig. 3b. In the model, open add space boundary condition in both sides of the z-axis has been applied to calculate far-field. In the case of the x-axis, the tangential electric field is kept zero, and the y-axis tangential magnetic field is also kept zero. Two waveguide ports are used in the +Z and -Z-axis for calculating the S-parameter of the proposed C-sandwich radome. As one can see easily that the power transmission efficiency curve in both cases EO and full-wave simulation are matching well. So, the optimization technique is proved to be effectively working and well executed. Figure 3b provides that the maximum power transmission efficiency for both techniques (EO and CST) is more

Table 2 Comparative analysis of optimization results of power transmission efficiency for proposed radome configurations

Radome configurations	Optimal geometrical design variables	Total radome layer thickness (mm)	Maximum power transmission efficiency (%) at frequency (GHz)	Transmission bandwidth (GHz)
Monolithic	$\epsilon = 3, d = 5.79, \delta = 0.001$	5.79	99.5 at 15	5 (13–18)
A-sandwich	$\epsilon_1 = 3, d_1 = 1.75, \delta_1 = 0.001, \epsilon_2 = 2.1, d_2 = 2.05, \delta_2 = 0.0035$	5.51	99.6 at 14.6	6.2 (11.8–18)
B-sandwich	$\epsilon_1 = 2, d_1 = 2.77, \delta_1 = 0.015, \epsilon_2 = 3, d_2 = 4.75, \delta_2 = 0.002$	10.29	97.5 at 10	10.5 (7.5–18)
C-sandwich	$\epsilon_1 = 3, d_1 = 1.001, \delta_1 = 0.001, \epsilon_2 = 1.08, d_2 = 3.37, \delta_2 = 0.005$	9.743	99.5 at 9.5	17 (1–18)

**Fig. 3** **a** Three-dimensional model of C-sandwich structure in CST, **b** comparison of EO and CST results for optimal radome at normal incidence angle in the range of 1–18 GHz for perpendicular polarization

than 99% at the same frequency of 9.5 GHz. On the other hand, the minimum power transmission efficiency is 91.3% in 14.9 GHz using the EO technique and 91.3% at 15.1 GHz for a full-wave simulation technique.

5 Conclusion

Investigation of the power transmission efficiencies is carried out for radome wall structures using the ABCD method enabled EO strategy. It is observed that the cost function used in the EO strategy is a multi-objective function as it continuously coordinating two or more objectives that are subject to certain limitations. One can easily find that C-sandwich has the best EM characteristics as compared to other proposed radomes. The results show the enormous potential of the EO approach in the optimal design of various EM structures.

References

1. Bruks D (2007) Antenna engineering handbook. Raytheon Company
2. Kozakoff D (2010) Analysis of radome-enclosed antennas, 2nd edn. Artech House, Norwood
3. Kraus JD, Marhefka RJ (2003) Antennas for all applications, 2nd edn. McGraw-Hill, New York
4. Pilato LA, Michno MJ (1994) Advanced composite materials. Springer
5. Ren H (2017) Reliability-based aircraft maintenance optimization and applications. Elsevier, Shanghai, pp 1–18
6. Saxena N (2010) Study of LiTiMg-ferrite radome for the application of satellite communication. *J Magn Magn Mater* 322:2641–2646
7. Sunil S (2001) A modified expression for determining the wall thickness of monolithic half-wave radomes. *Microw Opt Technol Lett* 30(5):2000–2002
8. Nair R (2012) Novel inhomogeneous planar layer radome design for airborne applications. *IEEE Trans Antennas Prop Lett* 11:854–856
9. Panwar R (2018) Performance and non-destructive evaluation methods of airborne and stealth structures. *Meas Sci Technol* 29:1–29
10. Nair R (2013) Broadband EM performance characteristics of single square loop FSS embedded monolithic radome. *Int J Antennas Prop* 2013:1–8
11. Liu Q (2009) Optimal design for ceramic radomes with A-sandwich structure. *Adv Synth Process Technol Mater* 66:29–32
12. Zhou L (2016) Dual-band A-sandwich radome design for airborne applications. *IEEE Antennas Wirel Propag Lett* 15:218–221
13. Nair R (2013) Application of metallic strip gratings for enhancement of electromagnetic performance of A-sandwich radome. *Defense Sci J* 63(5):508–514
14. Nair R (2007) Novel A-sandwich radome design for airborne applications. *IET Electron Lett* 43(15):787–788
15. Panwar R (2017) Progress in frequency selective surface-based smart electromagnetic structures: a critical review. *Aerospace Sci Technol* 66:216–234
16. Nair R (2009) EM performance analysis of double square loop FSS embedded C-sandwich radome. *Computational Electromagnetics Lab.*, pp 7–9
17. Panwar R (2015) Fractal frequency-selective surface embedded thin broadband microwave absorber coatings using heterogeneous composites. *IEEE Trans Microw Theor Tech* 63(8):2438–2448
18. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
19. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th international symposium on micro machine and human science, pp 39–43, IEEE, Japan

20. Yazeen P (2017) Electromagnetic performance analysis of graded dielectric inhomogeneous streamlined airborne radome. *IEEE Trans Antennas Prop* 65(5):2718–2723
21. Mani G (1994) Microwave materials. In: VRK Murthy et al. Springer, Berlin
22. Panwar R (2018) Experimental demonstration of novel hybrid microwave absorbing coatings using particle-size-controlled hard-soft ferrite. *IEEE Trans Magn* 54(11):1–5
23. Yadav R (2018) Extended Jaya's algorithm for optimal design of broadband layered microwave absorbing structures. *IEEE Magn Lett* 9:1–5
24. Faramarzi A (2020) Knowledge-based systems equilibrium optimizer: a novel optimization algorithm. *Knowl Based Syst* 191:1–21
25. Chen F (2010) Electromagnetic optimal design and preparation of broadband ceramic radome material with graded porous structure. *Progress Electromagnet Res* 105:445–461

Chapter 10

Design, Optimization, and Critical Analysis of Metamaterial Superstrate-Coupled High-Gain Microstrip Patch Antenna



Atipriya Sharma , Ravi Panwar , and Rajesh Khanna

1 Introduction

Microstrip patch antennas (MPAs) are physically vigorous and having non-complex planar properties, which leads to positively combine with microwave circuits [1]. Despite that, MPAs have a few shortcomings such as narrow bandwidth, radiation efficiency, and low gain. These shortcomings limit the usage of MPAs for multiple practical applications. To use the antenna for different applications, required to have high gain and a wide bandwidth [2]. Array antennas are also in use to intensify such shortcomings of MPAs, but their applications are limited due to high mutual coupling and volume of antenna structure [3]. Currently, artificial structures have become more popular to improve the performance characteristics of an antenna [4].

MM is an artificial composite structure, which shows remarkable property like a negative index of refraction, which is not found in natural materials [5]. MM and electromagnetic bandgap (EBG) absorbers have been well reported in their different potential application [6–9]. Metasurfaces of double-negative media [10], chiral materials [11], and polarization conversion metamaterials (PCM) [12, 13] got more attention from the researchers due to their extraordinary characteristics. The chiral medias are also reported as a type of PCM [11]. In [12], the authors demonstrated PCM, which is suitable for terahertz applications. Metasurfaces shape can vary from a

A. Sharma · R. Khanna

Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

e-mail: atipriya.sharma@thapar.edu

R. Khanna

e-mail: rkhanna@thapar.edu

R. Panwar

Discipline of Electronics and Communication Engineering, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, Madhya Pradesh 482005, India

single periodic pattern in the double periodic patterns with distinct shapes and sizes [14, 15]. In [15], the authors have designed a metasurface, in which the top layer was made up of a combined split-ring resonator (CSRR) metallic structure. Several pioneer research studies on high-gain antennas have been carried out by researchers worldwide, but the simultaneous achievement of low thickness and the high gain antenna is a difficult task. Therefore, it is of great importance to identify an efficient approach to resolve the aforementioned issue. In this direction, an effort has been made to couple a new metamaterial superstrate to microstrip patch antenna for the gain improvement. In this paper, a primary antenna is simulated, and then, MM surface was designed and used with the primary antenna as a superstrate to enhance the gain. The paper is arranged as follows. Section 2 illustrates the design of the proposed antenna. Section 3 devotes itself to the simulated results of the proposed antenna. Finally, Sect. 4 contains the concluding part of this article.

2 Design of Hybrid Metamaterial Superstrate-Coupled Microstrip Patch Antenna

MPA with a 2.0 mm thick FR4 substrate (dielectric loss tangent, $\tan \delta = 0.025$ and relative permittivity, $\epsilon_r = 4.3$) is designated as a primary antenna. The primary antenna configuration has been simulated using Computer Simulation Technology (CST), Darmstadt, Germany microwave studio software, in the time domain solver. To excite the antenna, waveguide port and coaxial feed are used. The optimal values of antenna geometrical design variables are described in Table 1.

Two FSS unit cells, namely FSS#1 and FSS#2, have been designed and simulated. The thickness of the substrate is 0.8 mm. Further, both unit cells are used in the development of MM superstrate for the design of MM loaded antenna. FSS#1 consists of a square geometry surrounded by a circle as elucidated in Fig. 1a, while FSS#2 comprises of a circle surrounded by a square as illustrated in Fig. 1b. FSS-based MM unit cell has been simulated in the frequency domain solver, using CST microwave studio software by employing periodic boundary conditions. MM consists of a copper layer printed on a dielectric substrate FR4 with a thickness “T1” of 0.2 mm. The structure of the superstrate layer is shown in Fig. 1c. Table 2 represents the design variables of the MM unit cell.

A higher iterated hybrid MM geometry has been loaded on the top of the primary patch antenna with an air gap of 7.0 mm in between for the design of the MM superstrate antenna. The three-dimensional view of the structure is illustrated in

Table 1 Design variables of the patch antenna

Parameter	Value (mm)	Parameter	Value (mm)
P	80.0	A	7.0
W	14.2	T	2.0
L	14.2	–	–

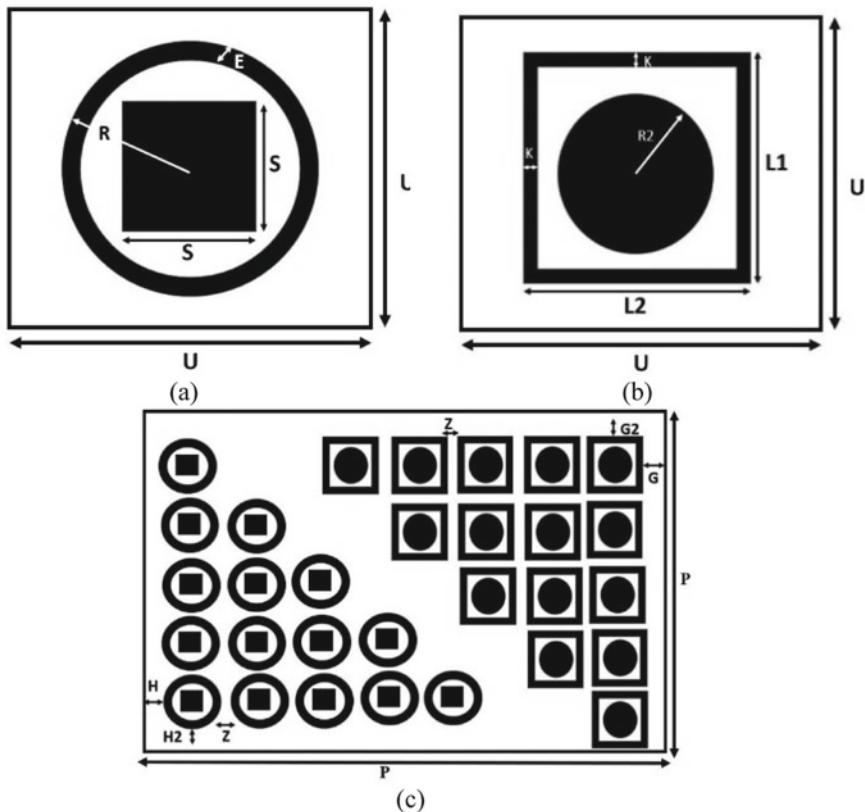


Fig. 1 FSS-based MM configurations utilized for the design of MM inspired antenna **a** FSS#1, **b** FSS# 2, and **c** hybrid MM layer

Table 2 Design variables of MM unit cell

Parameter	Value (mm)	Parameter	Value (mm)
R	4.8	L_1	9.60
E	0.8	L_2	9.60
R_2	3.5	U	10.0
S	4.2	K	0.80
H	1.2	G	16.2
$H2$	1.2	G_2	1.20
Z	10.6	T_1	0.2
P	80.0	—	—

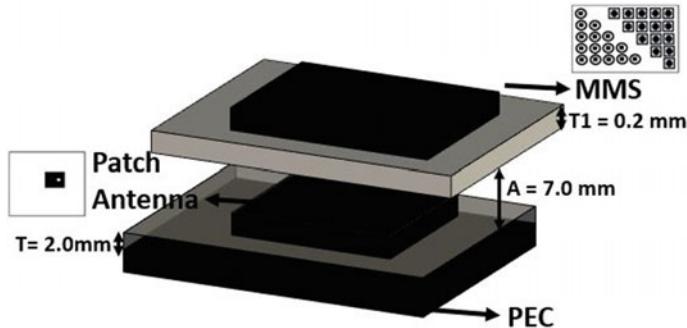


Fig. 2 Three-dimensional (3D) view of the proposed structure

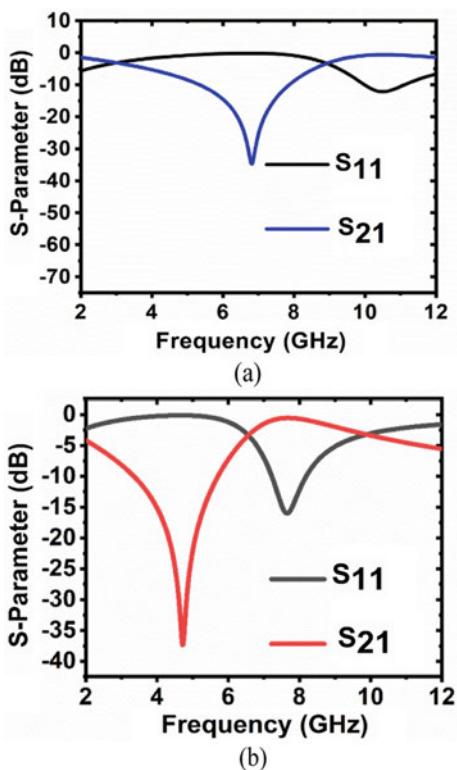
Fig. 2. The overall dimension of the structure, including primary MPA and impinged hybrid MM layer, was 80×80 mm. Finally, the MM superstrate antenna (composite of the primary patch antenna and hybrid MM) was simulated to estimate the reflection coefficient (S_{11}) characteristics and the gain response of the structure.

3 Critical Analysis and Simulated Results

The frequency-dependent scattering parameters (S-parameters) for the unit cell of FSS#1 and FSS#2 are illustrated in Fig. 3a, b. It is analyzed that the bandpass response of FSS#1 appears at 10.44 GHz, whereas the stopband response is observed at 6.82 GHz as depicted in Fig. 3a. Similarly, bandpass resonance of FSS#2 is observed at 7.69 GHz, and bandstop resonance of FSS#2 is observed at 4.73 GHz as shown in Fig. 3b, whereas for bandpass response of FSS#2 pass the frequencies from 4.3 GHz to 6.7 GHz. Figure 4a, b illustrates the S-parameters in terms of the co- and cross-polarization of FSS#1 and FSS#2 structures backed by a perfect electric conductor (PEC), respectively. The proposed FSS#1 has a minimum S_{11} of -15.9 dB at 5.5 GHz, whereas FSS#2 covers -10 dB absorption bandwidth from 9.5 GHz to 11.5 GHz. FSS#2 possesses S_{11} of -35.0 dB at 10.32 GHz, as illustrated in Fig. 4a, b, respectively. The normalized impedance graphs of both unit cells are shown in Figs. 5a, b. It has been perceived from Fig. 5a that at the resonant frequency, the value of the real part of the impedance is approximately equal to unity. A similar experience has been observed in Fig. 5b.

Further, FSS unit cells are utilized in the development of MM loaded antenna. The S_{11} of the primary and proposed antenna is illustrated in Fig. 6a. The S_{11} bandwidth of both antennas is the same (i.e., 800 MHz), whereas the resonating frequency is 4.55 GHz and 4.6 GHz for primary and proposed antennas, respectively. Comparison between primary and proposed antenna is illustrated in Table 3. Figure 6b illustrates the maximum gain over the frequency graph of primary and proposed antennas in the range of 2–12 GHz, which shows that the maximum gain attained by the primary

Fig. 3 Frequency-dependent S-parameters for **a** FSS#1 and **b** FSS#2



antenna is equal to 6.6 dB, where the maximum gain of the proposed antenna is equal to 7.2 dB, which proves the gain enhancement by the proposed structure.

4 Conclusion

A high gain has been achieved for an MSA by the application of a hybrid MM superstrate, which is comprised of a unique combination of FSSs. The two different FSS unit cells, namely FSS#1 and FSS#2, are designed and simulated to form the hybrid MM layer. A maximum gain value of 7.2 dB is observed for the proposed antenna. In this manner, low thickness and high gain both are simultaneously attained with the help of the proposed approach.

Fig. 4 Frequency-dependent S -parameters in terms of co- and cross-polarization of **a** FSS#1 and **b** FSS#2

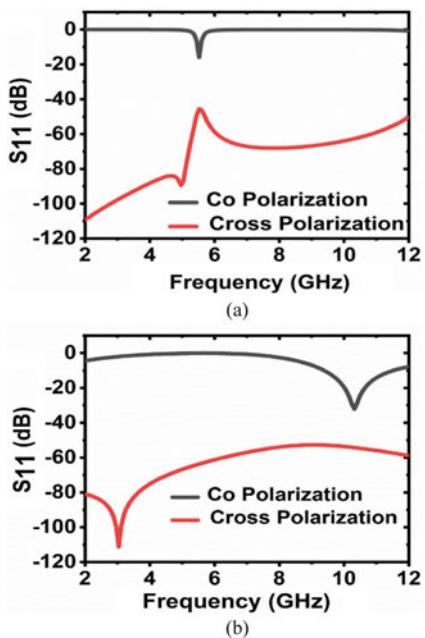


Fig. 5 Normalized impedance in terms of real and imaginary of **a** FSS#1 and **b** FSS#2

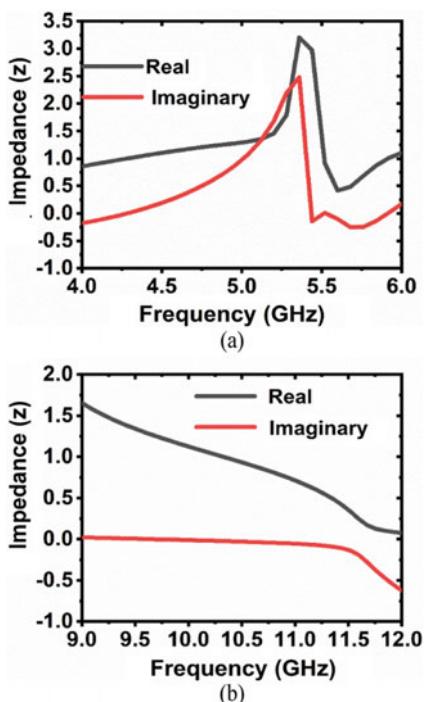


Fig. 6 Characteristics of MM-coupled antenna **a** S_{11} and **b** gain

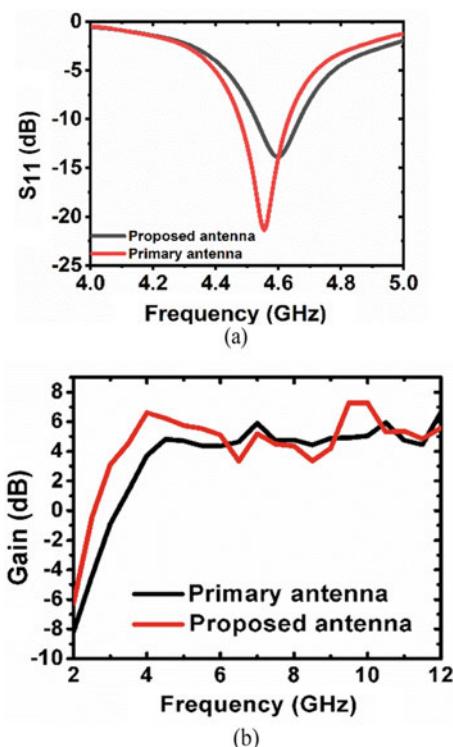


Table 3 Comparison between primary and proposed antenna

Antenna	S_{11} bandwidth (MHz)	RCS bandwidth (GHz)	Maximum gain (dB)
Primary	800	1.7	6.6
Proposed	800	8.1	7.2

References

- Ullah MH, Islam MT, Mandeep JS, Misran N, Nikabdullah N (2012) A compact wideband antenna on dielectric material substrate for K band. *Elektronika Ir Elektrotehnika* 123:75–78
- Durán-Sindreu M, Naqui J, Paredes F, Bonache J, Martín F (2012) Electrically small resonators for planar metamaterial, microwave circuit and antenna design: a comparative analysis. *Appl Sci* 2:375–395
- Nadeem I, Choi DY (2018) Study on mutual coupling reduction technique for MIMO antennas. *IEEE Access* 7:563–586
- Panwar R, Lee JR (2017) Progress in frequency selective surface-based smart electromagnetic structures: a critical review. *Aerospace Sci Technol* 66:216–234
- Engheta N, Ziolkowski RW (2006) Electromagnetic metamaterials: physics and engineering explorations. Wiley, New York
- Li YQ, Zhang YH, Fu Y, Yuan N (2008) RCS reduction of ridged waveguide slot antenna array using EBG radar absorbing material. *IEEE Antennas Wirel Propag Lett* 7:473–476

7. Sharma, A, Panwar, R, Khanna R (2019) Experimental validation of a frequency-selective-surface-loaded hybrid metamaterial absorber with wide bandwidth. *IEEE Mag Lett* 10
8. Panwar R, Lee JR (2019) Recent advances in thin and broadband layered microwave absorbing and shielding structures for commercial and defense applications. *Funct Compos Struct* 1:032001
9. Panwar R, Lee JR (2018) Performance and non-destructive evaluation methods of airborne radome and stealth structures. *Measur Sci Technol* 6:062001
10. Shelby RA, Smith DR, Schultz S (2001) Experimental verification of a negative index of refraction. *Science*: 77–79
11. Lindell I, Sihvola A, Tretyakov S, Viitanen A (1994) Electromagnetic waves in chiral and bi-isotropic media. Artech House, Boston
12. Chen H, Wang J, Ma H, Qu S, Xu Z, Zhang A, Yan M, Li Y (2014) Ultra-wideband polarization conversion metasurfaces based on multiple plasmon resonances. *J Appl Phys* 115:154504
13. Feng M, Wang J, Ma H, Mo W, Ye H, Qu S (2013) Broadband polarization rotator based on multi-order plasmon resonances and high impedance surfaces. *J Appl Phys* 114:074508
14. Doumanis E, Goussetis G, Papageorgiou G, Fusco V, Cahill R, Linton D (2013) Design of engineered reflectors for radar cross section modification. *IEEE Trans Antennas Propag* 61:232–239
15. Li Y, Zhang J, Qu J, Wang J, Chen H, Xu Z, Zhang A (2014) Wideband radar cross section reduction using two-dimensional phase gradient metasurfaces. *Appl Phys Lett* 104:1110–1113

Chapter 11

Cheating-Tolerant and Threshold-Based Secure Information Exchange Among Propinquity of Adversaries



Anindya Kumar Biswas and Mou Dasgupta

1 Introduction

Attacks, embezzlement and colossal misapplication of sensitive data are heightening expeditiously and are becoming a daily hindrance of both individuals and organizations. Importance of safeguarding sensitive information is, therefore, on the rise mainly due to substantial increase in security threats with enhanced sophistication and complexity of attacks over the insecure network [1]. These attacks flourish as they are not bound by geographical limitations and are exceedingly difficult to mitigate, which generally involves gargantuan time, effort and cost [2, 3], resulting in unproductive activity on the victim's part and various other potential damages [4]. So, defense strategies like the domain of secret sharing schemes (SSSs) [5] can provide cryptographic techniques for safeguarding private/sensitive information over the insecure/open network channel. The schemes deployed should be of good competence so that outsiders/cybercriminals are not able to obtain the real data conveyed.

This paper is structured as follows: Sect. 1 includes introductory background, Sect. 2 provides the reviews of related works, Sect. 3 describes Shamir's SSS, and Sect. 4 describes the proposed SSSs. Finally, Sect. 5 provides conclusions.

2 Related Works

The popularity of SSS is on the rise. Diffie–Hellman protocol [6] is of the earlier technique that enables secure key negotiation over the open network. This key can later

A. K. Biswas () · M. Dasgupta

Department of Computer Application, National Institute of Technology, Raipur, India

be used for encryption/decryption purposes. One approach in SSS is the threshold-based cryptography which was independently put forward by Shamir [7] and aids in decentralized information storage. The scheme, although popular, has drawbacks. Deceiving of honest participants by forging of shares is an inherent problem and was pointed out by Tompa [8]. By taking in $m - t$ extra shares, a method [9] of cheating detection was given by Harn-Lin. Here, members external to the group are included in the current group, thereby resulting in external participants becoming part of the group, which is unacceptable. The work [10] showed flaws in existing scheme and proposed ways to improve it including the possibility of cheater identification. By implementation of XOR operation, a procedure [11] of secret sharing (SS) was given. SSS based on hierarchy [12], where all the related participants are organized in some hierarchy and secret, can be recovered only when lower-level participants receive sufficient shares from higher levels. Because of distributed nature, future of fog and cloud-based computing are not secure [13] and work is done to bring security in its applications. Using two polynomials, a way out [14] was provided to achieve threshold secrecy with reliable cheating detection.

This paper provides two SSSs that can detect shares' cheating made by dishonest participants and/or dishonesties of a dealer (if present). In first scheme, a subset of Z_p is taken as *subset secret* such that a dealer forms a polynomial using an element of the set as the constant term, and if a secret as an element of the set is reconstructed, then no cheating is done probably. In other scheme, two related SSSs are proposed and they are: a dealer-based (t, t) SS for $t \leq n$ that can eliminate shares' cheating and a KGC-based (t, n) SS that eliminates shares' cheating and avoids dealer's misconducts as KGC replaces the dealer.

3 Shamir's (t, n) Threshold Secret Sharing Scheme

A short description of Shamir's SSS is given, where the minimum t participants, out of n participants, are necessary to collectively negotiate a secret value, and the negotiation certainly fails even if $t - 1$ participants are involved. The scheme is intelligible, conducive and popular and consists of two phases as provided below:

- (1) *Share Generation and Distribution Phase:* A trusted D takes a large set $Z_p = \{0, 1, 2, \dots, p - 1\}$, produces a $t - 1$ degree random polynomial $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$ with coefficients $a_i \in Z_p^*$ and sets a_0 as the secret to be negotiated. D computes necessary shares as $(i, f(i))$ and transmits to participant i for $i = 1, 2, 3, \dots, n$ secretly.
- (2) *Reconstruction Phase:* t or more participants exchange their shares and use Lagrange polynomial interpolation individually to reconstruct $f(x)$ and get secret $f(0) = a_0$, where $f(x) = \sum_{i=1}^t f(i) \prod_{j=1, j \neq i}^t \frac{x-j}{i-j}$.

The scheme provides information-theoretic security, because (i) reconstruction of the covert value is possible only when t or more shares are attainable and (ii) the same is impossible when $t - 1$ or lesser shares are known/combined. The scheme is not secure when one or more fake shares are exchanged during secret reconstruction phase [8].

4 Proposed Cheating-Tolerant (t, n) SS Schemes

Here, two SSSs detect shares' cheating and/or misconduct of dealer so that the honest participants are not exploited by adversaries, and they are simple and efficient.

4.1 A Cheating-Tolerant SSS Based on Subset Secret

In SSSs, users' shares cheating cannot be avoided completely, i.e., if ε is the cheating probability, then $\varepsilon \neq 0$. We propose a simpler scheme that can reduce cheating probability with varying probabilities. Consider a proper subset $S_d \subset Z_p$ such that $1 < d < p$ and $S_d = \{0, 1, 2, \dots, d - 1\}$, called *subset secret* and a secret $s = f(0) \in S_d$. The proposed (t, n) threshold SSS has three phases as given below:

- (1) *Secret Share Generation and Distribution Phase*: D selects d , sets a secret $s \in S_d$ and chooses a random $(t - 1)$ -degree polynomial $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$, where $a_0 = s$ and $a_j \in Z_p$ for $j = 1, 2, 3, \dots, t - 1$. D generates n shares $(i, f(i))$ and distributes to participant i secretly, where $i = 1, 2, \dots, n$.
- (2) *Secret Reconstruction Phase*: Any t or more participants exchange their shares, reconstruct the polynomial $f(x)$ individually using Lagrange interpolation and get the common secret $s = f(0)$, provided no share cheating is occurred.
- (3) *Verification Phase*: After T time, D declares d publicly, and all participants verify the secret using $s \in S_d$? If yes, no share cheating is occurred; otherwise, share(s) cheating has occurred and terminates the session.

Here, the cheating probability depends on d selected by D , if $d = p/4$, $|S_d| = \frac{Z_p}{4}$ and $\varepsilon = 0.25$. Since p is large, ε would reduce significantly if d/p is very small. A flow diagram (Fig. 1) of the proposed SSS is given below.

4.2 Two Cheating-Free SS Schemes Based on Dealer and KGC

The design of cheating-free SSS is difficult, and in this section, we propose a cheating-free (t, t) threshold SSS, where $t \leq n$ and the shares are accompanied by D 's signature.

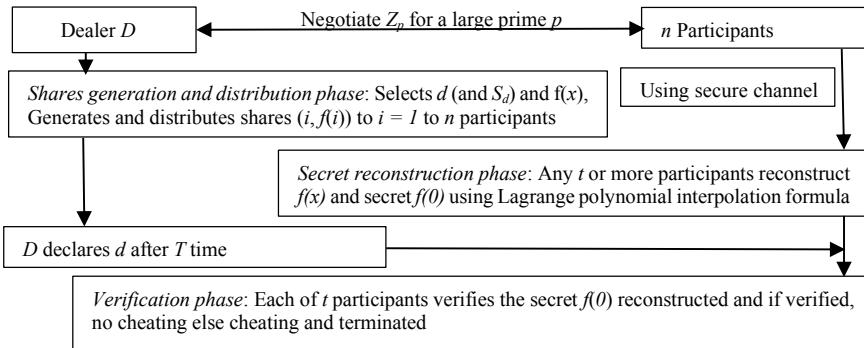


Fig. 1 Flow diagram of the proposed SSS

Since it is not a generalized (t, n) SSS, a KGC-based technique is also proposed so that different (t, t) schemes with varying t ($2 \leq t \leq n$) can be aimed. They are described now.

4.2.1 Proposed (t, t) SSS Using a Dealer

We know that in a (t, n) scheme, any t ($1 < t \leq n$) or more participants can reconstruct the secret; however, at particular session, $n - t$ participants (if $t < n$) remain idle and as a result, it's wastage to generate shares and transmitting securely to them, and insecure as well due to loss, theft, misplacement and misuse. Thus, the design of a (t, t) SSS is more justified than (t, n) one, where t varies from 2 to n participants. The proposed scheme is given below:

Share Generation and Distribution Phase

- (1) As before, D selects a large prime number p , considers the set $Z_p = \{0, 1, 2, 3, \dots, p - 1\}$ and creates a $(t - 1)$ degree polynomial $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$ with a $a_0 \in Z_p$ such that all t participants are combined together to reconstruct the secret.
- (2) On reporting by t participants, D generates shares $(i_j, f(i_j))$ for $i_1, i_2, \dots, i_j, \dots, i_t$ participants and distributes to all the t participants secretly. For security protection of shares, D performs the following:
 - (i) Computes $Y = \text{Hash}(f(i_1)f(i_2)\dots f(i_t))$
 - (ii) Computes signature as $Z = \text{SIG}(K_{PR}, Y)$, where SIG is signature function applied on Y with his/her private key K_{PR} .
 - (iii) Broadcasts the message $W = (\text{CERT}_D, Z)$ to all participants publicly, where CERT_D is the digital certificate of D (for authenticated public key to verify the computed Y and decrypted Y from Z).

Secret Reconstruction Phase

- (1) All t participants exchange their shares among themselves, use Lagrange interpolation equation individually and reconstruct the polynomial $f(x)$ and the secret $a_0 = f(0)$.

Cheating Detection Phase

- (1) After exchange of shares, each participant gets t shares and computes $Y' = \text{hash}(f'(i_1) \| f'(i_2) \| \dots \| f'(i_t))$, where $f'(i_j)$ is the share received from participant i_j .
- (2) From W , each participant decrypts Z and gets $Y = \text{DEC}(K_{PU}, Z)$, where K_{PU} is the public key of D obtained from CERT_D and DEC is decryption procedure.
- (3) Each participant now compares $Y' = Y$? (obtained from (1) and (2) above). If they are equal, no share cheating is occurred and correct secret is reconstructed, else share cheating is occurred and the session is terminated.

Note that it is a complete cheating-free (t, t) threshold SSS; however, it lacks the generality of (t, n) one. In the next subsection, a generalization using KGC is proposed.

4.2.2 Proposed (t, n) SSS Using KGC

The dealer-based (t, t) SSS proposed in Sect. 4.2.1 is made cheating-free by generating a dealer's signature on the fixed t shares; however, the same is not possible if t is unknown to D and varied as in general (t, n) SSS. Thus, in order to apply the same digital signature construct and to make cheating-free SSS, we replace dealer by KGC so that multiple groups with varying size (value of t) are deployable iteratively. By deploying KGC, users can generate a part of the key and the other part by KGC itself [15]. A common public encryption key can also be negotiated among a group using KGC [16]. In addition, KGC is cryptographically authentic (not a dealer) and as a result, the misbehaved activities (if any) of dealer can be avoided. A brief description of KGC is given below.

KGC: It is introduced in identity-based cryptosystem (IBC) as a trusted authority called key generation center (KGC) [17], which comprises a set of four algorithms, namely *setup*, *extract*, *encryption* and *decryption* as described below:

Setup: It takes two groups G_1 (additive) and G_2 (multiplicative) of same order q and a bilinear pairing map $e : G_1 \times G_1 \rightarrow G_2$. In addition, two hash functions are also specified as $H_1 : \{0, 1\}^* \rightarrow G_1$ //from identity to a point in G_1 and $H_2 : G_2 \rightarrow \{0, 1\}^m$ for some m // hash to message of length m . It takes a master secret s as KGC's private key and computes the corresponding public key $P_0 = sP$, where $s \in Z_q^*$ and P is a base point of G_1 . It then publishes the parameters $\langle G_1, G_2, e, q, H_1, H_2, H_3, P_0, P \rangle$.

Extract: This algorithm extracts private keys of participants from their identities. For a participant i , KGC computes $Q_i = H_1(\text{ID}_i)$ and then computes the corresponding private key $D_i = sQ_i$. Then, KGC sends the private key D_i through a secure channel to the participant i .

Encrypt: In order to send an encrypted message of $M \in \{0, 1\}^m$ to a participant j by i , i computes $Q_j = H_1(\text{ID}_j)$, $U = rP$ and $V = M \oplus H_2(e(Q_j, P_0)^r)$ for a random number $r \in Z_q^*$ selected by participant i . Then, the participant i sends the ciphertext $C = (U, V)$ to participant j .

Decrypt: On receiving C , the participant can decrypt and get the message as $M = V \oplus H_2(e(D_j, U))$.

In our KGC-based SSS, initially, all n participants must register by sending their public identities to KGC, for which KGC uses the following algorithms to implement (t, n) SSS:

Setup: It takes a large prime number p and sets the multiplicative finite cyclic group $G = \langle Z_p, \times \rangle$. It takes RSA parameters $N = a.b \pmod p$ for large primes a and b , and generates RSA private-public-key pair PR and PU. It considers a hash function $H : \{0, 1\}^* \rightarrow G$. It then publishes the parameters $\langle Z_p, p, n, N, PU, H \rangle$.

Share generation: It takes a random $(t-1)$ polynomial $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$, where $1 < t \leq n$ and $a_i \in Z_p$ for $i = 0, 1, \dots, t - 1$, and calculates the secret shares $f(j)$ for each user j , where $j = 1, 2, \dots, t$.

Signature generation: It computes $Y = \text{Hash}(f(1)||f(2)||\dots||f(t))$ and RSA signature $Z = Y^{PR} \pmod N$.

Share distribution: Each share $f(j)$ for $j = 1, 2, \dots, t$ is transmitted to participant j through a secret channel and publishes Z .

Our KGC-based (t, n) threshold SS algorithm SSS-KGC() is presented below:

```

SSS-KGC()// Iterates infinitely
{ while ()
  { 1) Group formation phase: Any  $t$  ( $1 < t \leq n$ ) participants send messages to KGC
    for negotiating a secret between them.

    1) KGC executes the following:
       i) Runs share generation, ii) Runs signature generation, iii) Share
          distribution

    2) Share verification phase: All  $t$  participants exchange their shares among
       themselves and verify individually with  $Z$  as follows:
       i) Decrypt  $Z$  to get  $Y$  as  $Z^{PU} \pmod N = Y^{PU, PR} \pmod N = Y \pmod N$ 
       ii) Compute  $Y' = \text{Hash}(f'(1)||f'(2)||\dots||f'(t))$ , where the share  $f'(j)$  is
          received from other participants.
       iii) If  $Y' = Y$ , there is no share cheating, else share(s) cheating is occurred
          and the process is terminated.

    3) Secret reconstruction phase: If it is not terminated, all  $t$  participants run
       Lagrange interpolation formula and reconstruct the correct secret  $s = f(0)$ .
  }

  }//end-while
}//end SSS-KGC

```

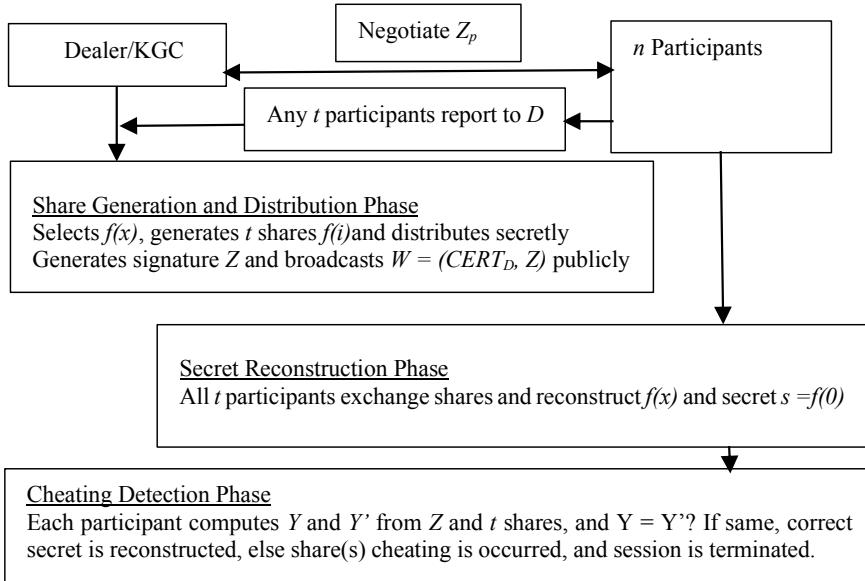


Fig. 2 Flow diagram based on dealer and KGC

Thus, our KGC-based SSS is cheating-free and secure. A flow diagram (Fig. 2) of the scheme is given below:

As a performance study, a comparison of the proposed SSSs with Shamir's one is given in Table 1.

where T_{pol} is time for polynomial computation, T_{Lag} is time for Lagrange interpolation, T_{sig} is time for signature generation, T_{ver} is time for signature verification, T_{hash} is time for hash operation and X is number of times $KGC\text{-}SSS()$ algorithm called.

5 Conclusion

Two designs over Shamir's (t, n) threshold SSS in terms of share cheating and/or disobey of dealer are presented. A simpler method with probable cheating detection is presented in the first scheme, and a (t, t) and a (t, n) using dealer and KGC, respectively, are proposed in the other scheme. All schemes are secure as Shamir's construct, which is based on information-theoretic security, is followed. Our SSSs are efficient as well, and in this regard, a comparison table (Table 1) is provided. Blockchain, an emerging and promising technology, is suitable for secured information exchange, and in the future we intend to implement threshold schemes through distributed ledger system.

Table 1 Comparison with existing methods

SSSs	Methods used	Security protection against		Computation cost	Remarks
		Dealer	Share cheating		
Shamir	Dealer-based (t, n) scheme and information-theoretic security	No	No	$nT_{\text{pol}} + tT_{\text{Lag}}$	Vulnerable and reduces applications
Proposed	Dealer-based (t, n) scheme and information-theoretic security	No	Probable secure	$nT_{\text{pol}} + tT_{\text{Lag}}$	Lesser vulnerable and enhances applications
	Dealer-based (t, t) scheme and information-theoretic security	No	Yes	$t(T_{\text{pol}} + T_{\text{Lag}} + T_{\text{sig}} + T_{\text{ver}} + 2T_{\text{hash}})$	More secure and enhances applications
	KGC-based (t, n) scheme and information-theoretic security	Yes	Yes	$X(t(T_{\text{pol}} + T_{\text{Lag}}) + T_{\text{sig}} + T_{\text{ver}} + 2T_{\text{hash}})$	Secured and supports full applications

References

- Li S, Zhuge JW, Li X (2013) Study on BGP security. Chin J Softw 24(1):121–138
- Saini H, Rao YS, Panda TC (2012) Cyber-crimes and their impacts: a review. Int J Eng Res Appl (IJERA) 2(2):202–209. ISSN: 2248–9622
- Lagazio M et al (2014) A multi-level approach to understanding the impact of cybercrime on the financial sector. Comput Secur Elsevier 45:58–74
- Kim J, Tong L, Thomas RJ (2014) Dynamic attacks on power systems economic dispatch. In: 48th Asilomar conference on signals, systems and computers, IEEE, pp 345–349
- Gutub A, AlKhadaidi T (2020) Smart expansion of target key for more handlers to access multimedia counting-based secret sharing. Multimed Tools Appl 79:17373–17401
- Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inf Theor IT-22(6):644–654
- Shamir A (1979) How to share a secret. Commun ACM 22:612–613
- Tompa M, Woll H (1989) How to share a secret with cheaters. J Cryptol 1:133–138
- Harn L, Lin C (2009) Detection and identification of cheaters in (t, n) secret sharing scheme. Des Codes Cryptograph ACM Dig Libr 52:15–24
- Biswas AK, Dasgupta M (2020) Cryptanalysis and enhancement of Harn-Lin's secret sharing scheme with cheating detection. In: 2020 1st international conference on power, control and computing technologies (ICPC2T), Raipur, India, pp. 27–31. <https://doi.org/10.1109/ICPC2T48082.2020.9071470>
- Fujii Y, Tada M, Hosaka N, Tochikubo K, Kato T (2005) A Fast $(2, n)$ -Threshold scheme and its application. CSS 2005:631–636
- Tassa T (2007) Hierarchical threshold secret sharing. J Cryptol 20(2):237–264
- Concone F, Lo Re G, Morana MSMCP (2020) A secure mobile crowd sensing protocol for fog-based applications. Hum Cent Comput Inf Sci 10:28
- Biswas AK, Dasgupta M (2020) Two polynomials based (t, n) threshold secret sharing scheme with cheating detection. Cryptologia:1–14. <https://doi.org/10.1080/01611194.2020.1717676>

15. Li C, Xu C, Zhao Y, Chen K, Zhang X (2020) Certificateless identity-concealed authenticated encryption under multi-KGC. In: Liu Z, Yung M (eds) Information security and cryptology. Inscrypt 2019. Lecture notes in computer science, vol 12020. Springer, Cham
16. Sun H, Li L, Zhang J, Huang W (2020) An improved dynamic certificateless authenticated asymmetric group key agreement protocol with trusted KGC. In: Tian Y, Ma T, Khan M (eds) Big data and security. ICBDS 2019. Communications in computer and information science, vol 1210. Springer, Singapore
17. Boneh D, Franklin M (2003) Identity-based encryption from the Weil pairing. SIAM J Comput 32(3):586–615

Chapter 12

Physical Layer Security-Based Relay Selection for Wireless Cooperative Networks: A Reinforcement Learning Approach



Anil Kumar Kamboj, Poonam Jindal, and Pankaj Verma

1 Introduction

With the rapid advancement in quantum computing and code-breaking techniques, the upper layer security approaches solely unable to secure wireless communication systems. Hence, it is advisable to fully utilize the attributes of the wireless channel and transmitter/receiver to design new security techniques at the physical layer. Several techniques have been proposed to enhance physical layer security, such as cooperative communication, antenna selection, beamforming, waveform design, and coding [1, 2].

In the last few years, numerous techniques introduced for relay selection by considering different performance metrics [3–10]. However, the physical layer secured relay selection for securing wireless transmission against adversaries drawing attention nowadays. Relay selection may be a game-changing approach for emerging technologies such as the Internet of Things (IoT) networks, D2D communication, unmanned aerial vehicles (UAV), cognitive radio networks, satellite communication systems, and vehicular communication systems [1, 9, 10]. The relay selection mainly classified into two types according to the processing of the received signal at the relay node as amplified and forward (AF), decode and forward (DF). The relay nodes which amplify the received signal and forward it to the user is AF. However, the relay node, using DF protocol decode, process and re-encode the received signal before forwarding it to the destination [2, 3].

A. K. Kamboj (✉)

Department of Electronics and Communication, NIT, Kurukshetra, Haryana, India

P. Jindal · P. Verma

Department of Electronics and Communication Engineering, NIT, Kurukshetra, Haryana, India

e-mail: poonamjindal81@nitkkr.ac.in

P. Verma

e-mail: pankaj@nitkkr.ac.in

The PLS-based relay selection for cooperative networks depending on the availability of CSI can be classified into conventional, optimal (full-CSI), and minimal relay selection schemes [4]. The selection of the best relay out of a group of relays using channel information enhances the spectrum efficiency of the system. Al-Qahtani et al. [4] investigated the secrecy enhanced opportunistic relay selection for two cases, first when CSI of the eavesdropper is available and second when the CSI of the eavesdropper is unavailable. Several relay selection techniques were introduced in the last decade for different environments and wireless networks [5, 6]. However, to fulfill the demands of emerging and future wireless technologies, intelligence needs to be integrated with the relay nodes [11]. Therefore, to embed the intelligence in the relay nodes, the authors in [12] introduced the Q-learning-based RS for reliability enhancement. Q-learning algorithm is one of the reinforcement learning (RIL) techniques and contains agents and an environment, wherever the agent tries to learn an unknown environment. It is one of the most applied off-policy schemes. Su et al. [13] presented a deep Q-learning algorithm for relay selection to enhance the reliability of the cooperative networks. However, the security of the network is equally important, and intelligent secrecy enhanced algorithms are the requirement of emerging wireless networks. Q-learning is an off-policy and less complex ML algorithm that introduces self-learning in each relay node and requires a small amount of information exchange between the nodes [14]. To the best of the author's knowledge, few machine learning-based relay selection algorithms to enhance the PLS are reported at the time of writing this paper. Motivated by this, our paper introduces a Q-learning-based PLS enhanced relay selection for cooperative networks. This work deals with the non-regenerative dual-hop wireless communication in the presence of a pair of transmitter and receiver, an eavesdropper, and a group of relays. The proposed algorithm combines the Q-learning with PLS, and a single relay is selected out of a group of relays to enhance the PLS. The remainder of this paper is organized as follows: Sect. 2 introduces the system model and relay selection, intelligent relay selection, and proposed algorithm presented in Sect. 3, results discussed in Sect. 4, and Sect. 5 belongs to the conclusion.

2 System Model and Relay Selection

2.1 System Model

Consider a wireless cooperative communication network which consists of a source denoted by U_1 , destination denoted by U_2 , M half-duplex relays denoted by R_1, R_2, \dots, R_M , and an eavesdropper denoted by E , as shown in Fig. 1. A direct link between source and destination assumed not to be existed. Each transmission from U_1 to U_2 is only possible with the help of relay nodes using amplified and forward (AF) protocol. Every node in the network is embedded with a single antenna and works in half-duplex mode. All the present channels assumed with channel reciprocity and

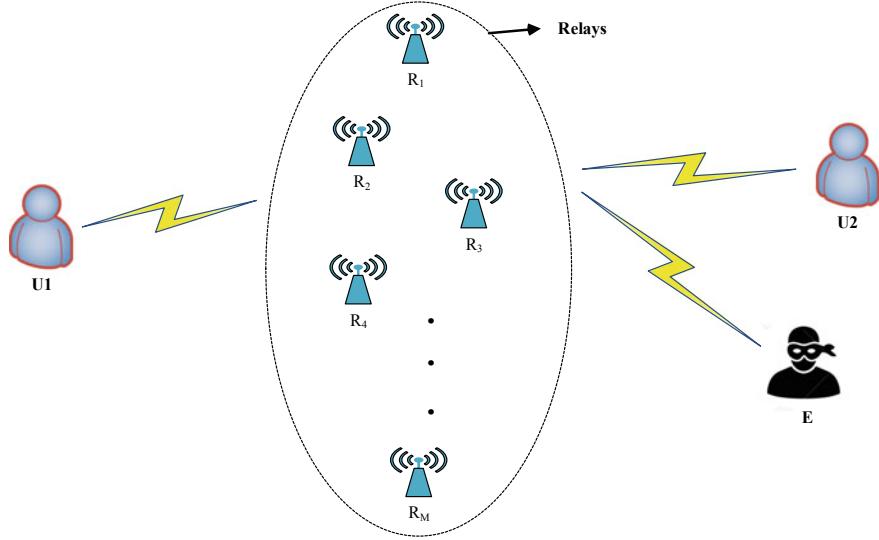


Fig. 1 System model for wireless cooperative networks

independent Rayleigh fading. The communication between source and destination completed in two-time slots, as shown in Fig. 1. In the first time slot, U_1 transmits the signal X with power P_S to the relays, and in the second time slot, one relay denoted by R_i is selected to forward the received information to U_2 with power P_R .

For the first time slot, the received signal at i th relays written as in [5]

$$Y_{U_1, R_i} = \sqrt{P_S} g_{U_1, R_i} X + N_{U_1, R_i} \quad (1)$$

where $g_{U_1, R_i} = \frac{|h_{U_1, R_i}|^2}{d_{U_1, R_i}^{-\gamma}}$ is channel gain of U_1 and i th relay link, N_{U_1, R_i} denotes additive white Gaussian noise (AWGN) with zero mean and variance σ_{U_1, R_i}^2 , distance between U_1 and i th relay denoted by d_{U_1, R_i} , path loss exponent denoted by γ , and h_{U_1, R_i} denotes the channel coefficients including the fading and shadowing.

In the second time slot, the signal received by U_2 is expressed as

$$Y_{R_i, U_2} = A_{R_i} g_{R_i, U_2} Y_{U_1, R_i} + N_{R_i, U_2} \quad (2)$$

where $A_{R_i} = \sqrt{\frac{P_R}{g_{U_1, R_i} P_S + \sigma_{U_1, R_i}^2}}$ is the R_i relay variable amplification factor, $g_{R_i, U_2} = \frac{|h_{R_i, U_2}|^2}{d_{R_i, U_2}^{-\gamma}}$ is the channel gain of i th relay and U_2 link, and N_{R_i, U_2} denotes additive white Gaussian noise with zero mean and variance σ_{R_i, U_2}^2 , distance between i th relay and

U_2 denoted by d_{R_i, U_2} , γ denotes the path loss exponent, and h_{R_i, U_2} denotes channel coefficients including the fading and shadowing.

The source U_1 assumed out of reach of the eavesdropper. The signal received by eavesdropper (E) expressed as

$$Y_{R_i, E} = A_{R_i} g_{R_i, E} Y_{U_1, R_i} + N_{R_i, E} \quad (3)$$

where $g_{R_i, E} = \frac{|h_{R_i, E}|^2}{d_{R_i, E}^\gamma}$ is the channel gain of i th relay and E link, $N_{R_i, E}$ denotes additive white Gaussian noise with zero mean and variance $\sigma_{R_i, E}^2$, distance between i th relay and E denoted by $d_{R_i, E}$, γ denotes the path loss exponent, and h_{U_1, R_i} denotes channel coefficients including the fading and shadowing.

Combining (1) and (2) to express the signal at the U_2 in terms of the input signal

$$Y_{R_i, U_2} = \sqrt{\frac{P_{R_i}}{g_{U_1, R_i} P_S + \sigma_{U_1, R_i}^2}} g_{R_i, U_2} (\sqrt{P_S} g_{U_1, R_i} X + N_{U_1, R_i}) + N_{R_i, U_2} \quad (4)$$

Similarly, combining (1) with (3) to produce the signal received by E

$$Y_{R_i, E} = \sqrt{\frac{P_{R_i}}{g_{U_1, R_i} P_S + \sigma_{U_1, R_i}^2}} g_{R_i, E} (\sqrt{P_S} g_{U_1, R_i} X + N_{U_1, R_i}) + N_{R_i, E} \quad (5)$$

2.2 Secrecy Analysis and Relay Selection

Single relay selected for information transmission out of a group of relays enhances the performance of system. The source to destination channel link mutual information is given by

$$\begin{aligned} I_{U_1, U_2}^{R_i} &= \frac{1}{2} \log_2 (1 + \gamma_{U_1, U_2}) \\ I_{U_1, U_2}^{R_i} &= \frac{1}{2} \log_2 \left(1 + \frac{\gamma_{U_1, R_i} \gamma_{R_i, U_2}}{1 + \gamma_{U_1, R_i} + \gamma_{R_i, U_2}} \right) \end{aligned} \quad (6)$$

where

$$\begin{aligned} SNR_{U_1, U_2}^{R_i} &= \gamma_{U_1, U_2}^{R_i} \\ &= \frac{\frac{P_S |h_{U_1, R_i}|^2}{N_{U_1, R_i}} \frac{P'_S |h_{R_i, U_2}|^2}{N_{R_i, U_2}}}{\frac{P_S P'_S |h_{U_1, R_i}|^2 |h_{R_i, U_2}|^2}{N_{U_1, R_i} N_{R_i, U_2}} + \frac{P_S |h_{U_1, R_i}|^2}{N_{U_1, R_i}} + \frac{P'_S |h_{R_i, U_2}|^2}{N_{R_i, U_2}} + 1} \end{aligned}$$

$$= \frac{\beta^2 P_S |h_{U_1, R_i}|^2 |h_{R_i, U_2}|^2 P'_S}{\beta^2 P_S P'_S |h_{U_1, R_i}|^2 |h_{R_i, U_2}|^2 + \beta P_S |h_{U_1, R_i}|^2 + \beta |h_{R_i, U_2}|^2 P'_S + 1}$$

is the signal to noise ratio (SNR) of the source to destination link.

For simplicity,

$$\begin{aligned} N_{U_1, R_i} &= N_{R_i, U_2} = 1/\beta \\ SNR_{U_1, U_2}^{R_i} &= \gamma_{U_1, U_2}^{R_i} = \frac{\gamma_{U_1, R_i} \gamma_{R_i, U_2}}{1 + \gamma_{U_1, R_i} + \gamma_{R_i, U_2}} \\ \gamma_{U_1, R_i} &= \frac{P_S g_{U_1, R_i}}{N_0}, \gamma_{R_i, U_2} = \frac{P'_S g_{R_i, U_2}}{N_0} \end{aligned}$$

For simplicity noises at each link assumed N_0 and power P_S .

Similarly, the mutual information of relay to eavesdropper link is given by

$$\begin{aligned} I_{U_1, E}^{R_i} &= \frac{1}{2} \log_2 (1 + \gamma_{U_1, E}) \\ I_{U_1, E}^{R_i} &= \frac{1}{2} \log_2 \left(1 + \frac{\gamma_{U_1, R_i} \gamma_{R_i, E}}{1 + \gamma_{U_1, R_i} + \gamma_{R_i, E}} \right) \end{aligned} \quad (7)$$

The achievable secrecy rate is represented by

$$\mathbb{R}_{\text{Sec}}^{R_i} = \mathbb{R}_{U_1, D}^{R_i} - \mathbb{R}_{U_1, E}^{R_i} = \frac{1}{2} \left[\log_2 \left(\frac{1 + \gamma_{U_1, U_2}^{R_i}}{1 + \gamma_{U_1, E}^{R_i}} \right) \right]^+$$

The secrecy outage occurs when the difference of the secrecy rate of legitimate link and eavesdropper link decreases below some fixed secrecy rate \mathbb{R}_F . The probability of secrecy outage expressed by

$$\begin{aligned} P_{\text{SOP}} &= \Pr \left[\mathbb{R}_{\text{Sec}}^{R_i} < \mathbb{R}_F \right] \\ &= \Pr \left[\frac{1}{2} \max \left(\log_2 \left(\frac{1 + \gamma_{U_1, U_2}^{R_i}}{1 + \gamma_{U_1, E}^{R_i}} \right) \right) < \mathbb{R}_F \right] \\ &= \Pr \left[\max \left(\frac{1 + \gamma_{U_1, U_2}^{R_i}}{1 + \gamma_{U_1, E}^{R_i}} \right) < 2^{2\mathbb{R}_F} \right] \end{aligned} \quad (8)$$

```

1:   Initialize  $S, \alpha = 0.6, \delta = 0.6, Q(S, A_i) = 0$ 
2:   For  $i = 1, 2, 3, \dots, M$  do
3:       Broadcast the message to all relay nodes
4:       Initialize current state  $S_t$ 
5:       Choose  $A_t \in A$  from  $S_t$  using (10)
6:       Take action  $A_t$ 
7:       Discover the reward using (8)
8:       Find new state  $S_{t+1} \in S$ 
9:       Update  $Q(S, A_i)$  using (9)
10:       $S_t \leftarrow S_{t+1}$ 
11:   end for
12:   return Optimal relay node

```

Fig. 2 Proposed intelligent algorithm for relay selection

3 Intelligent Relay Selection

Q-learning algorithm is one of the MDP-based RIL technique which contains an environment comprising an agent that tries to learn an unknown environment. The proposed algorithm for relay selection is shown in Fig. 2.

Quality function for particular state-action pair updated according to the Bellman's equation as given below [8, 9];

$$Q(S, A_i) \leftarrow Q(S, A_i) + \alpha [R + \delta \max Q(S', A') - Q(S, A_i)] \quad (9)$$

where α denotes the learning rate whose value lies between 0 and 1, R is the reward, δ denotes the discount factor, $Q(S, A_i)$ denotes Q-value function at time t , and $Q(S', A')$ is the Q-value function at time $t + 1$.

Q-learning algorithms use a Q-table to maintain the Q-values for each state-action pair, where rows represent states and columns actions. A ε -greedy algorithm is adopted to choose the optimal policy by using exploration and exploitation as

$$\pi(S) = \arg \max_{A_i \in A} Q(S, A_i) \quad (10)$$

where $\pi(S)$ denotes the policy, and Q-value is obtained by adopting the policy $Q(S, A_i)$ for an action A_i in state S .

4 Result and Discussion

The performance of the proposed algorithm is compared with the minimum, conventional, and optimal relay selection schemes [3]. The source transmits power assumed as 0.01 W, and relay nodes used variable gain to make their output power equal to

source power. The simulation environment consists of $M = 3$ uniformly distributed relays in a $50 \text{ m} \times 50 \text{ m}$ area. The nodes U_1 , U_2 , and eavesdropper are placed at $(0,0)$, $(100, 0)$, $(100, -75)$, respectively. The distance between source and destination is 100 m , and no direct link exists between them. Rayleigh fading channel adopted, and the path loss exponent of the channel is assumed as 2.5 . The learning rate and discount factor for the Q-learning algorithm are set at 0.6 .

Figure 3 shows the comparison of the proposed algorithm with the minimum, conventional, and optimal schemes. The secrecy outage probability versus signal to noise ratio plot exhibits that the performance of the Q-learning-based algorithm is approximately the same as that of the optimal scheme. The optimal and minimal RS schemes curves exhibit the same slope. However, the conventional selection curve displays the smallest slope because it does not consider the relay-receiver link CSI. The optimal selection schemes produce the best result due to the knowledge of all the links CSI. The proposed algorithm can get the performance of the optimal relay selection scheme and provide the self-learning in each relay node which decreases the complexity of the system. Moreover, in the Q-learning-based algorithm, the relay nodes exchange a small amount of information with other nodes which further improves its performance.

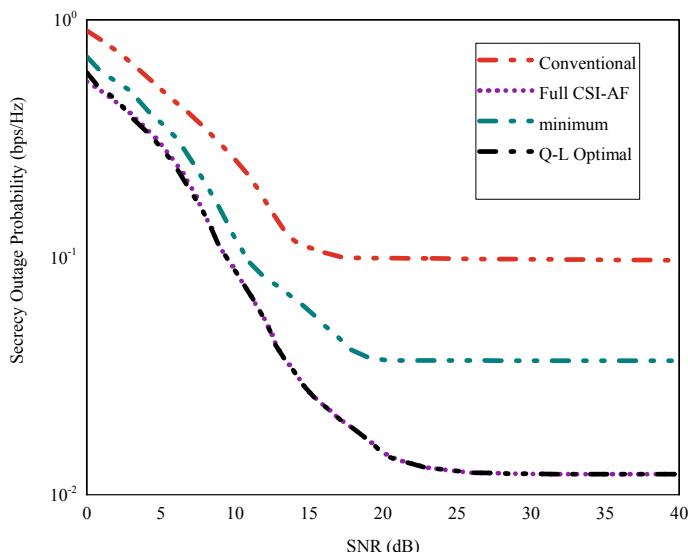


Fig. 3 Secrecy outage probability comparison of the relay selection techniques

5 Conclusion

This paper investigates the Q-learning-based relay selection for PLS improvement. The relay selection in cooperative communication improves the secrecy of the network. The Q-learning for relay selection is best suited when channel conditions change rapidly, and data is unavailable for training the ML algorithms. The simulation result shows that the proposed algorithms can mimic the optimal relay selection scheme in terms of secrecy outage probability. Moreover, the proposed algorithm is less complex due to the requirement of less information exchange between the nodes. In future work, we will consider the joint relay and jammer selection for the improvement of PLS.

References

1. Jameel F, Wyne S, Kaddoum G, Duong TQ (2018) A comprehensive survey on cooperative relaying and jamming strategies for physical layer security. *IEEE Commun Surv Tutorials* 21(3):2734–2771
2. Pahuja S, Jindal P (2019) Cooperative communication in physical layer security: technologies and challenges. *Wirel Pers Commun* 108(2):811–837
3. Krikidis I, Thompson JS, McLaughlin S (2009) Relay selection for secure cooperative networks with jamming. *IEEE Trans Wirel Commun* 8(10):5003–5011
4. Al-Qahtani FS, Zhong C, Alnuweiri HM (2015) Opportunistic relay selection for secrecy enhancement in cooperative networks. *IEEE Trans Commun* 63(5):1756–1770
5. Bao VNQ, Linh-Trung N, Debbah M (2013) Relay selection schemes for dual-hop networks under security constraints with multiple eavesdroppers. *IEEE Trans Wirel Commun* 12(12):6076–6085
6. Hui H, Swindlehurst AL, Li G, Liang J (2015) Secure relay and jammer selection for physical layer security. *IEEE Signal Process Lett* 22(8):1147–1151
7. Yang L, Chen J, Jiang H, Vorobyov SA, Zhang H (2017) Optimal relay selection for secure cooperative communications with an adaptive eavesdropper. *IEEE Trans Wirel Commun* 16(1):26–42
8. Khandaker MRA, Wong KK, Zheng G (2017) Truth-telling mechanism for two-way relay selection for secrecy communications with energy-harvesting revenue. *IEEE Trans Wirel Commun* 16(5):3111–3123
9. Liu D et al (2019) Task-driven relay assignment in distributed UAV communication networks. *IEEE Trans Veh Technol* 68(11):11003–11017
10. Ding X, Zou Y, Ding F, Zhang D, Zhang G (2019) Opportunistic relaying against eavesdropping for internet-of-things: a security-reliability tradeoff perspective. *IEEE Internet Things J* 6(5):8727–8738
11. Morocho-Cayamcela ME, Lee H, Lim W (2019) Machine learning for 5G/B5G mobile and wireless communications: potential, limitations, and future directions. *IEEE Access* 7:137184–137206
12. Jadoon MA, Kim S (2017) Relay selection algorithm for wireless cooperative networks: a learning-based approach. *IET Commun* 11(7):1061–1066
13. Su Y, Lu X, Zhao Y, Huang L, Du X (2019) Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks. *IEEE Sens J* 19(20):9561–9569
14. Jang B, Kim M, Harerimana G, Kim JW (2019) Q-Learning algorithms: a comprehensive classification and applications. *IEEE Access* 7:133653–133667

Chapter 13

A Review of Security Threats in Software-Defined Networking



Sukhveer Kaur , Krishan Kumar , and Naveen Aggarwal

1 Introduction

Traditional network comprises a set of network devices in which data plane (that forwards packets) and control plane (that instructs the data plane) are closely bounded into a same device. Moreover, each device is vendor specific and performs only the dedicated functionality. Any modification in the existing behavior and introducing new features is very tedious and incurs high cost [1]. On the other hand, software-defined networking (SDN) is a three-layered architectural framework that is defined by the Open Network Foundation (ONF) [2] where control part (control layer) is decoupled from the network device. The shifted control plane is called the SDN controller or brain of the network. Now, the network device (SDN switch) is only a simple merchant silicon box that contains only a forwarding functionality. It is the responsibility of control plane to insert the control logic into the SDN switches. The decoupling of both the planes allows the efficient and reliable network management through programming. In addition, the cost of SDN devices is less due to the use of open-source SDN controllers.

In spite of the immense advantages of SDN, organizations are not deploying the pure SDN network due to the various security issues. The centralized and decoupled architecture makes it vulnerable to various security attacks [3]. In the literature, Antikainen et al. [4] identified only those attacks on the SDN framework, which are launched by compromising the OpenFlow switch. However, they have not considered the other vulnerable points in SDN architecture, such as lack of trusted API, single point of failure at the controller, and lack of authorization mechanism at application plane. In [5], authors have given an abstract view of some of the attacks on the SDN environment. However, these attacks (policy conflicts, malicious applications, MITM, eavesdropping, and tampering), which are available in the literature, are

S. Kaur (✉) · K. Kumar · N. Aggarwal
UIET, Panjab University, Chandigarh, India

not classified. In [6], authors have identified some of the security attacks, but these attacks (flow table modification, topology spoofing, tampering, information disclosure, and resource exhaustion at application plane) are not discussed. To present a comprehensive review, we have given a hierarchical classification of attacks on SDN framework.

2 SDN Architecture

Open Network Foundation (ONF) [2], Internet Research Task Force (IRTF) [7], Internet Engineering Task Force (IETF) [8] are the major organizations that are working on conducting standardization activities for SDN. Figure 1 shows the components of SDN architecture.

- (i) **Application Plane:** The upper layer of SDN framework is the application layer. Different types of applications such as load balancer, NAT, firewall, IDS, intrusion prevention system (IPS) are running on application layer that provides different network functions. These applications interact with the controller using the Northbound API [9].
- (ii) **Control Plane:** The central layer of SDN architecture is called a control layer that manages the entire network. The responsibility of the control layer is to instruct the data plane where to send the packet by translating the flow rules and installing it into the data plane [10].
- (iii) **Data Plane:** This is also called infrastructure layer that comprises various forwarding devices. These devices are also called SDN switches and consist of physical or virtual switches [11]. The switches interact with the control plane to forward the packets using the southbound API (OpenFlow protocol).

3 Layered Taxonomy of SDN Security Threats

In this section, we proposed the taxonomy of SDN security threats based on the various vulnerable points in the SDN framework (Fig. 2).

3.1 Security Threats on Data Plane

Data plane comprises dump forwarding devices that make it susceptible to various security attacks such as flow table modification and topology spoofing attacks. Moreover, the limited memory of the switches is exploited by the attackers to perform flow table overloading and buffer saturation attack [12].

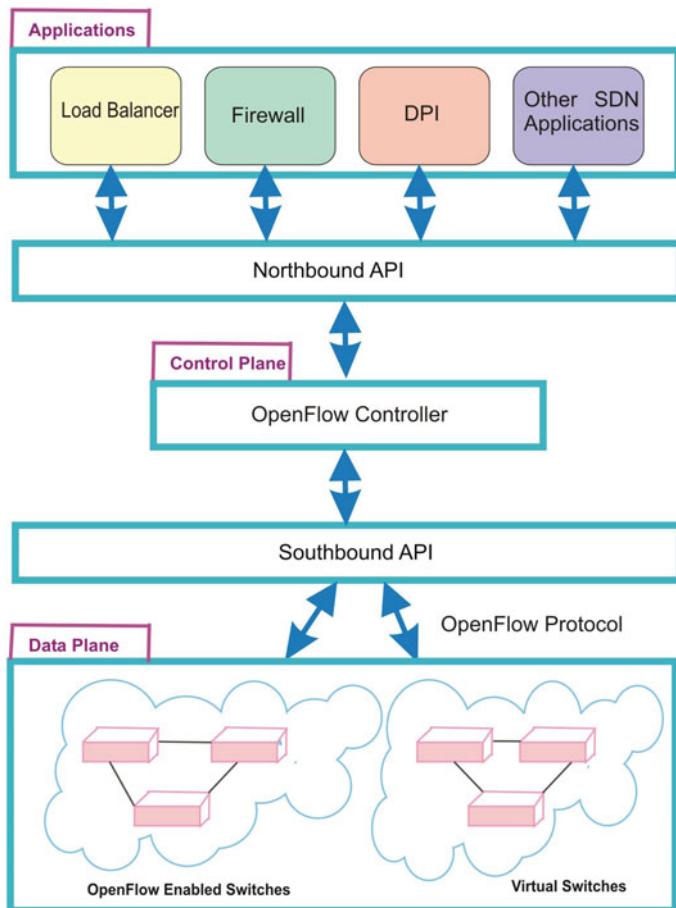


Fig. 1 SDN architecture

- Flow table modification:** In this, attacker inserts, delete, or modifies the flow entries of OpenFlow switches, which can lead to further attacks like eavesdropping and MITM attacks. In the eavesdropping attack, the attacker can insert the rule into the OpenFlow switch to duplicate the traffic that is passing through OpenFlow switch and send it to the attacker machine. Figure 3 [4] shows that the attacker modified the flow entry to duplicate the packet that is sent by host A, having IP address 172.24.0.5 to host B having IP address 172.24.0.10. Similarly, the attacker can launch a MITM attack by redirecting the packets that are going through the OpenFlow switch. In the MITM attack, the attacker can change the destination IP and port number in the flow entry to redirect the packets to the attacker machine.

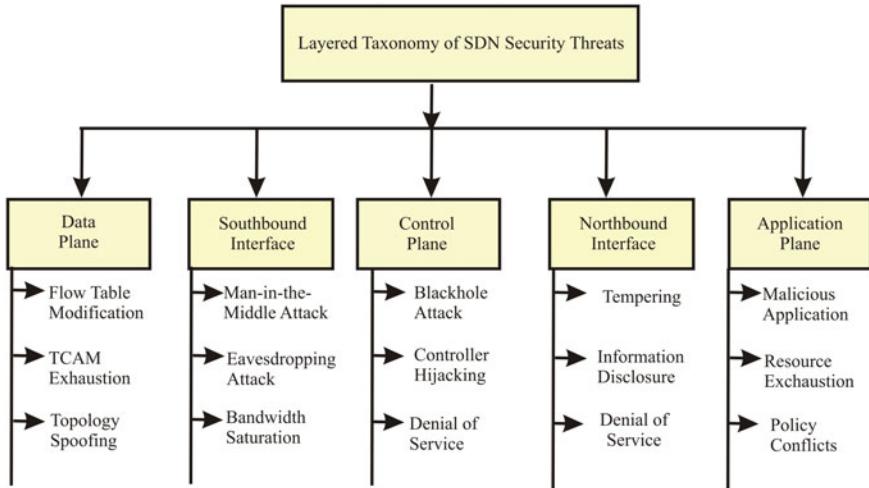


Fig. 2 Layered taxonomy of SDN security threats

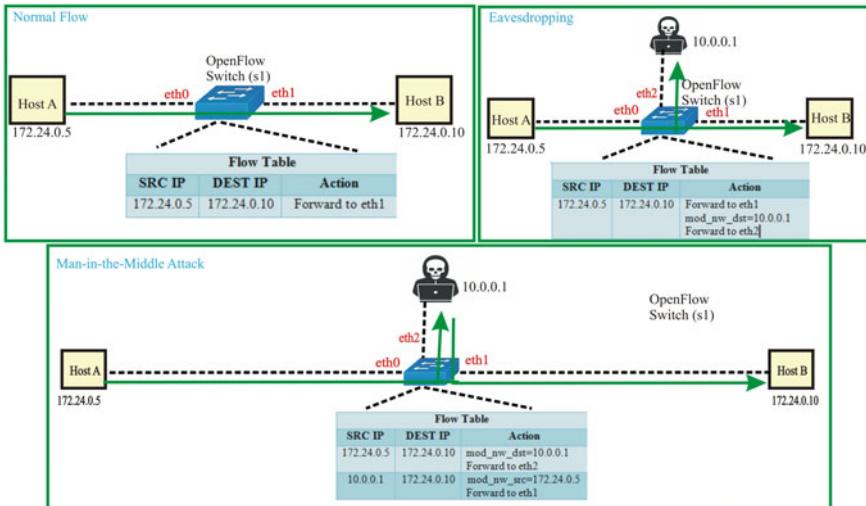


Fig. 3 Flow table modification attack

- (ii) **TCAM Exhaustion:** The flow table is residing in the TCAM (ternary content addressable memory) memory of the switch to store the flow rules. TCAM is a very high-speed memory that searches the data within the single clock cycle [13]. Due to the limited TCAM memory, the OpenFlow switch is vulnerable to DDoS attacks. The switch also has an OpenFlow agent that is used to forward the unmatched packets to the control plane. The OpenFlow agent has limited capacity to process unmatched packets as it runs on the low-end processor.

When it receives prodigious amounts of packets, it gets overloaded; therefore, it is not able to process legitimate packets.

- (iii) **Topology Spoofing:** The centralized SDN controller can visualize the entire network, so it can select the path to traverse the packets. The topology spoofing attack creates a bogus network map for the controller to redirect the packets. In this attack, an attacker can insert fake virtual links or switches in the path to change the network map for the controller [4, 14]. As the default implementation of TLS provides only controller authentication, leaving the possibility of adding fake virtual switches in the network.

3.2 *Security Threats on Southbound API*

The data plane switches communicate with the control plane using the southbound API. However, due to the limited bandwidth and lack of TLS security [15], this interface is suffering from the following security threats.

- (i) **Man-in-the-middle Attack (MITM):** In the MITM attack, malicious node put itself between controller and switches to intercept the communication that has been taking place. It can modify the packets that are transmitted by OpenFlow switch to the controller [6]. In addition to that, when the controller tries to insert the instructions in the OpenFlow switch, it can modify those instructions to drop or redirect these packets to malicious nodes.
- (ii) **Eavesdropping Attack:** This is a kind of passive attack whose purpose is to get sensitive network information which may be exploited by adversaries to launch more serious attacks such as to gain illegitimate access to the network or to perform a MITM attack. For preventing this attack, two-way TLS level security should be implemented in the OpenFlow protocol to send the message in the encrypted form [6].
- (iii) **Bandwidth Saturation:** OpenFlow switch forwards unmatched packets to the controller for decision making. By default, it sends only 128 bytes of the packet and stores the packet payload in the buffer. When an attacker sends excessive amounts of traffic to OpenFlow switch, then its buffer gets overloaded. To make it worse, OpenFlow 1.4 says that when the buffer of OpenFlow agent overflows, then it needs to send the entire packet to the controller, which can consume the bandwidth of the data-control plane communication channel that incurs a delay for the legitimate users [16].

3.3 *Security Threats on Control Plane*

The controller manages the entire network from the single point that makes it an essential component of SDN framework. While the centralized architecture of SDN

makes the network management efficient and reliable, but if it goes into the wrong hands, then it has devastating effects on overall network architecture [5].

- (i) **Blackhole Attack:** The controllers use the OpenFlow Discovery Protocol (OFDP) to find the connection between OpenFlow switches [17]. However, there are some limitations in the OFDP protocol as it uses non-authenticated and clear LLDP packets to discover the link, which makes it susceptible to various security threats such as switch spoofing and link fabrication.
- (ii) **Controller Hijacking:** An attacker that hijacked the controller machine gets the control of all network devices. After getting the access of the controller, an attacker can manipulate the network in any way. By doing this, the attacker can add, delete, modify flow table rules, dropping the packets, and modify the destination address to redirect the packets to malicious hosts [6]. Therefore, it may affect the availability of the controller, the integrity of packets, and the confidentiality of sensitive information. For preventing this type of attack, the proper authentication mechanism must be required for the controller to make any changes in the network devices.
- (iii) **Denial of Service:** An attacker can compromise one or more host machines to send massive amounts of malicious packets to forwarding devices for flooding the network. An attacker modifies the header of packets in such a way that the forwarding device needs to send these unmatched packets to the controller machine for decision making [18]. These malicious packet-in messages exhaust the controller's computational resources (storage and processing power), thus making it unavailable for legitimate users [22].

3.4 Security Threats on Northbound Interface

Northbound API provides the communication between the control layer and application layer. This interface has a lower investment cost and a rapid development cycle because it does not need any specialized equipment. However, it suffers from the following security threats.

- (i) **Tampering:** The communication over the northbound interface can be insecure if the data is not encrypted with SSL/TLS. When the malicious user gets control over this interface, he/she can alter the packets before sending them to the controller or application. The alteration of packets can downgrade the whole network [15, 21]. Therefore, SSL/TLS security should be enforced to secure the entire network.
- (ii) **Information Disclosure:** The SDN applications send requests to the controller to collect the network configuration and statistics information. If this information is disclosed to the malicious user, then it can have catastrophic impacts on the network [5]. The confidentiality of this interface can be achieved by encrypting the communication.

- (iii) **Denial of Service:** In this attack, the malicious applications make the northbound interface unavailable for the authorized application by sending resource-intensive superfluous requests to the controller [5]. For preventing this attack, the proper access control mechanism is required to give minimal access to the applications.

3.5 *Security Threats on Application Plane*

It contains various types of applications such as traffic monitoring, load balancing, and firewall. Due to the absence of authentication and authorization methods, they easily become the target of the following security threats [19].

- (i) **Malicious Applications:** Due to the integration of third-party applications in the SDN framework, an attacker may run the malicious applications to get the access of the entire network. Similar to the hijacked controller, they have a catastrophic impact on the entire network architecture by exploiting the information that they have collected from the network during deep packet inspection [14, 20]. To prevent this type of attack, the proper authentication mechanism and information access policy are required.
- (ii) **Resource Exhaustion:** The malicious applications send massive amount of requests to the controller to exhaust the bandwidth of northbound API and to consume the computational resources of the controller [5]. To prevent this attack, the controller should restrict the application to access the northbound interface. Each application needs to access the northbound interface for a different purpose. Therefore, the proper access control mechanism is required to limit the access of an interface by an application so that a single application cannot consume all the resources of interface and controller.
- (iii) **Policy Conflicts:** The multiple applications running on the SDN controller insert different security policies that may conflict with each other. The policies conflict occurs when OpenFlow applications instruct the controller to accept or reject the particular packet that is otherwise inversely rejected or accepted by existing OpenFlow applications [14].

4 Conclusion

This study identified the various vulnerable points in the SDN architecture that give birth to various security threats. The lack of TLS adoption on the communication channel makes it vulnerable to eavesdropping and MITM attacks. Moreover, dumb switches and centralized controller make it vulnerable to topology spoofing, controller hijacking, and DDoS attacks. To make it worse, the absence of authentication

and access policy mechanism in the application plane may have catastrophic effects on the SDN architecture. Therefore, there is a need to address these issues to fully deploy the SDN network in an organization.

References

1. Farhady H, Lee HY, Nakao A (2015) Software-defined networking: a survey. *Comput Netw* 81:79–95
2. Open Network Foundation (ONF). <https://www.opennetworking.org/>. Last Accessed 24 June 2020
3. W. Li, W. Meng, L.F. Kwok.: A survey on OpenFlow-based software defined networks: security challenges and countermeasures. *J Netw Comput Appl* 68:126–139 (2016)
4. Antikainen M, Aura T, Särelä M (2014) Spook in your network: attacking an sdn with a compromised openflow switch. In: Nordic conference on secure IT systems, Springer, pp 229–244
5. Dayal N, Maity P, Srivastava S (2016) Research trends in security and DDoS in SDN. *Secur Commun Netw* 9(18):6386–6411
6. Spooner J, Zhu SY (2016) A review of solutions for SDN-exclusive security issues. *Int J Adv Comput Sci Appl* 7(8):113–122
7. The Internet Research Task Force. <https://irtf.org/>. Last Accessed 24 June 2020
8. The Internet Engineering Task Force (IETF). <https://www.ietf.org/>. Last Accessed 24 June 2020
9. Braun W, Menth M (2014) Software-defined networking using OpenFlow: protocols, applications and architectural design choices. *Future Internet* 6(2):302–336
10. Jarraya Y, Madi T, Debbabi M (2014) A survey and a layered taxonomy of software-defined networking. *IEEE Commun Surv Tutorials* 16(4):1955–1980
11. Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T (2014) A survey of software-defined networking: past, present, and future of programmable networks. *IEEE Commun Surv Tutorials* 16(3):1617–1634
12. Gao S, Li Z, Xiao B, Wei G (2018) Security threats in the data plane of software-defined networks. *IEEE Netw* 32(4):108–113
13. Xu T, Gao D, Dong P, Foh CH, Zhang H (2017) Mitigating the table-overflow attack in software-defined networking. *IEEE Trans Netw Serv Manage* 14(4):1086–1097
14. Khan S, Gani A, Wahab AWA, Guizani M, Khan MK (2016) Topology discovery in software defined networks: threats, taxonomy, and state-of-the-art. *IEEE Commun Surv Tutorials* 19(1):303–324
15. Alsmadi I, Xu D (2015) Security of software defined networks. *A Surv Comput Secur* 53:79–108
16. Xu J, Wang L, Xu Z (2020) An enhanced saturation attack and its mitigation mechanism in software-defined networking. *Comput Netw* 169:107092
17. Azzouni A, Trang NTM, Boutaba R (2017) Limitations of openflow topology discovery protocol. In: 2017 16th annual mediterranean Ad hoc networking workshop (Med-Hoc-Net), pp 1–3
18. Kalkan K, Altay L, Gür G, Alagöz F (2018) JESS: joint entropy-based DDoS defense scheme in SDN. *IEEE J Sel Areas Commun* 36(10):2358–2372
19. Scott-Hayward S, O'Callaghan G, Sezer S (2013) SDN security: a survey. In: 2013 IEEE SDN for future networks and services (SDN4FNS), pp 1–7

20. Chica JCC, Imbachi JC, Botero JF (2020) Security in SDN: a comprehensive survey. *J Netw Comput Appl* 159:1–23
21. Pradhan A, Mathew R (2020) Solutions to vulnerabilities and threats in software defined networking (SDN). *Procedia Comput Sci* 171:2581–2589
22. Yue M, Wang H, Liu L, Wu Z (2020) Detecting DoS attacks based on multi-features in SDN. *IEEE Access* 8:104688–104700

Chapter 14

Hardware-Based Analysis of PCG Signal for Heart Conditions



Takhellambam Gautam Meitei, Sinam Ajitkumar Singh,
and Swanirbhar Majumder

1 Introduction

For any animal, the heart plays an important role and any abnormality in it can lead to major health concerns. Hence, early detection of the heart condition in a non-invasive manner can be very efficient and helpful. Our heart produces a sound signal known as phonocardiogram (PCG), during the cardiac cycle. This signal highlights information on the condition of the heart. Predicting the heart condition via auscultations has been followed by doctors over the years, yet it is still insufficient to efficiently diagnose a patient. Automated feature extraction research on PCG is quite new, and hardware implementation using PCG signals is few.

Here the work is progressed as an extension of the work processed in [1]. The authors performed variational mode decomposition (VMD) to deal with the PCG signal and the detected peaks are fed in to an ANN to classify the heart signals and studied [2]. The ANN has three hidden layers, trained via applying the input patterns one after the other until the system is trained with the whole dataset. In this paper, the peak detection process and the trained neural network are taken and a hardware-based system is designed. The hardware system is designed using Xilinx system generator, which is a DSP design tool provided in MATLAB/Simulink, which can produce an environment for FPGA design using only the Xilinx block sets available in the Simulink [3].

T. G. Meitei

Department of Photonics, National Chiao Tung University, Hsinchu, Taiwan

S. A. Singh · S. Majumder (✉)

Department of IT, Tripura University, Agartala, Tripura, India

e-mail: swanirbhar@ieee.org

2 Theory

2.1 Variational Mode Decomposition (VMD)

Konstantin Dragomiretskiy and Dominique Zosso introduced VMD in 2014. [2] Basically, VMD can be said as a process of noise reduction in a signal. This algorithm decomposes a signal to k number of multivariate modes which are densely populated around the center frequency, v_k . The variation of the input signal, $u(t)$ is shown as,

$$\min_{(i_N, v_k)} \left\{ \sum_k \left\| \partial_t \left[\left(\sigma(t) + \frac{j}{\pi t} \right) i_k \right] e^{-jvt} \right\|_2^2 \right\} \quad (1)$$

where each decomposed mode is denoted by, $\sum_k i_k = u$; i_k ($k = 1, 2, 3, \dots L$).

The signal is then reconstructed from the decomposed signal with the least noise. The reconstruction problem is addressed with the introduction of Langrangian multipliers and quadratic terms [3]. The Langrangian argument Γ is expanded as under:

$$\begin{aligned} \Gamma(i_k, V_k, \lambda) = \alpha \sum_k & \left\| \partial_t \left[\left(\sigma(t) + \frac{j}{\pi t} \right) i_k \right] e^{-jwt} \right\|_2^2 \\ & + \left\| u - \sum i_k \right\|_2^2 + (\lambda_1 u - \sum i_k) \end{aligned} \quad (2)$$

where σ is Dirac distribution, λ is actually a dual ascent and α is a Lagrange multiplier. Subsequently, the step i_k^j is update to i_k^{j+1} where j represents the number of iterations and V_n^j is updated version of V_n^{j+1} [4, 4].

2.2 Artificial Neural Network (ANN)

An ANN is a complex interconnection for information processing that mimics the connectivity of human brain to calculate a situation presented. A basic ANN consists of an input layer, one or more hidden layers where each layer has its own weights and biases with activation functions applied on it and one output layer. Each neuron also consists of linear or nonlinear activation functions, and the learning process of the ANN is basically done by changing the weight of the neuron.

The authors in [1] implemented ANN consisting of 1 input layer, 3 hidden layers, and 1 output layer, with nonlinear functions in the hidden layers and a linear function in the output as activation functions [6] (Fig. 1).

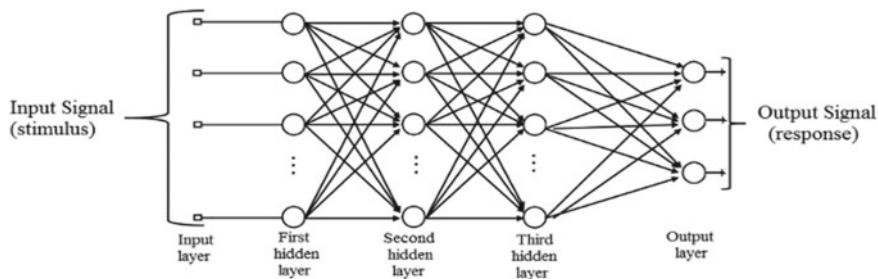


Fig. 1 ANN architecture with three hidden layers

2.3 Xilinx System Generator

Xilinx system generator operates in the Simulink with very limited block sets that can be converted into hardware RTL designs which can further be implemented on FPGA. A few drawbacks are limited number of Xilinx block sets, compatibility issue between MATLAB, Xilinx ISE Version, and Operating System. System generator block offers concept of Mathematics, logic, memories, and DSP function that can be implemented for DSP system designs that work on arbitrary precision fixed-point values and Boolean. The design in a Xilinx blocks sets understands and works within two specific blocks, “The Gateway-in” and “The Gateway-out,” also to activate the block sets a system generator token is placed on the design to set an FPGA boundary (Fig. 2).

Through the system generator token, the conversion of the design into RTL is done. The FPGA boards depending upon the application is selected, and the desired HDL code is generated. The clocking between the designs is taken care of while generating.

3 Methodology

For each PCG signal analysis, the peak detection for S1 and S2 is the most important task. In this paper, the preprocessing steps on the signal are implemented in a software



Fig. 2 FPGA boundary blocks

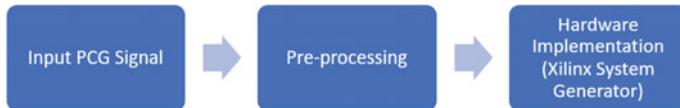


Fig. 3 Step-by-step algorithm

and the post-processing, and the artificial neural network is simulated on Xilinx system generator [7, 7] (Fig. 3).

3.1 Data Collection

The data is obtained from the year 2016s Challenge [9] of physionet.org. About 300 PCG signals were downloaded including both normal and abnormal heart sounds, which is then resampled to 2000 Hz and stored in.wav format.

3.2 Preprocessing

The signals are decimated by a factor of 10 before processing it with VMD. The decimated signal is normalized and decomposed. In [1], the authors implemented the third mode for reconstruction stating that the third mode had the least average noise (Fig. 4).

3.3 Hardware Analysis

The reconstructed signal is taken and fed into the design. The absolute value of the signal is taken and normalized and given to a 1D median filter of window size-3 and the Shannon energy of the signal is thereby calculated in order to obtain the envelope of the PCG signal. Dynamic thresholding is applied to the Shannon energy plot to find the desired peaks. The peaks detected from the dynamic thresholding are fed to the designed ANN for to give a yes or no output, where 1 represents abnormal while 0 represents normal heart sound (Fig. 5).

The normalization is done to provide data integrity, and it does not affect in the abnormality detection. The normalization is realized according to the given equation:

$$\text{norm} = \frac{\text{signal}}{\max(\text{signal})} \quad (3)$$

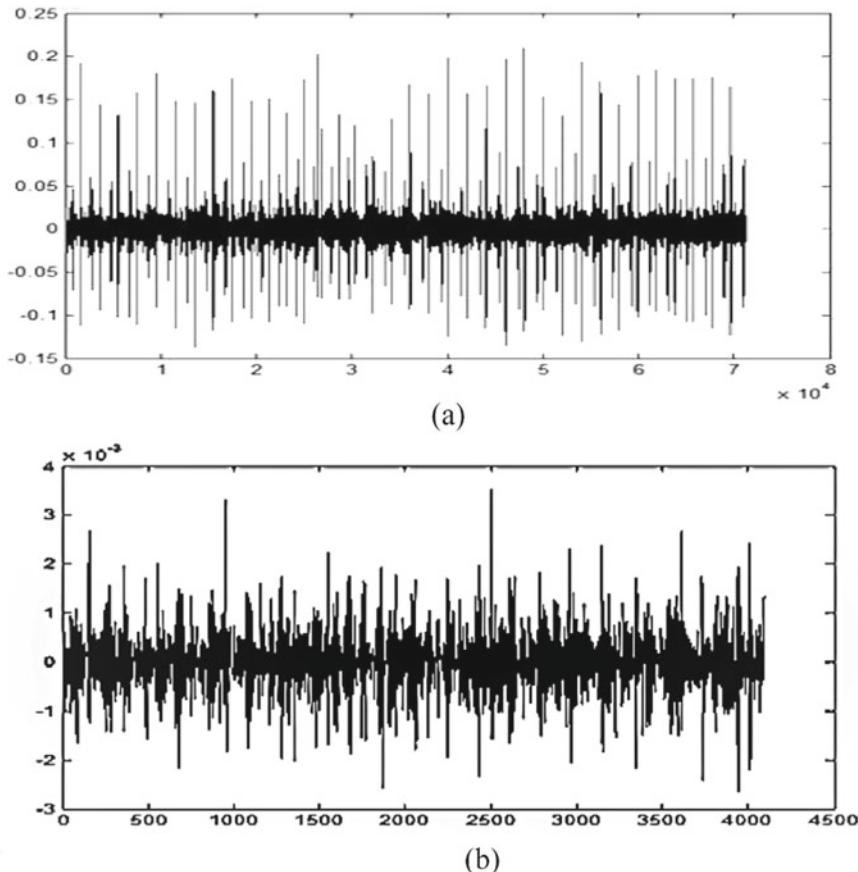


Fig. 4 **a** Preprocessed PCG signal. **b** Reconstructed signal



Fig. 5 Hardware analysis in Xilinx system generator

Here, the *signal* denotes the absolute value of the reconstructed signal. The normalized values are now given to a median filter realization, to remove unwanted spikes or out of range noise from a signal. The 1D median filter is designed to process with a window size 3. The normalized signal is taken three times with the first having two-unit delays and the second with one-unit delay and the third with no delay. The three inputs are compared, and the median output is obtained for each time period. The Shannon energy is then obtained from the median output to detect the PCG envelope as information which is required are carried by the PCG envelope.

The Shannon energy system is realized according to the equation:

$$\text{Sh_en} = \sum \left(x(n)^2 \times \log_2 \frac{1}{x(n)^2} \right) \quad (4)$$

Here, $x(n)$ denotes the signal from the median filter. The authors implemented Shannon energy as energy calculations through Shannon energy attenuated the lower amplitude components as compared to the higher amplitude components while the medium range components get highly emphasized.

The dynamic thresholding is done with a threshold limit of 75% of the maximum value. The dynamic threshold value is set with the prior idea of the fact that the processed signal has higher energy at S1 locations with respect to the S2 locations and other variable components and features like murmurs [1]. The authors in [10] differentiated between S1 and S2 by estimating the time difference between S1 and S2 and neglecting smaller peaks between those time differences except for higher peaks. If two high peaks are encountered between short time differences, the higher altitude is taken while the smaller one is neglected. To determine S2, a prominent peak between two S1 peaks is taken into consideration (Fig. 6).

The peaks obtained from the dynamic thresholding are stored in a.mat file and kept ready for feeding into the ANN. An average of 200 peak for each heart sound is taken and stored and for the heart sound having less than 200 peaks, zero padding done before storing to reach 200.

In [1], the neural network implementation for abnormality analysis is presented, after training the neural network with the Levenberg–Marquardt backpropagation and the best accuracy was obtained using three hidden layers with 80 (30,30,20)

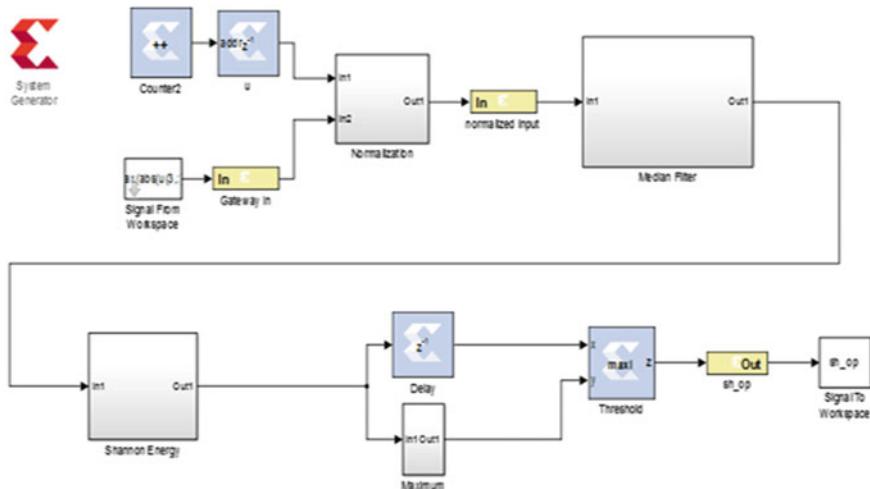


Fig. 6 Post-processing for peak detection

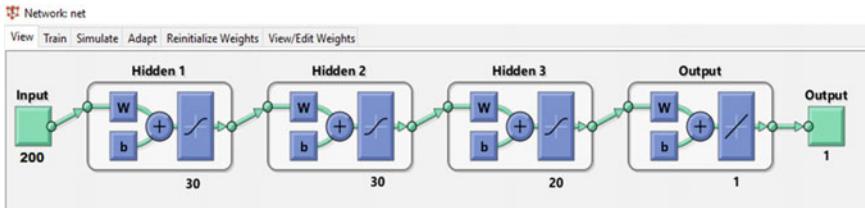


Fig. 7 Trained ANN design obtained from MATLAB

neurons. So, a hardware base design depending upon the neural network used is prepared. The weights and biases are obtained from the neural network simulations (Fig. 7).

A basic neuron design is implemented from the given formula, using the *AddSub* and *CMult* blocks with a *Constant* block as bias.

$$a = f(\sum w p + b) \quad (5)$$

where *p* is the input, *w* is the weight of the neuron, and *b* is the biasing constant (Fig. 8).

The activation function implemented in the first, second, and the third layers is the tanh function while the output neuron is implemented with a linear activation function. The tanh activation function is implemented using the Maclaurin series of the Sigmoid function [11, 11]. The relation between tanh function and sigmoid function is given by,

$$\tanh(x) = 2\sigma(2x) - 1 \quad (6)$$

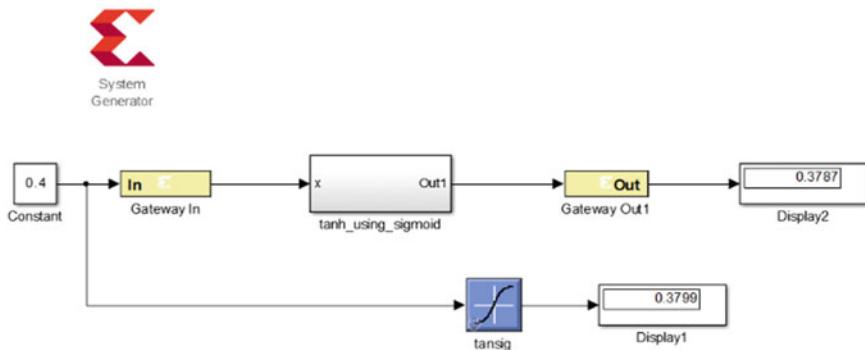


Fig. 8 Comparison of the designed tanh function with a MATLAB function

where $\sigma(x)$ is the sigmoid function, and the activation function is realized from the obtained output.

$$\tanh(x) = 2(0.5 + 0.25(2x) - 0.02083(2x)^3 + 0.00208(2x^5)) - 1 \quad (7)$$

The designed tanh function is also tested with a MATLAB tanh function.

4 Result and Discussion

The designed ANN is tested for 55 known random heart sounds, 29 abnormal and 26 normal heart sounds. The ANN provided an accurate result for 35 heart sounds out of which 22 gave false negative and 13 gave true positive. The remaining 20 heart sound gave wrong predictions of which 7 gave false negative and 13 gave true negative. The ANN was designed with three hidden layers and 80 (30,30,20) neurons. Activation function tanh was implemented in the hidden layers while a linear function was used for the output layer. False positive, meaning the abnormal PCG is detected as abnormal, and false negative meaning the abnormal sound is detected as normal, and vice versa for true positive and true negative. The problem arises when an abnormal heart is detected as normal, i.e., in the case of false negative. The overall accuracy of the designed ANN gave about 63.63% accuracy. The false negative prediction is about 12.73%, which can still be improved by implementing different approaches and steps [13]. Less number of accuracies in the hardware design might be due to difference in operation using floating point and fixed-point precisions in software and hardware, respectively, and also, the approximation of tanh activation function implemented can cause small errors.

5 Conclusion

In this paper, the abnormality detection is implemented using both software and hardware simulated together using MATLAB and Xilinx system generator [14]. The signals are decomposed via VMD in MATLAB and with the help of Xilinx system generator the decomposed signal is analyzed to find the S1 and S2 peaks with the help of median filtering and Shannon energy calculations. The detected peaks are then stored for further testing with ANN. It was tested for 55 heart sounds and showed an accuracy of 63.63%. The hardware design still shows lesser accuracy as compared to the software implementation of the ANN, as the values operating in the design differs as one works on floating point precision and other works on fixed-point precision values. In the future, the design can still be improved with an optimized design with proper implementation of activation function in the ANN. A few studies also suggested that the normalized average Shannon energy was sensitive to heart murmurs which can lead to false peak detection. Hence, implementing a third-order

Shannon energy equation can be helpful for better calculation. Future application of the hardware analysis can be of huge help in the applications of medical system designs that require understanding a patient's heart conditions, for example, in an intensive care unit (ICU).

Acknowledgements The authors would like to acknowledge project number BT/COE/34/SP28408/2018 of Department of Biotechnology (DBT), Government of India for the financial support via the NECBH Twinning grant with Dr. C. N. Gupta, of Department of BSBE, IIT Guwahati.

References

1. Singh SA, Verma A, Chhetry S, Majumder S (2017) Abnormality analysis of PCG Signal using VMD and MLP neural network. In: 7th international symposium on embedded computing and system design (ISED), IEEE, India
2. Dragomiretskiy K, Zosso D (2014, February) Variational mode decomposition. In: IEEE transaction on signal processing, vol 62, issue no 3. IEEE, , pp 531–544
3. Singh SA, Meitei TG, Majumder S (2020) Short PCG classification based on deep learning. In: Deep learning techniques for biomedical and health informatics. Elsevier Academic Press. <https://doi.org/10.1016/B978-0-12-819061-6.00006-9>
4. Singh SA, Majumder S (2020) Short unsegmented PCG classification based on ensemble classifier. Turk J Elec Eng Comp Sci 28: 875–889. Copyright © 2020 <https://doi.org/10.3906/elk-1905-165>. E-ISSN: 1303-6203, ISSN: 1300-0632
5. Beritelli F, Serrano S (2007) Biometric Identification based on frequency analysis of cardiac sounds. IEEE Trans Inf Forensics Secur 2(3): 596–604 (Italy)
6. Mert A (2016) ECG feature extraction based on the bandwidth properties of variational mode decomposition. Physiol Meas 37(4):530–543 (Turkey)
7. Phua K, Chen J, Dat TH, Shue L (2007) Heart sound as a biometric. Pattern Recogn 41(3):906–919 (2007). (Singapore)
8. Banerjee S, Mishra M, Mukherjee A (2016) Segmentation and detection of first and second heart sounds (S1 and S2) using variational mode decomposition. IEEE EMBS conference on biomedical engineering and science (IECBES). IEEE, India, pp 565–570
9. PhysioNet Homepage. <https://physionet.org/physiobank/database/challenge/2016>. last accessed 2017/05/24
10. Temurtas F, Gulbag A, Yumusak N (2004) A study on neural network using Taylor series expansion of sigmoid activation function. In: Laganà A et al (eds) ICCSA 2004, LNCS 3046, pp 389–397. © Springer, Berlin, Heidelberg
11. Choi S, Jiang Z (2008) Comparison of envelope extraction algorithms for cardiac sound signal segmentation. Exp Syst Appl 34(2):1056–1069
12. Maji U, Mitra M, Pal S (2017) Characterization of cardiac arrhythmias by VMD technique. Biocybern Biomed Eng 37(3):578–589
13. Saini M (2016) Proposed algorithm for implementation of Shannon energy envelope for heart sound analysis. Int J Electron Commun Eng Commun Technol 7(1). ISSN: 2230-7109
14. Sulaiman N, Obaid ZA, Marhaban MH, Hamidon MN (2009) Design and implementation of FPGA-based systems—a review. Australian J Basic Appl Sci 3(4), 3575–3596, (2009) ISSN 1991–8178 © 2009. INSInet Publication, Australia

Chapter 15

Elliptic Curve Cryptography: A Software Implementation



Sumit Singh Dhanda, Brahmjit Singh, and Poonam Jindal

1 Introduction

Provisioning of the security to information flow in wireless networks has been one of the most important and critical requirements. This has become more difficult and challenging in the resource constrained applications like Internet of things. These setups are limited in terms of computing power, storage, and energy. RSA algorithm has been widely employed in commercial applications for so long. However, the key size of RSA is quite large in context with the emerging applications. This makes the RSA unsuitable for resource constrained environment. Elliptic curve cryptography (ECC) has been proposed as a potential and promising alternative to RSA.

The ECC has come a long way since it was first used for the purpose of security by Miller [1] and Koblitz [2] independently. Initially, ECC was used for the purpose of securing Internet. But researchers have put great efforts to adapt the ECC to secure wireless sensor networks (WSNs). ECC being an asymmetric method was compared to RSA cipher. It was able to provide same level of security with much smaller key sizes in comparison with RSA, e.g., for the 80-bit security, ECC requires key of 160 bits and RSA will utilize a key of 1024 bits [3]. It is second popular choice for security solutions after advanced encryption scheme (AES). Symmetric key cryptographic techniques outperform their asymmetric counterparts in terms of key sizes and execution speed. This makes it necessary for the researchers to improve the implementation speeds of the asymmetric ciphers continuously.

S. S. Dhanda (✉) · B. Singh
National Institute of Technology, Kurukshetra 136119, India
e-mail: poonamjindal81@nitkkr.ac.in

P. Jindal
Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra 136119, India

Security of elliptic curve cryptosystems lies in the hardness of elliptic curve discrete logarithm problem (ECDLP) which is increased by involving large exponentiation. This is achieved using a very large prime number as the order of the field [4]. It is the basis of many security solutions from IPSec to blockchain technologies [5].

There are mainly two type of implementation of elliptic curves: software implementation and hardware implementation. Each type of implementation is favored by a specific type of curve. Prime curves over finite fields are chosen for the software implementation while binary curves are selected for the hardware implementation in ECC [6–9]. Some new curves have also been proposed to avoid backdoor traps [10]. It improves the safety of the curves by using a rigid deterministic process. Finally, the implementation is carried out using a protocol like elliptic curve Diffie–Hellman (ECDH) or elliptic curve digital signature algorithm (ECDSA) [11]. These protocols also include the key exchange for the security.

In this paper, we present software implementation of ECC based on the Weierstrass curve to show its efficacy for the intended applications. The implementation is carried out in C++ language on an HP laptop with 64-bit dual core i-5 processor and DevCpp as the IDE.

1.1 Organization of the Paper

The organization of the rest of the paper is as follows: Mathematical background and procedures are explained in the Sect. 2. Implementation details of ECC are discussed in Sect. 3. Numerical results are analyzed in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Elliptic Curves Mathematical Background

An elliptic curve is a non-singular curve defined over a finite field [6]. It is a relation that maps the q (points on x -axis) with p (points on y -axis) as per the equation below:

$$p^2 \bmod k = q^3 + aq + b \bmod k \quad (1)$$

This equation is called as Weierstrass equation. Squaring operation on the left-hand side of the equation introduces symmetry in the equation. This curve must not follow the condition

$$4a^3 + 27b^2 \equiv 0 \bmod k \quad (2)$$

It is the condition for non-singular equations of characteristic more than 3, these equations have multiple root which are not suitable for the cryptographic applications. This equation is defined over the field Z_k ; i.e., it is a prime curve defined over a finite

field Z_k . k is the order of the field which informs about the number of points in the field. All these points lie in field Z_k .

To find a new point on the curve with the help of two existing points X and Y , two methods are used. Tangent method is based on the use of co-ordinate geometry. There are two operations that are used under two different cases. First, point addition which is used $X \neq Y$ while point doubling is used when $X = Y$.

$$Z = X + Y \text{ (point addition)}$$

Case 1: $X \neq Y$ Here, point addition is used to find a new point on the curve. As shown in Eq. (2), co-ordinates are used to find the slope ‘ L ’ of the line which is formed by joining X and Y .

$$L = \frac{q_x - q_y}{p_x - p_y} \quad (2)$$

This slope is then used with available points to find the two co-ordinates of the new points as illustrated in Eqs. (3) and (4) below

$$p_z = (L^2 - p_x - p_y) \bmod k \quad (3)$$

$$q_z = -q_x + L(p_x - p_y) \bmod k \quad (4)$$

Case 2: $X = Y$ Point doubling is used to find out the unknown third point when the two points are same. It is also known as tangent method. Equations (5), (6), and (7) are used to find the slope of the tangent and co-ordinates of the unknown point.

$$L = (3 * p_x * p_x + a) * (2 * q_x)^{-1} \bmod k \quad (5)$$

$$p_z = L^2 - 2 * p_x \bmod k \quad (6)$$

$$q_z = -q_x + L(p_x - p_y) \bmod k \quad (7)$$

These equations replace the graphical method. Apart from the above-mentioned point addition, point doubling and inversion operations used in Eqs. (2) and (5). One more operation that is used prominently in ECC is scalar multiplication. If one wants to calculate $m * X$ then consecutive m additions of X will be performed, i.e.,

$$m * X = X + X \dots + X \text{ ($m - times$)}$$

One can also use point additions and point doublings to calculate the same.

$$m * X = X + \dots + (2(2(\dots X + 2(X + 2(X + 2x)))))$$

Different optimization algorithms are utilized to speed up the elliptic curve scalar multiplication [9, 10].

3 Implementation Details

This is very basic implementation of ECC that is carried out on 64-bit dual core processor. The curve used is shown in Eq. (1). Firstly, the points are calculated that lies on the elliptic curve. Then, we create functions for calculating extended Euclidian algorithm, point addition, point doubling, modulus reduction, encryption, and decryption.

Modulus function ensures that the calculation that one do remains within the finite field. Apart from above functions, we have used elliptic curve Diffie–Hellman algorithm for the process.

- (i) Sender and receiver agree on a prime number k which decides the size of the field and elliptic curve parameters a and b of Eq. (1) which defines the elliptic curve.
- (ii) From the finite field Z_k a base point K is decided which can value between 1 and $k - 1$.
- (iii) Two large integers' m and n are chosen from 1 to $k - 1$ as private key of sender and receiver, respectively. These are used to generate the public key for sender (U_s) and receiver (U_r) as $m * K$ and $n * K$.
- (iv) These public keys are exchanged between sender and receiver to calculate the secret keys. ($SK = n * m * K = m * n * K$)
- (v) Then a message point is chosen from the points on elliptic curve $X(pX, qX)$. And $C = X + m * U_s$ is calculated as the encrypted message. $T(m * K, C)$ is the transmitted message from sender to receiver.
- (vi) On receiving $T(m * K, C)$, receiver calculates secret key and use it to decrypt the received message as follows:

$$\begin{aligned}
 C + (-n) * (m * K) &= C - n * m * K \\
 &= M + m * U_r - n * m * K \\
 &= M + m * n * K - n * m * K \\
 &= M
 \end{aligned}$$

4 Results and Discussion

The first and most important thing to do is to know whether a point lies on the curve or not for this we write a simple function to identify the point and store it in a container.

Figure 2 shows the point on the curve shown by Eq. (1) for $k = 23$, $a = 1$, $b = 0$ (Fig. 1).

Figure 2 shows the complete details of the encrypted and decrypted output for the curve in Eq. (1) with the above said parameters. These two results demonstrate how to carry out the encryption and decryption with elliptic curves. It also demonstrates

```
C:\Users\brewberry\Documents\Dpp\ecc2020imp.exe
put a prime number:23
Put a value of a:1
Put a value of b:0
-----
Points on Elliptic Curve
-----
(0, 0)
(1, 5)
(1, 18)
(9, 5)
(9, 18)
(11, 10)
(11, 13)
(13, 5)
(13, 18)
(15, 3)
(15, 20)
(16, 8)
(16, 15)
(17, 9)
(17, 13)
(18, 10)
(18, 13)
(19, 1)
(19, 22)
(20, 4)
(20, 19)
(21, 6)
(21, 17)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
Select a base point (x,y) from the curve:
```

Fig. 1 Calculating points on the curve

```
C:\Users\brewberry\Documents\Dpp\ecc2020imp.exe
put a prime number:23
Put a value of a:1
Put a value of b:0
-----
Points on Elliptic Curve
-----
(0, 0)
(1, 5)
(1, 18)
(9, 5)
(9, 18)
(11, 10)
(11, 13)
(13, 5)
(13, 18)
(15, 3)
(15, 20)
(16, 8)
(16, 15)
(17, 9)
(17, 13)
(18, 10)
(18, 13)
(19, 1)
(19, 22)
(20, 4)
(20, 19)
(21, 6)
(21, 17)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
Select a base point (x,y) from the curve: 17
0
select a private key for Alice: 13
public key of Alice is (19,1)
select a private key for Bob: 7
public key of Bob is (11,10)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
Encryption/Decryption
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
Select a Message point (x,y) from the curve(for encryption): 13
0
Cipher is (20,4)
Alice send message pair((19,1),(20,4))
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -
Bob receive the message and start decrypting
Decrypted message is (19,18)
```

Fig. 2 Encryption and decryption output of ECC

that if a malicious user is able to get its hand on the public key of the sender even then he will not be able to extract the information from it.

5 Conclusion

In this paper, we have implemented the ECC in simulation environment. In the present realization, ECC is implemented over a prime field. It is observed that the field size affects the security of the curve. Security of ECC is a function of ECDLP. Order of the field has a direct effect on the security of the field, by using a very large prime number field security can be increased. Point addition, point doubling, scalar multiplication, and inversion are important operations in ECC implementation. Execution speed can be improved by optimizing scalar multiplication and inversion. Finally, ECC is evaluated using a protocol like ECDH or ECDSA. ECC is envisaged to offer excellent security level against the threats and may be the preferred choice for numerous solutions ranging from IPSec to blockchain technologies. In the future, we will implement this encryption on 8-bit and 16-bit device and will try to optimize for the execution speed.

References

1. Miller VS (1985) Use of elliptic curves in cryptography. CRYPTO 1985
2. Koblitz N (1987) Elliptic curve cryptosystems. Math Comput. van Leeuwen J (ed) (1995) Computer science today. Recent trends and developments. Lecture notes in computer science, vol 1000. Springer, Berlin Heidelberg, New York
3. Dhanda SS, Singh B, Jindal P (2020) Demystifyingelliptic curve cryptography: curve selection, implementation and countermeasures to attacks. J Interdisc Math 23(2):463–470. <https://doi.org/10.1080/09720502.2020.1731959>
4. Dhanda SS, Singh B (2020) Jindal P (2020), Lightweight cryptography: a solution to secure IoT. Wirel Pers Commun 112:1947–1980. <https://doi.org/10.1007/s11277-020-07134-3>
5. Dhanda SS, Singh B, Jindal P (2020) IoT security: a comprehensive view. Peng S-L et al (eds) Principles of Internet of Things (IoT) ecosystem: insight paradigm, intelligent systems reference library, vol 174. https://doi.org/https://doi.org/10.1007/978-3-030-33596-0_19
6. Bernstein DJ, Birkner P, Joye M, Lange T, Peters C (2008) Twisted edwards curves. Progress in cryptology. Springer, Berlin, Germany, pp 389–405
7. Montgomery PL (1987) Speeding the Pollard and elliptic curve methods of factorization. Math Comput 48(177):243–264
8. Bos JW, Costello C, Longa P et al (2016) (2016) Selecting elliptic curves for cryptography: an efficiency and security analysis. J Cryptographic Eng 6:259–286. <https://doi.org/10.1007/s13389-015-0097-y>
9. Liu Z, Huang X, Hu Z, Khan MK, Seo H, Lu Z (2017) On emerging family of elliptic curves to secure Internet of Things: ECC comes of age. IEEE Trans Dependable Secure Comput 14(3):237–248
10. Liu Z, Seo H (2019) IoT NUMS: evaluating NUMS elliptic curve cryptography for IoT platforms. IEEE Trans Inf Forensics Secur 14(3)
11. Mehibel N, Hamadouche M (2020) A new enhancement of ellipticcurve digital signature algorithm. J Discrete Math Sci Crypt. <https://doi.org/10.1080/09720529.2019.1615673>

Chapter 16

When Distributed Ledger Technology Meets Traditional Payment Systems—Benefits and Challenges



Saurabh Jain, Adarsh Shukla, and Kashish Srivastava

1 Introduction

At present, everyone is dependent on online methods for every possible work. One of the examples can be taken as online shopping. The reason is the absence of time in the busy lifestyle of people. By shopping online, people can get everything they need at their place by staying in their position. However, like everything, online shopping has many benefits as well as disadvantages or we can say legitimate risks. When the user login to any web application for purchase, he/she gives their input and these credentials go to the server. The server authenticates them. This is where the threat of payment system attack comes into rolls.

Payments systems are built conventionally and are vulnerable to several kinds of threats. The credentials are provided by the user and reach the server indirectly following the malicious path. These are captured by an attacker in the middle itself and the user's credentials are passed to the attacker. The attacker secures the credentials and allows the user to access the web application. In this way, many times the user do not know that the attack has occurred. After much research, to eliminate the possibility of an attack, the authors have found an optimized solution using blockchain technology [1–3].

This paper is organized as follows: Introduction section is described as the payment system and needs for a better payment system that the conventional being followed. Section 2 reviews the blockchain technology and types of blockchains present in the system; Sect. 3 focuses on the working of consensus algorithms and their variations in traditional systems. Section 4 reviews the real-world problems and how the proposed model reduces the vulnerability of various attacks and providing

S. Jain (✉) · A. Shukla · K. Srivastava

School of Computer Science, University of Petroleum and Energy Studies, Bidholi, Dehradun, India

e-mail: saurabh.jain@ddn.upes.ac.in

security better than traditional systems. Section 5 discusses the challenges faced by traditional payment systems and solutions provided by blockchain technology for them. In the next section, this paper presents a blockchain technology-based payment system application advantages. The last section of this paper defines the conclusion and future work.

2 Blockchain Technology

As the name “Blockchain” suggests it is a combination of two words “Block” that can be also said as units in general explanation and “Chain” meant as linking, “Blockchain” is a kind of shared or distributed ledger which contains transactional records without any singular authorization of an entity. In a generic explanation, this work can also say that it is a kind of database that supports reading and appending based on transactions. It ensures security with the help of cryptographic hashing, translucency, and decentralization [4–6]. It records and stores all the transactions occurring on a network by eradicating the need for “trusted” external parties such as payment processors. Referring to blockchain as a trust machine is like trusting innovation in a cynical world. It acts as an evolution with no third-party validation in any exchange [7, 8].

Due to decentralization, a peer-to-peer network operates without the need for trusted intermediaries or authorities. The nodes of the external users in network 2 are then verified the transactions by the computed rules of the system to make sure everything within the system is valid before it gets executed. The affirmation is essential because all transactions and records in a blockchain are unchangeable or immutable. The elegance of the blockchain is that it obviates the need for a central authority to verify trust and the transfer of value [9]. It transfers power and control from large entities to the many, enabling safe, fast, cheaper transactions even though we may not know the entities we are dealing with (Fig. 1).

2.1 Types of Blockchain Networks

Essentially, there are two types of blockchain: public blockchain and private blockchain. Alternatively too, as consortium and hybrid blockchain. Blockchain is differentiated due to their different uses in different industries. So understanding all the types in order: Public blockchain is the type that will be accessible to everyone without any restriction of the participant (authorized/unauthorized). No control is kept over the network, hence ensures security and immutability as no individual could make changes in the blockchain. Private blockchain is a kind of blockchain that requires permissions for access and to participate in the network for validation. These permissions are authorized by the blockchain developers while creating the blockchain. They are used to store-sensitive information and only available to the

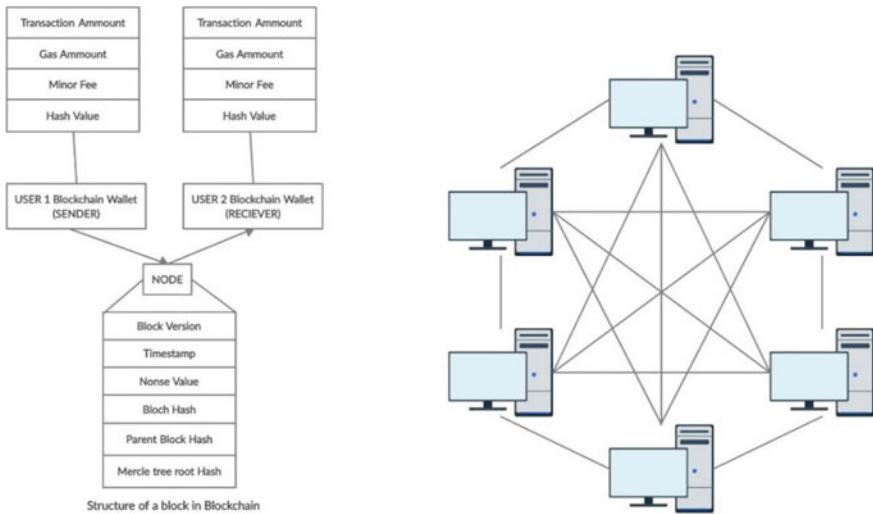


Fig. 1 Working of blockchain network

group of people present within an organization. Consortium blockchain can be also referred to as a subdivided part of private blockchain. The difference lies in the authorization of the blockchain as it is handled or governed by a group of people rather than a single entity. They are quite efficient collectively in collaborating with the business of a kind. Hybrid blockchain is a combination of the benefits received by the public blockchain-based on transparency and private blockchain-based on privacy or security maintained. It is also referred to as the multi-chain network of blockchains. This provides flexibility in the business to segregate the data by transparency and privacy [10].

3 Consensus Algorithms

The consensus blockchain is mainly about a familiar agreement state within the distributed ledger to establish trust amongst all the peers in the shared computing environment. The blockchain consensus protocol has some specific criteria such as understanding an agreement, co-operation, collaboration, inclusion, an equal process to every node and each node is mandatory to participate in the consensus process. This paper would like to elaborate on the strengths and weaknesses of traditional and blockchain payment systems. With respect to traditional systems, blockchain will help in expanding the provision of financial services in less or under-developed areas [11–14]. If a condition is designed for international transactions, traditional money wire methods would increase the complexity of the transactions.

Consensus algorithms are algorithms that ensure the security and integrity of data on distributed systems and processes. The consensus algorithm provides a community to decide whether the transaction is authenticated and not authenticated. The same idea about the transactions made around the agreement. Consensus algorithms prevent double-spending problems and make transactions in the blockchain more reliable. The goal of consensus algorithms is to encourage mutual agreement, promote economic incentive, ensure equity, and fault-tolerant blockchain mechanism.

Steps involved in consensus algorithm:

- User mines a block in the blockchain
- Predefined interfaces will be implemented by consensus algorithms such as PoW and PoS
- These algorithms help in calculating the timestamp of a particular block
- Along with the time stamp, the hash value is also calculated by the algorithms
- Check whether the blockchain is valid or not of the mined blocks.

4 Blockchain Technology and Payment Systems Interplay

Traditional payment systems are centralized and hence are dependent on a central entity. If the attack is made on the central entity, the whole network gets affected. Whereas in blockchain this problem is solved due to its decentralized property. Also, blockchain technology has the potential to lower the costs of security, auditability, and governance significantly. Blockchain systems may offer services at a fine price level than the traditional payment systems. They are suitable solutions for corporate organizations and could potentially generate a significant shareholder value to traditional systems. Understanding a situation based on international transactions the complexity would be much higher. Blockchain technology requires high computational resources and might be slower when compared but are secure more along with outperforming incumbents.

Blockchain is recognized as an innovation to secure administration and remote trade, presented as money related wrong doings as a rule. It is one of the prime reasons why money related firms should select their blockchain applications for the administration of advancement organizations. In this work, when any transaction is added. They are checked that either they are validated or rejected after addition into the system in the pool of all unconfirmed transactions. Within the pool of transactions, a set of them is chosen for the block. The next step of the blockchain-based model is to apply any of the consensus algorithms such as proof of work. At this moment, miners would come into the role. They are paid fees for security, validation, execution of the smart contracts as well. The solved block by miners is broadcasted and verified. Now, a new block is added into the chain (blockchain) after confirmation and transaction confirmed between sender and receiver. If a participant node tampers with a block, its results will be reflected in changing of hash, mismatch of hash values, the local chain of node rendered in an invalid state. Safe payments are made by blockchain technology. Hashing and asymmetric key encryption are used for securing the chain

and for efficient validation and verification. Mostly we use digital signatures by elliptic curve cryptography in blockchain networks. A transaction for transferring assets will be authorized, non-repudiable, and unmodifiable. They are first examined by the digital signing process and then apply to that transaction. Digital signatures confirm that data is hashed and encrypted. In a blockchain's block, first computing the state root hash, transaction root hash, and then receipt root hash are shown at the bottom of the block header. These roots and all the other entities in the header are combined in a hash together with the variable nodes to solve the proof of work puzzle.

Most people have many types of cards like credit cards and debit cards. They can use it to pay for things. But some also have cryptocurrencies such as bitcoin, Ethereum at their disposal. There are many advantages of using blockchain ledger in a transaction in place of a credit/debit card or other online fund transfer options. Key benefits of using blockchain are decentralization (blockchain is a decentralized ledger) trust (in a blockchain, trust level amongst stakeholders is high) and security (every transaction in the blockchain is verified by all the members of the network which restricts manipulation and improves security). Blockchain also helps in smart contracts because it gives the facility of the computer code, storage of any type of digital information. Blockchain accelerates the process of funds clearing and settlement. Blockchain helps in the process of syndicating the loans. Usually, syndicating loans take an average of 19 days for banks to complete the process, but blockchain reduces the time by reducing the intermediate steps or processes. The blockchain provides a secure payment system. Chances of operational and financial fraud risks are very less using blockchain technology (Fig. 2).

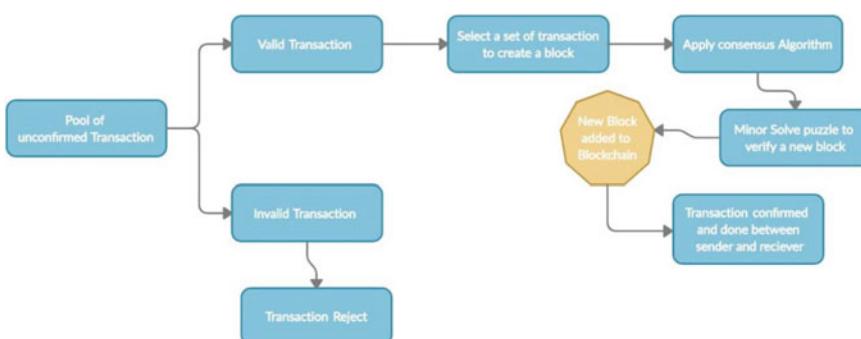


Fig. 2 Blockchain transactions authentication and validation

5 Challenges in Traditional Payment System and Its Countermeasures

Many attacks are possible in online web applications. Now, dealing with them is a very big challenge. The main attacks/challenges are XSS, broken authentication, security misconfiguration, sensitive data exposure, or man-in-the-middle attack. Now discuss XSS (cross-site scripting) attack, this attack is also-ally done on websites that receive data from the user. With an XSS attack, attackers can steal cookies, session tokens, and other sensitive information from the user's browser. Our paper provides challenges and countermeasures to many security issues like in the current payment system, such as integrity, privacy, auditing, transparency, etc., explores various issues caused by various web attacks in transactions XSS attacks and man-in-the-middle attacks etc. (Table [1]). These are OWASP real-time trending attacks [15–17].

6 Blockchain Technology-Based Payment Applications

Blockchain being a peer-to-peer as well as a decentralized network can have a wide range of usability in certain organizations by industries. Applications of blockchain are present in various methods, thus describing it in payment-based applications. Many people and organizations can complete successful transactions without acknowledging the identity of each other due to blockchain. In Fig. 3, it can be observed that the employees of the banks, staff and other workers from the hospitals, IoT device users, supermarket or mall maintenance people can do transactions using a single blockchain according to the requirements of their system. For every pool of valid transactions, a blockchain block will be created. In a blockchain-based decentralized payment system, any person can check the transaction detail at any time when they want which is not possible centralized payment system. Maintaining privacy as well as integrity along with security via hashing is improving standards of blockchain technology in the modern upcoming area and along with other technologies such as IoT, security, and more.

7 Conclusion and Future Work

This paper concludes by the challenges, comparison, and applications, advantages that blockchain technology will help get rid of the traditional payment systems for securing online payments. This can be implemented in all areas or fields for secure online payments or transactions over a vast network. At the current time, 77% of finance-related companies plan to use blockchain in near future, indicating that many are happy with what they can do. At present, this innovation requires some work to integrate effectively. Co-operation, scalability, and energy use are just a few couples

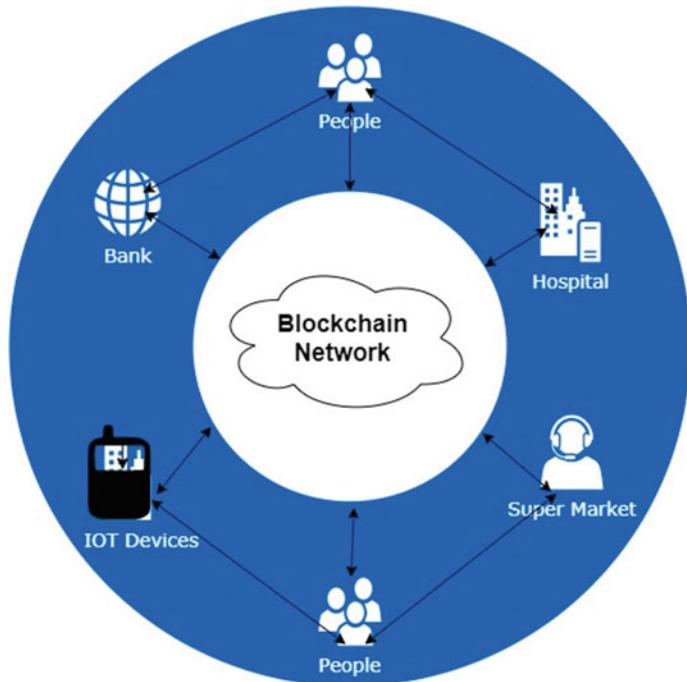


Fig. 3 Applications of blockchain-based payment system

of examples of specialized foundations that should be defeated to get compelling outcomes with the help of blockchain from the square. Blockchain has progressed significantly in a brief timeframe. All the organizations that implement payment options in online mode can move to blockchain technology for a scalable and secure platform orientation.

Table 1 Challenges and solutions

Challenges faced by traditional approach in payment systems	Solutions provided by blockchain technology in payment systems
The integrity of original transaction records is not publicly visible within the central financial institution	In a proposed public blockchain, all transaction records are open and transparent; therefore, all the nodes involved in the storage of transaction block data can review the present transaction data. Hence, integrity of each transaction record is maintained over the system [18]
If the central entity of the traditional payment system fails or gets hampered, all the other set of entities connected to the system will stop their working the instant	Failure of any entity does not affect the normal operation of the blockchain network due to its decentralized property. For blockchain network, people do not need to operate, manage, and maintain manually at any time [19]
Storage for traditional payment systems is limited and hence increases the chances for cyber-attacks	Reducing the risk of cyber-attacks because the storage of blockchain is distributed [20]
Data concerning the central-based financial system is at stake to be modified and deleted within the system	Blockchain network transaction data cannot be modified and deleted because of the concepts of hashing [21]
Cross-border payments are difficult in traditional systems for payments	Blockchain can improve cross-border payments by offering added security, higher transfer speed, and lower conversion fees [21]
Verification for KYC as well speed for transactions is dependent on entities	Blockchain can speed up the accounts payable and receivable process with its immediate ledger update and accuracy of the information, especially for insurance companies and vendors along with verification
For larger companies, it is difficult to keep track of numerous business deals and accounts financially	Blockchain is a distributed peer-to-peer network approach. All transactions are in a centralized system maintained by a single server; but in blockchain, each peer can view the transactions within the network. Thus, large organizations make it easy to track transactions [22]
Traditional payment systems do not comprise of consensus algorithms	The consensus mechanism is a key feature of the blockchain to improve the overall robustness and integrity of shared ledgers. The consensus mechanism among network participants is a prerequisite to validating new blocks of data and mitigates the possibility that a hacker or one or more compromised network participants can corrupt or manipulate a particular ledger [23]

(continued)

Table 1 (continued)

Challenges faced by traditional approach in payment systems	Solutions provided by blockchain technology in payment systems
Privacy of the sender and receiver is not maintained	Blockchain-based payment systems do not have their account numbers or names only hash values are available that cannot be idealized recovered; hence, privacy is maintained [24]
Traditional financial transactions are impossible without intermediaries like banks. Banks are the central link that makes sure that the money being transferred will get to the predetermined recipient. After the transaction, the only wish for a sender is to wait for the recipient	With a blockchain, it is possible to avoid interference in many cases. One can send their digital payments from their virtual wallet to a recipient's virtual wallet with the help of a set of digital keys. For performing such transactions, the address of the recipient must be known. Such transactions are quick, secure, and irreversible, which makes them more advanced than the traditional systems [24]
Auditing of transactions and records is not possible in conventional systems	Blockchain structure is beneficial for real-time audits and making it secure from modifications of any kind
The traditional systems cannot get rid of payment delays and the time-consuming procedures of the outdated payment system	With a blockchain, instant, secure transactions can be an affordable, implementable, and unique alternative for many business ventures/companies [25]

References

1. McAndrews J (1997) Network issues and payment systems. *Federal Reserve Bank of Philadelphia Business Review*
2. Agarwal S, Khapra M, Menezes B, Uchat N (2007, December) Security issues in mobile payment systems. In: Proceedings of ICEG 2007: the 5th international conference on E-Governance, pp 142–152
3. Nigam V, Jain S, Burse K (2014, April) Profile based scheme against DDoS attack in WSN. In 2014 Fourth international conference on communication systems and network technologies (pp. 112-116). IEEE
4. Crosby M, Pattanayak P, Verma S, Kalyanaraman V (2016) Blockchain technology: beyond bitcoin. *Appl Innov* 2(6–10):71
5. Gupta SS (2017) *Blockchain*. Wiley
6. Risius M, Spohrer K (2017) A blockchain research framework. *Bus Inf Syst Eng* 59(6):385–4093
7. Zhang Y, Deng RH, Liu X, Zheng D (2018) Blockchain-based efficient and robust fair payment for outsourcing services in cloud computing. *Inf Sci* 462:262–277
8. Khalil R, Gervais A (2017, October) Revive: rebalancing off-blockchain payment networks. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp 439–453
9. Ølnes S, Ubacht J, Janssen M (2017) Blockchain in Government: benefits and implications of distributed ledger technology for information sharing
10. Chan PMW, Lee JJS, Haldenby PAJ (2019) U.S. Patent No. 10,282,711. U.S. Patent and Trademark Office, Washington, DC

11. Nguyen GT, Kim K (2018) A survey about consensus algorithms used in blockchain. *J Inf Process Syst* 14(1)
12. Kumar A, Jain S (2019) Proof of game (PoG): A game theory based consensus model. In: International Conference on Sustainable Communication Networks and Application (pp. 755–764). Springer, Cham
13. Zoican S, Vochin M, Zoican R, Galatchi D (2018, November) Blockchain and consensus algorithms in the Internet of Things. In: 2018 international symposium on electronics and telecommunications (ISETC). IEEE, pp 1–4
14. Gramoli V (2020) From blockchain consensus back to byzantine consensus. *Future Gener Comput Syst* 107:760–769
15. Kirda E, Kruegel C, Vigna G, Jovanovic N (2006, April) Noxes: a client-side solution for mitigating cross-site scripting attacks. In: Proceedings of the 2006 ACM symposium on applied computing, pp 330–337
16. Bisht P, Venkatakrishnan VN (2008, July) XSS-GUARD: precise dynamic prevention of cross-site scripting attacks. In: International conference on detection of intrusions and malware, and vulnerability assessment. Springer, Berlin, Heidelberg, pp 23–43
17. Jain S, Tomar DS, Sahu DR (2012) Detection of javascript vulnerability at Client Agen. *Int J Sci Technol Res* 1(7):36–41
18. Chen P.W, Jiang B.S, Wang C.H (2017) Blockchain-based payment collection supervision system using pervasive Bitcoin digital wallet. In: 2017 IEEE 13th international conference on wireless and mobile computing, networking and communications (WiMob)
19. Holotiuk F, Pisani F, Moermann J (2017) The impact of blockchain technology on business models in the payments industry
20. Chugh N, Sharma D. K, Singhal R, Jain S, Srikanth P, Kumar A, Aggarwal A (2020) Blockchain-based Decentralized Application (DApp) Design, Implementation, and Analysis With Health-care 4.0 Trends. In: *BASIC & CLINICAL PHARMACOLOGY & TOXICOLOGY* (Vol. 126, pp. 139–140). 111 RIVER ST, HOBOKEN 07030-5774, NJ USA: WILEY
21. Chowdhury MJM, Ferdous MS, Biswas K, Chowdhury N, Muthukumarasamy V (2020) A survey on blockchain-based platforms for IoT use-cases. *Knowl Eng Rev* 35:e19
22. Kiayias A, Russell A, David B, Oliynykov R (2017, August) Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Annual international cryptology conference. Springer, Cham, pp 357–388
23. Duong T, Fan L, Zhou HS (2016) 2-hop blockchain: combining proof-of-work and proof-of-stake securely. *Cryptology ePrint Archive*, Report 2016/716
24. Chen PW, Jiang BS, Wang CH (2017, October) Blockchain-based payment collection supervision system using pervasive Bitcoin digital wallet. In: 2017 IEEE 13th international conference on wireless and mobile computing, networking and communications (WiMob). IEEE, pp 139–146
25. English E (2018) Can Blockchain help reduce the financial Industry's cyber risk? MMC Publication

Chapter 17

Wireless Lighting System for Rural Households in India



Shantanu Acharya , Priya Debnath, Dipta Chakraborty, Rimpi Baishya, and Sayan Deb

1 Introduction

More than 80% of global electricity is generated by fossil fuel-based power generating station [1], generally located in the outskirt of the urban region. Despite remarkable development in modern society, 46% of the world's population lives in rural areas [2]. This percentage is higher (67% [2]) in developing countries like India. Most of the rural residents in this country sustain their life without power. According to Bloomberg [3], 240 million Indians do not have access to electricity. The primary reason for this scarcity is the hostile topography for which it is difficult to install electrical utilities in these regions. Sometimes it is not economically viable to extend the transmission line up to the required location. Poverty is another big issue for which the occupants are unable to draw electricity to their homes and pay bills. Thus, they do not get sufficient light at night. Consequently, many mishaps like an animal attack, snake bite, house catching fire from the luminary sources (candle, lantern, etc.) take place.

The lighting-related problem of rural households can be alleviated by employing a renewable energy-based standalone system. Solar PV (SPV) with battery storage is a good option. But, the price and weight of the SPV and storage system are the prime limiting factor for its use in rural areas. The conventional SPV system is suitable for well to do inhabitants living in buildings with concrete roofs. Generally, the families living in the rural areas belong to the financially weaker section. Due to their economic condition, they can hardly afford a comfortable life. The contemporary houses in rural areas have mud-walls with roofs made up of thatch, grass, leaves, polythene sheet sandwiched between split bamboos, 'Khapra'-curved country fired tiles, corrugated tin, etc. [2]. In India, 21.9% [4] houses have roofs of grass, thatch, bamboo, wood,

S. Acharya (✉) · P. Debnath · D. Chakraborty · R. Baishya · S. Deb

Department of Electrical and Electronics Engineering, Institute of Chartered Financial Analysts of India (ICFAI) University, Agartala, Tripura, India

mud, etc., and 32.6% have tiles. None of these roofing materials is strong enough to bear a load of large-sized SPV system for the augmented weight. Also, they are incapable of holding the SPV supporting structure even in low wind. Moreover, the power cables and charge controllers enhance the load on roof and expenditure.

It is possible to transfer electricity without wires [5]. Thus, an integrated wireless power transmission (WPT) scheme having a low wattage LED lamp and a small-sized SPV backed up with battery storage can prove to be an effective solution. Among the luminaries, LED is more preferred because, for producing the same amount of light, LED consumes the least power. For example, for 800 lm of light, an LED, an incandescent and a CFL lamp will consume 6–8 W, 40 W and 9–13 W, respectively [6].

This paper presents an experimental study on incorporation of SPV with WPT for eliminating the difficulties associated with the illumination in rural areas at an affordable price. Analysis and discussion of the obtained results have also been done in the manuscript. This inevitable issue and its low-cost solution have not been addressed hitherto in any of the existing literature.

2 Methodology

WPT is a process of transferring electricity without any physical link. Based on distance, it can be classified into two categories—near-field and far-field. In this research, near-field WPT has been adopted. In the scheme, power is transferred over a short distance using inductive coupling between two coils.

The proposed system is composed of a low power-rated PV module, a battery charging circuit, a storage device (low voltage rating Lithium-Ion battery), a voltage boosting inverter circuit and two coils (transmitter and receiver). The conceptual model installed on the thatch roof of a typical rural house is presented in Fig. 1.

A battery with low voltage rating but considerable capacity is selected to reduce the size and the weight of the system. The battery is charged by the PV system via a voltage regulator IC. In general, the size and weight of the PV module including the supporting structure vary with the power rating of the panel. As the battery rating is low, the weight and size of the SPV panel will be diminutive, and hence, there will be no need of heavy metallic supporting structure. Thus, the whole system can be implemented even on roofs made up of thatch, tiles, etc. The transmitting coil will be placed on the outer surface of the roof, whereas the receiving coil will be affixed on the inner surface, inside the room. These coils will interactively transfer the power without any physical contact. For protection from rain and other weathering effects, the transmitting coil along with the electronic circuitry and other components (except solar panel) will be housed in a box (transmitting box) mounted on the roof. The complete system can be divided into two parts—a solar battery charging system and a wireless power transferring system.

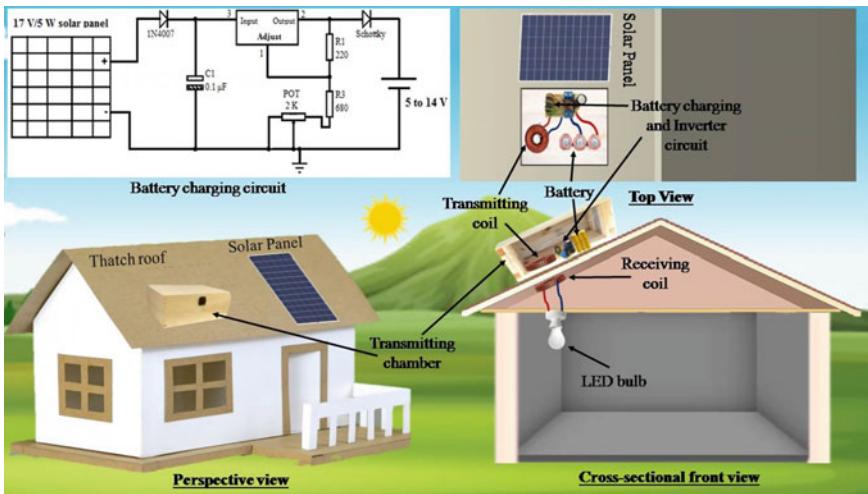


Fig. 1 Conceptual model

2.1 Solar Battery Charging System

The solar battery charger is a very simple circuit. A typical circuit for charging the batteries of variable voltage rating (5–14 V) is shown in the top left corner of Fig. 2.

In the circuit, the solar panel will trap the light energy coming from the sun and convert it to electricity. Since solar radiation is intermittent, a voltage regulator is required to provide steady power for charging the battery. In the above circuit variable voltage regulator-LM317T is used for voltage and current regulation. The diode D1 protects solar panel from the reverse polarity and capacitor C1 prevents static discharge. As the circuit is designed to charge batteries of different rating, reverse power that will come across the regulator IC (when the battery is not charging) may be of high value. So, there should be a protecting component which can handle high power. A Schottky diode has this ability, and hence, it is employed for the protection

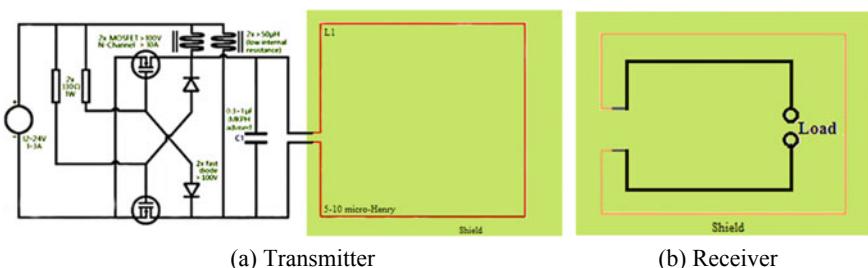


Fig. 2 Wireless transmitter and receiver circuit

of LM317T. The POT is used to vary the output voltage for charging batteries of different values.

2.2 Wireless Power Transferring System

The WPT involves the inductive transmission of energy from a transmitting coil to a receiving coil in the vicinity via an oscillating magnetic field. An inverter circuit is required to convert the steady current from the battery into time-varying current for producing alternating fluxes. An inverter generally consists of a switching device and an inductor. The switching device conducts and blocks the DC at some fixed interval yielding pulsating current, whereas the inductor helps in waveform shaping of the obtained output. There is a capacitor for resonating the circuit and an inductive shield for reducing the leakage of fluxes. For short distance WPT, higher switching rate is preferred as transmission efficiency improves with increment in frequency [7]. A typical circuit for wireless transfer of electrical power is shown in Fig. 2a. In the circuit, two N-channel MOSFETs and two ferrite core inductors of $50\ \mu\text{H}$ are connected for generating high-frequency oscillation. Capacitor C1 cancels the inductive reactance for transferring more power. The circuit changes the DC into AC at a high frequency of 100 kHz. Thus, fluctuating fluxes are generated around the transmitting coil and if these fluxes link with any nearby receiver coil, induce e.m.f. in it. There is another shield behind the receiver coil as shown in Fig. 2b, to prevent the outflow of the fluxes. When any load is connected across the receiver coil, current starts to flow and the load (here LED) starts to operate. A prominent example of short distance WPT is wireless mobile charging.

3 Experimentation

It was not possible to experiment with an actual sized device in the real condition as permission was not granted by any of the house owners in the nearby rural area. Therefore, a small-sized prototype has been developed in the laboratory (Fig. 3.) and experiment is carried out with different materials like thatch, tile, polythene, tin, (Fig. 4.) which are generally used for roofing purpose in these regions.

The main objective of the experiment is to study the behaviour of the system with various roofing materials at different distances. Thus for the ease of experimentation, a very simple setup consisting of a solar panel (of 6 V, 60 mA), a switch, a transistor (BC547), a resistor ($1\ \text{k}\Omega$) and two coils (one fixed and one movable; made up of 0.5 mm diameter copper wire) has been developed. The resonating component, battery and charging section has been discarded to keep the system simple. The steady power, analogous to battery output is supplied by the solar panel. To keep the output stable for better assessment of the system, the solar panel is powered by an incandescent lamp of 100 W rather than intermittent solar radiation. In the

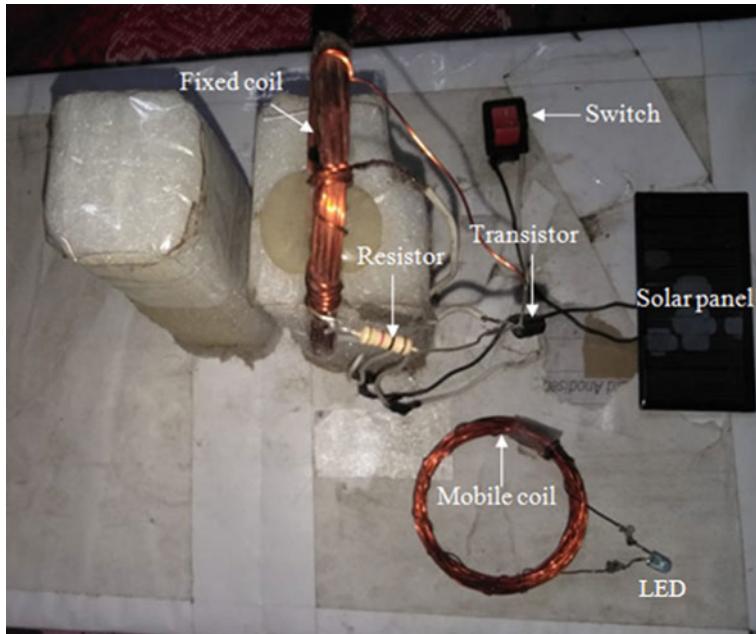


Fig. 3 Experimental set-up

experiment, the transmitting coil of the actual concept is represented by the fixed coil, the receiving coil by a mobile coil and the luminary by an LED.

Under the incandescent lamp at a distance of 10 cm, the open-circuit voltage and short-circuit current of the solar panel is measured 6.53 V and 16.73 mA, respectively. When the transmitting coil is connected in the circuit, the voltage drops to 530 mV and current to 11.77 mA. The transistor converts this steady current to pulsating current to generate a fluctuating flux and induce e.m.f. in the receiving coil. The flux linkages and hence the induced e.m.f. are proportional to the distance between the transmitting and the receiving coil. So, voltage and current are measured at fixed distances of 0.8, 1, 1.5, 2, 2.5, 3, 3.5 cm from the transmitter separately for each of the roofing material placed in between to replicate the real condition.

3.1 Results and Discussion

I-V characteristics. Figure 5 shows the relationship between current and voltage for each roofing material at different distances. In case of thatch (Fig. 5a), tiles (Fig. 5b) and polythene (Fig. 5c), the current drops by $1 \mu\text{A}$ for first 0.2 cm (0.8–1 cm) change in position. But, the induced voltages do not vary concurrently. Maximum drop of 10 mV for first 0.2 cm is observed with polythene, whereas, with thatch and tiles, it

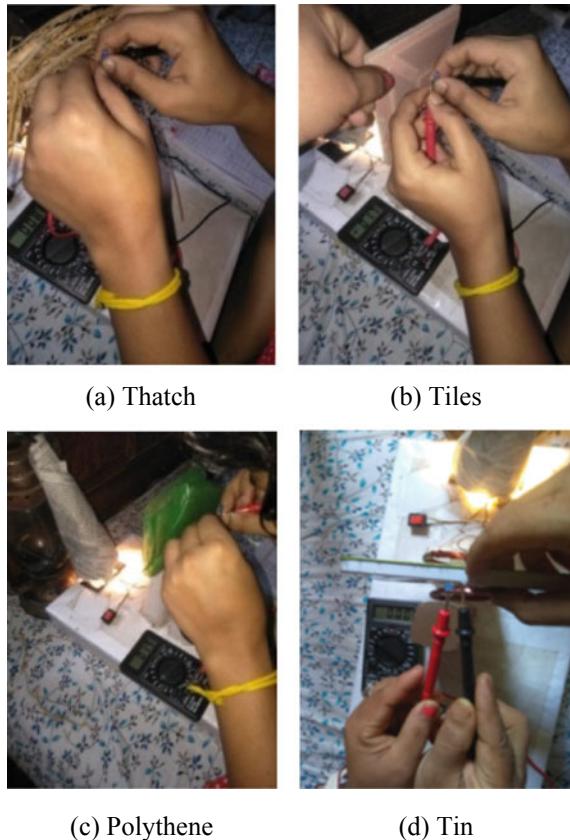


Fig. 4 Experiment with roofing materials

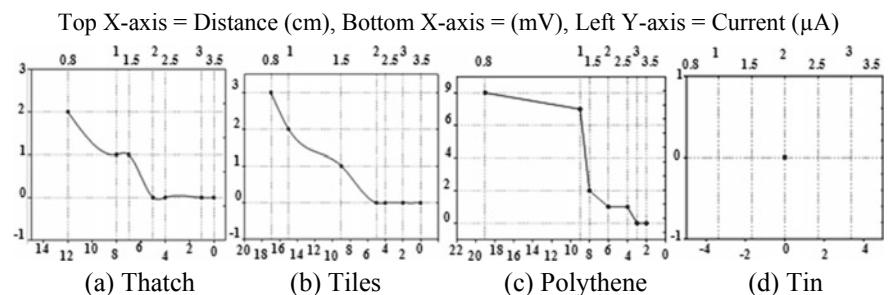


Fig. 5 I-V characteristics of the system for each roofing materials at different distances

is found to be 2 and 4 mV, respectively. Current is realized up to 2.5 cm for polythene and 1.5 cm for each tiles and thatch.

Contrary to these materials, tin behaves differently. It acts as an inductive shield and blocks the fluxes from reaching the receiver coil. Thus, no e.m.f. is induced in the receiver and hence no current flows through the load (LED). Consequently, current and voltage are zero as visible in Fig. 5(d).

Top X-axis = Distance (cm), Bottom X-axis = (mV), Left Y - axis = Current(μA)

Power at different distances. The variation in power with distance is shown in Fig. 6. It is apparent from the figure that the wireless power reception significantly relies on distance. In the figure, it is observed that highest power transfer at 0.8 cm takes places with the polythene ($0.152 \mu W$), followed by tiles ($0.051 \mu W$) and thatch ($0.024 \mu W$), respectively. With tiles and thatch, power is available up to 1.5 cm and becomes zero at 2 cm, whereas, with polythene, electricity can be perceived up to 2.5 cm and turns to zero at 3 cm. In case of tin, as all the emitted fluxes links with it, no power is available in the receiver as visible in the figure.

Power reduction rate. During the experiment, it is observed that maximum wireless power is available at the vicinity of the transmitter coil due to utmost flux linkages. Highest power measured in the receiver coil is $0.36 \mu W$ and rate of power diminution is calculated with respect to this maximum available power. Decremental rate of power transfer is presented in Fig. 7. It is detected in the figure that polythene has most versatile reduction rate. Diminishing power rate in case of tiles at 0.8, 1, 1.5 and 2 cm is 93.63, 91.66, 97.50 and 100%, whereas in case of thatch at similar distances are 93.33, 97.78, 98.06 and 100%, respectively. Since no power is transferred in case of tin, the attenuation rate is 100% in all position. The average declination rate for tiles, polythene, thatch and tin is 95.69, 88.84, 97.29 and 100%, respectively. For every 0.1 cm change in distance, power for tiles, polythene and thatch are found to vary averagely by 0.15, 0.07 and $0.16 \mu W$, respectively.

Fig. 6 Reception at different distances

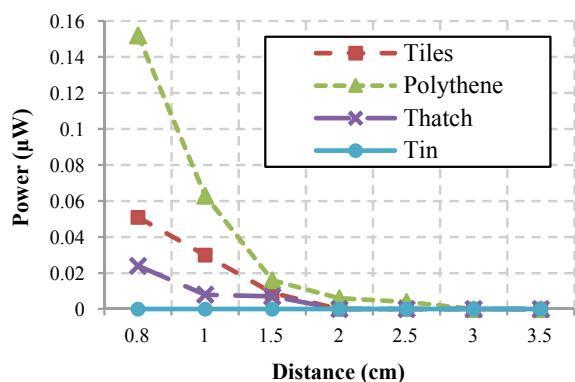


Fig. 7 Power decremental rate

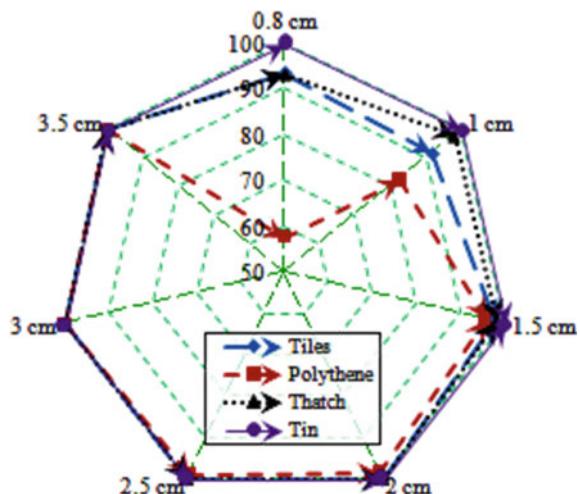


Fig. 8 Experiment with concrete



(a) With rods

(b) Without rods

Experiment with concrete.

Another experiment has been performed to verify the possibility of implementing the proposed system in building made up of concrete. For that purpose, two slices of concrete—with and without rods, have been placed in between the transmitting and receiving coil, as shown in Fig. 8a, b. It is observed that in case of reinforced concrete, the LED did not glow (Fig. 8a) as all the transmitted flux got linked with the reinforcing rods. On the contrary, flux penetrates through the concrete and links with the receiving coil in absence of rods. Thus, the LED connected with the receiving coil glows (Fig. 8b).

4 Conclusion

Following conclusion can be drawn from the experiment.

- The proposed system is a lightweight and affordable solution to lighting problems in rural areas.
- Roofs made up of polythene are the best option for the proposed scheme, but the system can also be installed in tiled and thatched roofs.
- The proposed system cannot be implemented in houses having tin roofs and reinforced concrete building.
- WPT is inversely proportional to the distance between transmitter and receiver.
- Power loss is more in the absence of resonating component.

References

1. Acharya S, Bhattacharjee S (2014) Stirling engine based solar-thermal power plant with a thermo-chemical storage system. Energy Conversion Manag 86:901–915
2. Kulshreshtha Y, Mota NJA, Jagadish KS, Bredenoord J, Vardon PJ, Loosdrecht MCMV, Jonkers HM (2020) The potential and current status of earthen material for low-cost housing in rural India. Constru Build Mater 247:118615
3. Living in the Dark: 240 Million Indians Have No Electricity. <https://www.bloomberg.com/news/features/2017-01-24/living-in-the-dark-240-million-indians-have-no-electricity>. Last accessed 6 Feb 2020
4. Housing. https://censusindia.gov.in/census_And_You/housing.aspx. Last accessed. 6 Apr 2020
5. Wakte G, Nadu HK (2016) Wireless transmission of electrical energy by using inductive coupling. Int Res J Eng Technol 3(7):1779–1785
6. LED Watt Conversion & Light Replacement Guide. <https://idavidmcallen.wordpress.com/2014/05/05/led-watt-conversion-light-replacement-guide>. Last accessed 6 June 2020
7. Wireless power transfer. https://en.m.wikipedia.org/wiki/Wireless_power_transfer. Last accessed 6 June 2020

Chapter 18

A Study of OpenStack Networking and Auto-Scaling Using Heat Orchestration Template



Karamjeet Kaur , Veenu Mangat , and Krishan Kumar

1 Introduction

Cloud computing is an Internet-based/on-demand computing paradigm that is rapidly flourishing and restructuring the academic world. Cloud computing as a technology is gaining lot of attention over the past few years. Cloud computing platform providing IT infrastructure and services has become a noteworthy element for IT companies [1]. The National Institute of Standard and Technology (NIST) articulated most commonly used definition of cloud computing as follows [2]: “Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, and applications) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” Cloud computing offers essential characteristics such as rapid elasticity, resource sharing, on-demand service, billing service, and broad network access service [3].

In cloud-based new IT world, infrastructure is moving from traditional dedicated mechanism to the new innovative, scalable, cost-effective, and dynamic cloud-based system. The server virtualization means servers are moving from physical to virtual mode by virtualizing the compute and storage on top of commercial-off-the-shelf (COTS) server [4]. Although server virtualization offers significant flexibility to the cloud infrastructure, but to gain full power of cloud computing, the networking landscape is also required to change. In traditional network architecture, servers, networking, and applications are tightly bundled with each other. But in present days, servers and networking are fully flexible to support complex applications. Software-defined networking (SDN) [5] and network function virtualization (NFV) [6] are two emerging and hyped technologies that provide networking and server virtualization, respectively. Both these technologies offer flexibility and agility that

K. Kaur · V. Mangat · K. Kumar
UIET, Panjab University, Chandigarh, India
e-mail: vmangat@pu.ac.in

are required by cloud computing. Overall, cloud computing is a significant platform with the main aim to provide higher scalability, lower operating cost, less up-front investment in infrastructure during deployment, easy access through a Web interface, reduced risks, and maintenance expenditure.

The main contributions of this paper is to illustrate practical hands-on neutron networking using OpenStack Ussuri release along with its scalability issue. The paper also demonstrates how the heat orchestration template (HOT) is used to solve these issues with minimal efforts. The structure of this paper is organized in the following manner. OpenStack Ussuri releases along with its different components, and currently, added features are discussed in Section II. Section III elaborates the experiment setup to implement neutron networking and heat orchestration and presents its evaluation. The last section concludes the paper with observations and future scope for research.

2 OpenStack Ussuri

OpenStack is an open-source cloud computing project that provides Infrastructure as a Service (IAAS) cloud on top of commodity hardware for both public and private clouds. Rackspace and NASA actively collaborated in July 2010, and then, they launched ubiquitous open-source cloud computing platform named OpenStack [7]. OpenStack is written in Python language and is freely available under Apache license. OpenStack basically consists of a large number of open-source software projects that are used to deliver various components of cloud infrastructure. OpenStack provides full administrative control by the provisioning of resources through Web user interface or a command line interface. Each of the OpenStack projects consists of REST API so that all resources can be managed through the dashboard (Web interface). The most widely used core projects are compute, storage, network, identity, and image that collaborate with other dozens of projects to deliver the requisite cloud to the end users [8].

Ussuri is a 21st release of OpenStack on 13 May, 2020. OpenStack uses six-month time-based release cycle for the development. They assigned alphabetically order code names to distinguish one release from the second one. The OpenStack foundation is supported by the IT vendors such as Rackspace, Red Hat, Intel, Ericsson, AT&T, Huawei, SUSE, IBM, Dell, and so on. The current release of OpenStack, Ussuri received more than 24,000 code changes by more than 1,000 developers from 188 organizations over 50 different countries. Moreover, OpenStack community included 107333 community members, 700 supporting organizations, and 187 countries registered till May 2020. So, it is one of the fastest growing open-source cloud computing communities in the world. OpenStack is an open-source, openly designed, openly developed solution by the open community. The following are the additional features added by the OpenStack community in Ussuri release (May 13, 2020) than the previous release Train (October 16, 2019) [9].

- Nova (Added support for cold migration and resizing the server)
- Kuryr (Bridge between OpenStack and container networking, support for IPV6)
- Ironic (Bare metal provisioning, added support for hardware)
- Octavia (Load balancing service in specific availability zone)
- Kolla (Containerized deployment, added support for TLS encryption)
- Neutron (Stateless security group added)
- cyborg (Accelerator such as FPGA, GPU lifecycle management)
- Magnum (Container infrastructure management service, operating system of Kubernetes cluster)
- Zun (Container service, API to create pod)

OpenStack architecture consists of a set of software tools that are responsible for some dedicated functionality and exposes some REST API [10]. Among all of them, some of the core projects are Keystone (Identity), Glance (Images), Nova (Compute), Neutron (Networking), Swift (Object Storage), Cinder (Block Storage), Horizon (Dashboard). OpenStack project is growing every day by adding a greater number of projects into it for different purposes by the OpenStack community. In addition to the core services, some other runner-up projects are Heat (Orchestration) [11], Magnum (Containers on top of OpenStack), Sahara (Data Processing), Trove (Database as a Service), Tacker (NFV Orchestration), Kuryr (Container Service), Octavia (Load Balancing as a service), Designate (DNS as a service), Barbican (Key Management), Searchlight (Indexing and Search), etc.

3 Experiment Setup and Evaluation

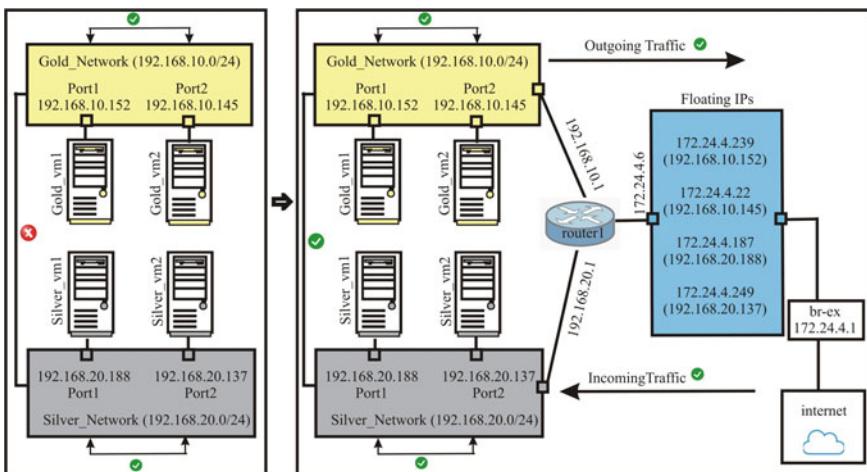
We have implemented OpenStack current release (Ussuri) based on OpenStack Packstack utility using all-in-one method. OpenStack is installed on a personal laptop in which type-2 hypervisor (VMWare Player) [12] is used to create virtual machine having Linux (centos8) operating system [13]. Using all-in-one mode, OpenStack services such as Keystone, Glance, Nova, Horizon, Swift, Cinder, Heat, Neutron are installed on a single virtual machine. Although installation of OpenStack using Packstack is less customisable, but it offers hands-on experience on OpenStack with limited hardware resources. Table 1 depicts the virtual machine configurations to implement OpenStack Ussuri release.

We have created virtual network topology in OpenStack that consists of two private networks, viz. Gold_Network (192.168.10.0/24), Silver_Network (192.168.20.0/24), and one default public network (172.24.4.0/24). Then, we have created Gold_vm1, and Gold_vm2 in Gold_Network and having IP addresses 192.168.10.152 and 192.168.10.145, respectively. Also, we created Silver_vm1, and Silver_vm2 in Silver_Network and having IP addresses 192.168.20.188 and 192.168.20.137 as shown in Fig. 1.

We have implemented three different scenarios for the evaluation of our experiment. In first scenario, after the creation of networks, Gold_vm1, and Gold_vm2

Table 1 Configuration of virtual machine to implement OpenStack

Resource	Configuration
RAM	12 GB
Hard disk	60 GB
Processor	Intel VT-x/EPT or AMD-v/RVI
Operating system	Centos8
Network adapter (2)	NAT, Bridged
Openstack software	Ussuri

**Fig. 1** Network topology having Neutron-OpenvSwitch and L3-agent

<pre> 1: ip a s 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue [a] link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 brd 00:00:00:00:00:00 scope host lo inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000 link/ether fa:10:3e:1c:a4:dd brd ff:ff:ff:ff:ff:ff inet 192.168.10.152 brd 192.168.10.255 scope global eth0 inet6 fe80::fa10:3e!%eth0/64 brd ff:ff:ff:ff:ff:ff scope link valid_lft forever preferred_lft forever ping 192.168.10.152 PING 192.168.10.152(192.168.10.152) 56 data bytes 4 packets transmitted, 4 packets received, 0% packet loss round-trip min/avg/max = 1.971/46.500/106.363 ms -- 192.168.10.152 ping statistics -- 4 packets transmitted, 3 packets received, 0% packet loss round-trip min/avg/max = 1.971/46.500/106.363 ms </pre>	<pre> 1: ip a s 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue [b] link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 brd 00:00:00:00:00:00 scope host lo inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000 link/ether fa:10:3e:1c:a4:dd brd ff:ff:ff:ff:ff:ff inet 192.168.10.145 brd 192.168.10.255 scope global eth0 inet6 fe80::fa10:3e!%eth0/64 brd ff:ff:ff:ff:ff:ff scope link valid_lft forever preferred_lft forever ping 192.168.20.137 PING 192.168.20.137(192.168.20.137) 56 data bytes 4 packets transmitted, 4 packets received, 0% packet loss </pre>
---	---

Fig. 2 **a** Ping results of intra-domain networking, **b** Ping results of inter-domain networking

attached to Gold_Network and Silver_vm1, and Silver_vm2 attached to Silver_Network in OpenStack network topology. As shown in Fig. 2a, Gold_vm2 is able to send ping request to Gold_vml, but cannot send the traffic to Silver_vm2. Therefore, by default L2-layer plug-in (neutron-OpenvSwitch-agent) is able to perform only intra-networking and inter-networking is not possible with this agent as shown in Fig. 2b.

```

(a) ip a s
: 1: lo <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 0.0.0.0 scope host lo
        valid_lft forever preferred_lft forever
: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:1e:6f:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.137/24 brd 192.168.20.255 scope global eth0
        netmask 255.255.255.0
        valid_lft forever preferred_lft forever
ping 192.168.10.145
PING 192.168.10.145 (192.168.10.145) 56 data bytes
8 bytes from 192.168.10.145: seq=0 ttl=63 time=10.632 ms
8 bytes from 192.168.10.145: seq=1 ttl=63 time=4.963 ms
--- 192.168.10.145 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 4.963/7.297/10.632 ms

(b) ip a s
: 1: lo <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 0.0.0.0 scope host lo
        valid_lft forever preferred_lft forever
: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:1e:6f:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.137/24 brd 192.168.20.255 scope global eth0
        netmask 255.255.255.0
        valid_lft forever preferred_lft forever
ping 192.168.20.137
PING 192.168.20.137 (192.168.20.137) 56 data bytes
8 bytes from 192.168.20.137: seq=0 ttl=63 time=9.568 ms
8 bytes from 192.168.20.137: seq=1 ttl=63 time=5.179 ms
--- 192.168.20.137 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 5.179/7.373/9.568 ms

(c) ping -c1 172.24.4.6
PING 172.24.4.6 (172.24.4.6) 56 data bytes
4 bytes from 172.24.4.6: seq=0 ttl=254 time=3.179 ms
--- 172.24.4.6 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.179/3.179/3.179 ms

(d) ping -c1 192.168.10.145
PING 192.168.10.145 (192.168.10.145) 56(84) bytes of data.
^C
--- 192.168.10.145 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
[root@server108 ~]# ping -c1 192.168.10.145
PING 192.168.10.145 (192.168.10.145) 56(84) bytes of data.
^C
--- 192.168.10.145 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
[root@server108 ~]#

```

Fig. 3 **a** Ping results from Silver_Network to Gold_Network, **b** Ping results from Gold_Network to Silver_Network, **c** Ping results from the internal to the external network, **d** Ping results from the external to the internal network

In the second scenario, we used Layer 3 plug-in (neutron-L3-agent) to solve the inter-networking problem that was encountered in the neutron-OpenvSwitch-agent. We created router named “router1” and set the 172.24.4.6/24 as an external gateway and added 192.168.10.1/24 and 192.168.20.1/24 as an internal interface. Now, we are able to communicate between two different networks as illustrated in Fig. 3a, b. Further, we created a network connection between internal (OpenStack VMs) and external (host operating system) network. As shown in Fig. 3c, d, we can send the ping request traffic from the internal to the external network, but it is not working in the reverse direction.

To resolve this issue, we have added the security rules (ICMP and TCP) into the default security group to define which type of traffic is allowed or not. Moreover, when any external host wants to access the internal cloud network, they need a floating IP address (public address) of cloud virtual machine. We have added the security rules and assigned the public IP address to each cloud virtual machine as shown in Fig. 1. Now, we are able to Ping, and SSH from the external network as shown in Fig. 4a, b. As in Fig. 5c, we can also access the Web services from the internal network. As a result, now both ingress and egress traffic are allowed. But when Gold_vm1 wants to access the Internet service, it is still not working. To solve it, add “net.ipv4.ip_forward=1” line into a router configuration file (/etc/sysctl.conf). Now, our virtual machine (Silver_vm2) successfully able to access the Internet.

To make the above solution more scalable and faster, OpenStack provides a heat orchestration module for the deployment. Heat orchestration is responsible for automatic deployment of infrastructure, services, and applications using flexible YAML templates as shown in Fig. 5. We can scale up and down the OpenStack cluster using this template. Heat orchestration is able to understand the AWS cloud formation (CFN) [14] template that is written in JSON language. But, it has its own heat

```

[a] root@server188 ~# ping -c1 172.24.4.249
PING 172.24.4.249 (172.24.4.249) 56(84) bytes of data.
64 bytes from 172.24.4.249: icmp_seq=1 ttl=63 time=5.88 ms
--- 172.24.4.249 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.882/5.882/5.882/0.000 ms
root@server188 ~#
[b] root@server188 ~# ssh cirros@172.24.4.249
The authenticity of host '172.24.4.249 (172.24.4.249)' can't be established.
RSA key fingerprint is SHA256:yM62zJH6tkTLR8iUbkaSpf6WuMcP89tqBb4ut4084.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.24.4.249' (RSA) to the list of known hosts.
cirros@172.24.4.249's password:
$ ls
$ cat >silver_vm2
Hello, I am a silver_vm2 user
$ 
[c]
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:f0:a5:2a brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.137/24 brd 192.168.20.255 scope global eth0
        inet6 fe80::fa16:3eff:fe:52a/64 scope link
            valid_lft forever preferred_lft forever
[d]
$ curl http://172.24.4.2
***** Welcome to Ussuri Openstack *****

```

Fig. 4 **a** Assigning floating IPs, **b** Ping request to the internal network, **c** ssh Access to the internal network, **d** Web service Access from the external network

HOT template to add different networking module			
heat_template_version: "2015-10-15" description: "create nova instance" resources: Server_1: type: "OS::Nova::Server" properties: security_groups: - "security group ID" networks: - subnet: "subnet ID of private network" name: Gold_vml flavor: "m1.tiny" image: "Image ID" availability_zone: nova key_name: gold_key	heat_template_version: "2015-10-15" description: "create floating ip" resources: FloatingIP_1: type: "OS::Neutron::FloatingIP" properties: floating_network: "public network ID"	heat_template_version: "2015-10-15" description: "assign floating ip to vms" resources: FloatingIPAssociation_1: type: "OS::Neutron::FloatingIPAssociation" properties: floatingip_id: "Floating IP" port_id: "Port of VM"	heat_template_version: "2015-10-15" description: "create volume" resources: Volume_1: type: "OS::Cinder::Volume" properties: name: volume1 size: 2 volume_type: "iscsi"
heat_template_version: "2015-10-15" description: "create router" resources: Router_1: type: "OS::Neutron::Router" properties: external_gateway_info: enable_snat: true network: "public network ID" name: router	heat_template_version: "2015-10-15" description: "attach network with router interface" resources: RouterInterface_1: type: "OS::Neutron::RouterInterface" properties: router: "router ID" subnet: "subnet ID of private network"	heat_template_version: "2015-04-30" description: "attach volume" resources: VolumeAttachment_1: type: "OS::Cinder::VolumeAttachment" properties: instance_uuid: "instance ID" volume_id: "volume ID"	

Fig. 5 Different networking modules of .yaml file

orchestration template (HOT) written in YAML format [15] and is easily accessible to admins, architects, and other non-coders. Basically, instead of creating different operations such as instances, volumes, security group, floating IPs, images individually, we can define a STACK that consists of a set of resources in a text file.

(a)									
ID	Stack Name	Project		Stack Status	Creation Time	Updated Time			
a4ef1d36-61fb-4a6b-bb72-5389127e8296	stack-1		d5926d1337164292965a211aad561dce	CREATE_COMPLETE	2828-07-01T13:30:34Z	None			
(b)									
Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	
vm1	cirros	192.168.10.144	m1.tiny	orchestration_key	Active	nova	None	Running	

Fig. 6 a Check created stack, b Created nova instance using stack

Here as an example, we showed that how the single VM instance is to be launched using HOT template. We created Stack “stack-1” using the dashboard and checked the creation progress in CLI mode as shown in Fig. 6a. After successful creation of a stack, VM instance is successfully created as shown in Fig. 6b.

4 Conclusion

This paper demonstrated the full-fledged neutron networking in OpenStack having neutron-OpenvSwitch-agent and neutron-L3-agent and highlighted scalability issues. To remove these auto-scaling issues of Neutron component, heat orchestration template (HOT) is used to define the infrastructure in a yaml format. Rather than deploying a single operation individually, using Heat, we can launch all operations in a one-step using stack. A study of a number of research papers in this field showed that the practical aspects of Neutron and Heat are missing in all these papers. At the end, we have concluded that this is only a small subset of what functionality can be provided with Heat. We can further use the heat template to deploy complex applications consisting of database and Web servers, etc.

As a future work, we want to explore OpenStack Magnum project to deploy containers for Neutron networking rather than VMs. Magnum project also uses Heat for auto-scaling of containers.

References

- Mastelic T, Oleksiak A, Claussen H, Brandic I, Pierson J, Vasilakos A (2014) Cloud computing: survey on energy efficiency. ACM Comput Surv 47(2):1–36
- NIST Cloud Computing Program, <https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp>. Last Accessed 24 June 2020
- Goyal S (2014) Public vs private vs hybrid vs community-cloud computing: a critical review. Int J Comput Netw Information Security 6(3):1–20
- Jain N, Choudhary S (2016) Overview of virtualization in cloud computing. In: Proceedings—2016 symposium on Colossal Data Analysis and Networking (CDAN). IEEE, pp 1–4
- Son J, Buyya R (2018) A taxonomy of software-defined networking (SDN)-enabled cloud computing. ACM Comput Surv 51(3):1–36
- Barakabitze AA, Ahmad A, Mijumbi R, Hines A (2020) 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. Comput Netw 167:1–40

7. Rosado T, Bernardino J (2014) An overview of OpenStack architecture. In: Proceedings—18th international database engineering and applications symposium. ACM, pp 366–367
8. Sefraoui O, Aissaoui M (2012) OpenStack: toward an open-source solution for cloud computing. *Int J Comput Appl* 55(3):38–42
9. OpenStack Ussuri. <https://releases.OpenStack.org/ussuri/>. Last accessed 28 June 2020
10. Kumar R, Gupta N, Charu S, Jain K, Jangir S (2014) Open source solution for cloud computing platform using OpenStack. *Int J Comput Sci Mobile Comput* 3(5):89–98
11. Chen CC, Chen SJ, Yin F, Wang WJ (2015) Efficient hybriding auto-scaling for OpenStack platforms. In: Proceedings—2015 IEEE international conference on Smart City/SocialCom/SustainCom (SmartCity). IEEE, pp 1079–108
12. VMWare Player. <https://www.vmware.com/in/products/workstation-player.html>. Last accessed 10 June 2020
13. CentOS. <https://www.centos.org/>. Last accessed 10 June 2020
14. AWS CloudFormation template formats. <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-formats.html>. Last accessed 15 June 2020
15. Heat Orchestration template. https://docs.OpenStack.org/heat/rocky/-template_guide/hot_guide.html. Last accessed 15 June 2020

Part II

**Intelligent Algorithms: Recent
Developments**

Chapter 19

Recent Development, Challenges and Futuristic Trends in Cloud Computing—A Survey



Sahul Goyal and Lalit K. Awasthi

1 Introduction

Cloud computing is a scheme to deliver flexible and cost-effective computing services for storage, software analytic, artificial intelligence over the Internet and databases, etc. [1]. Remote accessibility of data at economical cost making cloud storage best alternative to hard drives as it follows pay-as-you-go model. Additionally, the modern high-end computer networking, hardware virtualization, along with service-oriented architectures, contribute to the growth of cloud computing at rapid pace. Despite, swift developments in cloud computing technologies, still scientist hasn't outline standard definition of cloud computing [2]. Several attempts have been carried out in standardized definition of this emerging technology [3–5]. Cloud computing service models have been categorized majorly in three main sections: The first category is defined as software as-a-service (SaaS) where application is located at cloud space and access to users is provided through web interfaces [6]. The second category in cloud computing is described as Platform-as-a-Service (PaaS) that provide platform to the user for developing and executing various applications by means of programming interface [7]. The last category of cloud computing is Infrastructure-as-a-Service (IaaS) which provides access various virtual infrastructures such as storage and virtual servers [7]. Beside these categories, cloud is also categorized into various specialized services combined under anything as a service (XaaS). Cloud services can also be characterized as public, private, community and hybrid cloud service model [8, 9]. The integration of public, private and community cloud models is known as hybrid cloud model. Despite all of these classifications and previous

S. Goyal (✉) · L. K. Awasthi
Dr. B. R. Ambedkar, National Institute of Technology, Jalandhar, India
e-mail: sahulg.cs.19@nitj.ac.in

L. K. Awasthi
e-mail: director@nitj.ac.in

reports on cloud computing, there are still many gaps to be filled, such as under-developed theories and cloud models, minimal research on frameworks and partial perceptions for methodologies to establish successful near-future cloud systems, all of which must be discussed in order to outline current challenges and future work directions. These gaps motivate the authors to present an extensive review on existing approaches in cloud computing highlighting the current challenges to the community of cloud system. However, future directions are outlined for cloud system development to facilitate the professional of cloud computing field.

The manuscript is structured as follows: Sect. 2 describes the key development in cloud computing infrastructure in the last decade followed by challenges and approaches in Sect. 3. The prospective research directions are outline to assist the professionals of cloud computing community for smooth progress of this emerging and challenging technology.

2 Development in Cloud Computing

The cloud computing technology has been developed gradually in last decade through a number of phases which include utilities computing, grid computing, software services and application service provision. Conversely, this technology is knitted with the advancements in the Internet services as well as the business technologies.

The first commercial cloud computing services was launched by sales force organization in 1999 to deliver cloud resources by means of global networks [10]. Earlier, a renowned scientist “John McCarthy” took first step to set the map for cloud computing by sharing timesharing approach to effectively use the expensive mainframes within the organizations. Significant leap toward concept of cloud services appears with the introduction of virtual software (like VMware) to run different operating systems on isolated environment. Afterward, in 1999, the in vision of sales-force.com started delivering enterprise applications via simple Website. The first civic release of Xen in 2003, namely virtual machine monitor (VMM), permitted the implementation of numerous virtual guest operating systems concurrently on a solitary machine [11]. The Amazon introduced its first cloud services called elastic compute cloud (EC2) in 2006 [12] that uses the concept of pay as you go by providing access to computer for all along with the development of individual applications. During 2010, software giant Microsoft stepped in cloud computing marketing space with Azure [13] to simplify the expansion of Web application over the Web. These days, this platform is comprised of the entire three cloud service platforms such as SaaS, PaaS and IaaS [14]. At the mean time, open-source cloud computing platform was introduced by Rackspace and NASA in the mid of 2010. This is a network of unified components used to handle the diverse multi-vendor hardware along with storage space and networking assets within a data center. After that, an enterprise class computing service was pioneered by IBM with the name of SmartCloud which was integration of diverse collaborators and SaaS applications. Furthermore, in 2013, IaaS cloud service model emerged as fastest growing cloud service delivery model and

attracted the investor in Worldwide Public Cloud Services Market [14]. Moreover, the spending for global marketing infrastructure and cloud-related services escalated by 20% within a single year having approximated cost of £103.8 bn. Furthermore, the Google Compute Engine (GCE) as an extension to Google cloud platform was previewed in 2012 and officially launched in 2013 [15]. GCE played a significant role in cloud computing as it offers the virtual machine on demand feature. In early 2015, the development of virtual technologies, availability of fast Internet speed and popular apps in various domains grew the presence of cloud computing. Furthermore, Dell introduced hybrid cloud computing, and cloud computing service is back bone of machine learning [16]. The arena of cloud computing services is extended in 2016, with the evolution of the Internet of Things (IoTs) [17, 18]. Recently, in 2017, IBM cognitive cloud computing platform got popularized followed by decentralized cloud reference architecture commonly known as fog computing [19, 20]. Lately, Microsoft has launched sea data center under the project Natick [21]. All these cloud computing approaches are divided into three phases as: The developments during 2005–2011 are placed in first-generation cloud, while the techniques developed during 2012–2017 are second-generation clouds. The recent improvement since 2018 to till date falls under the category of next-generation cloud computing. The cloud computing generation is depicted in Fig. 1. The various efforts in the development of cloud computing make this technology crowded, but still a lot of improvement is needed as multimedia and social networking content are increasing gradually, hence the challenges. In the next section, recent challenges are outlined along with prospective approaches to battle these challenges.

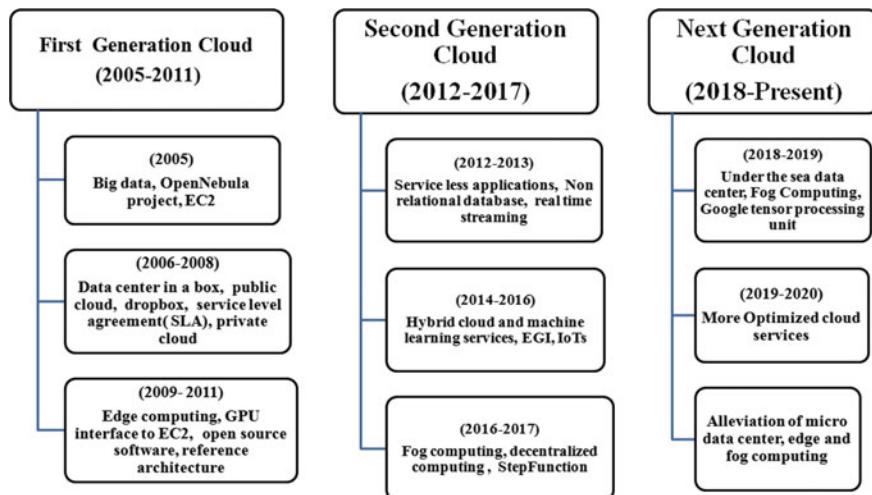


Fig. 1 Generation and development of cloud computing

3 Challenges in Cloud Computing

Despites various benefits and application areas of cloud computing, there are many management, technological, security and legal issues associated with this upcoming technology which need to be address to make it more effective in the near future. The critical challenge in cloud computing is data security and privacy [22, 23]. Therefore, in some cases, the “service-level agreement (SLA)” does not provide assurance to audit data; hence, the loss of data governance may lead to rigorous impact on the client data usage strategies [24]. Therefore, a huge research is being carried out to combat the security and privacy issues. Organizations are developing new data centers to accommodate exponentially growing new cloud. These data centers have added operation expenditures along with high energy consumptions and CO₂ emission rate. As a result, reducing data center energy usage without sacrificing on-demand cloud resource delivery is a critical challenge in future cloud technology [25]. Another key challenge in cloud computing is data lock-in as some clients may have to transfer data or service from one service provider to other [26]. Unfortunately, most of the cloud service suppliers have limited service interoperability and make it hard to migrate from one platform to another, which will be a highly needed feature in the near future.

Despites these technological challenges, cloud computing society is also facing legal issues such as such as intellectual property rights, contract law, data privacy and jurisdiction [26]. Therefore, an appropriate IT supremacy is needed to guarantee implementation; IT assets should be controlled properly according to well-defined procedures and policies. Conversely, the data quality management under government polices is highly required. Further, the cloud computing has not been concentrated into single cloud; however, multi-cloud state has been gradually grown in the last few years [27]. In this direction, multiple industries and organizations are integrating the public and private clouds as being lead by Alibaba and Amazon. Earlier, it was reported that approximately 81% of enterprises are using multi-cloud strategy. Therefore, concentration of cloud is becoming a challenge to improve the effectiveness of cloud computing. The forefront technologies, for instance “artificial intelligence (AI),” “machine learning (ML)” and big data, further argument the maturity of cloud computing technology, hence becoming a challenge to cope the rapidly growing data-based technologies [28]. As these challenges are paving stones in the efficiency, growth and execution of cloud computing systems, therefore it is needed to address precisely. Therefore, all major challenges are depicted in Fig. 2. Although if challenges exist, it gives birth to new opportunities for individuals; therefore, the future directions along with opportunities are addressed in the next section.

4 Futuristic Trends in Cloud Computing

Advancements on AI, ML, big data and Internet of Things (IoTs), with anticipated billions of sensor and devices, have increased the data load on servers. However,

Fig. 2 Challenges in cloud computing



these technologies have generated network traffic and latency in communication. Hence, to alleviate the network traffic and communication latency, fog computing and edge computing are emerging as an excellent contender [29, 30]. Further, in fog computing system, the applications are vertically scaled into diverse computing tiers which allow solitary necessary data traffic outside the data source. The open fog consortium has initiated the first step in direction of fog computing. Further, data security and privacy are key challenge at present, and block chain techniques in cloud computing will get fame in the near future. Edge computing will play major part with 5G to latency-free output for virtual reality and many more data-intensive applications. Edge computing technique creates mini-server or edge clouds at edge of network that enables to process massive data for superior real-time experience [31]. At the same time, integration of machine learning approaches attracting the professionals of cloud computing domain due its ability to predict user preferences as well as to diverse workloads. Therefore, more advanced machine learning techniques will be seen as a future prospective to improve the efficiency of cloud computing technology. Moreover, fine grain billing model is gaining attentions compared to most popular pay-as-you-go cost model, especially in serverless computing scheme. In serverless computing, server is not rented as traditional cloud server does and developers assume that applications are on cloud virtual machines [26]. However, serverless computing is also famous like event-based programming and “Function-as-a-Service” [32, 33]. Another opportunity is to develop programming model with ability to offer high-level abstraction to smooth the progress of the serverless computing. Additionally, the tradeoff between the utilization of external services and serverless cloud computing needs to investigate in for orchestration of next-generation cloud system. Further, software-defined networking (SDN) will be next research area to efficiently manage the increased data volume while supporting dynamic architectures [34]. This approach will also be beneficial to isolate the hardware

within the network that controls data traffic [29]. Despite that, research initiations are essential to assist the physically distributed protocols to logically control tasks while maintaining the “Quality of Service (QoS)” by incorporating the network and cloud infrastructures. The integration of resilience and software-defined computing has also foreseen as an emerging cloud computing concept. An additional avenue related to cloud computing is cognitive computing. The cognitive computing model is focused on machine learning algorithms that enable the cloud to become smarter by acquiring knowledge and finding solutions on its own [35]. Another emerging type of distributed computing is volunteer cloud computing, where unused resources are voluntarily denoted by private cloud providers to community people for big data-specific problems. This technique is at its early stage and has a lot scope in next-generation computing technologies. The most important challenge in volunteer cloud computing is to minimize the overheads in highly virtualized environment. Therefore, in the future, researcher will pay more attentions to mitigate the overheads issues. As cloud computing is also arising environment problems, therefore attentions need to be paid to reduce the data center power consumptions and emission of harmful gasses such as carbon dioxide. A report by Gartner estimated that ICT industry is accountable for 2% of global emission of CO₂ [36]. Therefore, problems will worse in coming days as cloud services are increasing day by day, hence attention needs to be paid to combat the energy combustion and harmful gasses emission. However, a new cloud computing environment is emerging as green computing architecture. However, more research efforts are needed to mitigate energy consumption and environmental issues. Conversely, cloud is getting crowded and technology is expanding gradually; still, golden period of cloud computing technology is yet to sunup on us.

5 Conclusion

In this manuscript, the significances and tasks preformed by cloud computing are presented. It is an emerging technology and currently in development phase and facing diverse challenges. Since public cloud storage has multi-tenancy clients, private data can be open to others, this technology has a significant security and privacy flaw. However, various developments in cloud computing are outlined in this manuscripts. The challenging trends have resulted in the requirements of new computing architecture to provide future ready cloud computing system. These upcoming cloud computing technique are expected to impact the various areas, for instance dedicated connection, data-intensive computing and self-learning. Further, a roadmap is suggested to mitigate the current challenges and outline the future research route to realize the prospective of the next-generation cloud computing systems.

References

1. Senyo PK, Addae E, Boateng R (2018) Cloud computing research: a review of research themes, frameworks, methods and future research directions. *Int J Inf Manag* 38(1):128–139
2. Senyo PK, Effah J, Addae E (2016) Preliminary insight into cloud computing adoption in a developing country. *J Enterp Inf Manag* 29(4):400–422. <https://doi.org/10.1108/JEIM-02-2014-0017>
3. Bayramusta M, Nasir VA (2016) A fad or future of IT?: a comprehensive literature review on the cloud computing research. *Int J Inf Manag* 36(4):635–644
4. Buyya R, Buyya R, Yeo CS, Yeo CS, Venugopal S, Venugopal S, Brandic I (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gene Comput Syst* 25:599–616. <https://doi.org/10.1016/j.future.2008.12.001>
5. Mell P, Grance T (2011) The NIST definition of cloud computing
6. Zissis D, Lekkas D (2012) Addressing cloud computing security issues. *Future Gene Comp Syst* 28(3):583–592
7. Mateescu G, Gentzsch W, Ribbens CJ (2011) Hybrid computing—where HPC meets grid and cloud computing. *Future Gene Comput Syst* 27(5):440–453
8. Hsu PF, Ray S, Li-Hsieh YY (2014) Examining cloud computing adoption intention, pricing mechanism, and deployment model. *Int J Inf Manage* 34(4):474–488
9. Mouratidis H, Islam S, Kalloniatis C, Gritzalis S (2013) A framework to support selection of cloud providers based on security and privacy requirements. *J Syst Softw* 86(9):2276–2293
10. Garg SK, Versteeg S, Buyya R (2013) A framework for ranking of cloud computing services. *Future Gene Comput Syst* 29(4):1012–1023
11. Awadallah A, Rosenblum M (2002) The vMatrix: a network of virtual machine monitors for dynamic content distribution. In: Proceedings of the 7th international workshop on web content caching and distribution (WCW 2002)
12. Amazon elastic compute cloud, May 2011. <http://aws.amazon.com/security>
13. Vaughan-Nichols, Steven J. (2019) Microsoft developer reveals Linux is now more used on azure than windows server. ZDNet. Retrieved July 2 2019
14. Parasuraman K, Srinivasababu P, Rajula Angelin S, Arumuga Maria Devi T (2014) Secured document management through a third-party auditor scheme in cloud computing. In: 2014 International conference on electronics communication and computational engineering (ICECCE) pp 109–118
15. Google (2014) Containers on google cloud platform. Google compute engine documentation. Retrieved 10 June 2014
16. Hybrid cloud architecture: selecting the best of both worlds available online: <https://cloudian.com/guides/multi-cloud-management/hybrid-cloud-architecture>. Last accessed on 26 June 2020
17. Mattern F, Floerkemeier C (2010) From the internet of computers to the internet of things. In: Sachs K, Petrov I, Guerrero P (eds), *From active data management to event-based systems and more*, pp 242–259
18. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): a vision, architectural elements, and future directions. *Future Gener Comput Syst* 29(7):1645–1660
19. Zhou B, Dastjerdi AV, Calheiros R, Srivatsa S, Buyya R (2016) mCloud: a context-aware offloading framework for heterogeneous mobile cloud. *IEEE Trans Serv Comput* (99):1–1
20. Deng S, Huang L, Taheri J, Zomaya AY (2015) Computation offloading for service workflow in mobile cloud computing. *IEEE Trans Parallel Distrib Syst* 26(12):3317–3329
21. MicroSoft Project Natick phase 2, available online: <https://natick.research.microsoft.com/>. Last accessed 26 June 2020
22. Stojmenovic I, Wen S, Huang X, Luan H (2016) An overview of fog computing and its security issues. *Concurr Comput Pract Exp* 28(10):2991–3005

23. Wang Y, Uehara T, Sasaki R (2015) Fog computing: issues and challenges in security and forensics. In: IEEE 39th annual computer software and applications conference, vol 3, pp 53–59
24. Buyya R, Garg SK, Calheiros RN (2011) SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. In: Proceedings of the international conference on cloud and service computing, pp 1–10
25. Yuan X, Min G, Yang LT, Ding Y, Fang Q (2017) A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds. Future Gener Comput Syst 76:63–72
26. https://d0.awsstatic.com/whitepapers/compliance/Cloud_Computing_and_Data_Protection.pdf. last accessed 27, June 2020
27. Barker A, Varghese B, Thai L (2015) Cloud services brokerage: a survey and research roadmap. In: Proceedings of the 8th IEEE international conference on cloud computing, pp 1029–1032
28. Assuno MD, Calheiros RN, Bianchi S, Netto MA, Buyya R (2015) Big data computing and clouds: trends and future directions. J Parallel Distrib Comput 79(80):3–15
29. Orsini G, Bade D, Lamersdorf W (2015) Computing at the mobile edge: Designing elastic android applications for computation offloading. In: 8th IFIP wireless and mobile networking conference, pp 112–119
30. Wang S, Urgaonkar R, Zafer M, He T, Chan K, Leung KK (2015) Dynamic service migration in mobile edge-clouds. In: IFIP networking conference, pp 1–9
31. Hassan N, Yau KA, Wu C (2019) Edge computing in 5G: a review. IEEE Access 7:127276–127289. <https://doi.org/10.1109/ACCESS.2019.2938534>
32. Hendrickson S, Sturdevant S, Harter T, Venkataramani V, ArpacıDusseau AC, Arpacı-Dusseau RH (2016) Serverless computation with openlambda. In: 8th USENIX workshop on hot topics in cloud computing
33. McGrath G, Short J, Ennis S, Judson B, Brenner P (2016) Cloud event programming paradigms: applications and analysis. In: IEEE 9th international conference on cloud computing, pp 400–406
34. Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T (2014) A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun Surv Tutor 16(3):1617–1634
35. Brown AD, Furber SB, Reeve JS, Garside JD, Dugan KJ, Plana LA, Temple S (2015) SpiNNaker—programming model. IEEE Trans Comput 64(6):1769–1782
36. Mohta A, Sahu RK, Awasthi L (2020) Robust data security for cloud while using third party auditor

Chapter 20

A Comparative Approach for Email Spam Detection Using Deep Learning



Akhil Pratap Singh, Ashish Singh, and Kakali Chatterjee

1 Introduction

In recent days, spam mails cause almost 45% of total email traffic on the internet. About 14.5 billion spam emails are sent every day, and 0.36% of all spam is from some advertising company. As per the report [1], 3–6 million spam emails were detected in 2015. In 2017, it increased by 56%. These spam emails contain malicious contents such as suspicious codes, macros, malware's, spyware, etc.; sometimes, these harms the procedures of the operating system, pass confidential data like ID, passwords or corrupt files, etc. Also, a large volume of spam emails flowing through the networks perform some destructive effects on the memory space of email servers, communication bandwidth, CPU power, and user time [2]. Not only this, it develops many threats in system security. It can lead to denial of service, buffer overflow attacks, and phishing attacks [3, 4], which hamper the data security. Hence, spam filtering is essential for the protection of the system as well as network security.

There are two popular approaches very often found in email spam filtering which are knowledge-based engineering and ML-based techniques [1]. In knowledge engineering, rule-based filtering tools are used for the detection of mail spams. But, it requires continual updates because the forms of mail spams have been changed day by day. For this reason, failure rates are high for the less aware persons. This

A. P. Singh (✉) · K. Chatterjee

Computer Science and Engineering, National Institute of Technology Patna,
Patna, Bihar 800005, India

e-mail: akhilp.phd19.cs@nitp.ac.in

K. Chatterjee

e-mail: kakali@nitp.ac.in

A. Singh

School of Computer Engineering, KIIT Deemed to be University,
Bhubaneswar, Odisha 751024, India

e-mail: ashish.singh@kiit.ac.in

failure can be covered up by ML-based approaches. In these approaches, no rules are required. In the training module, pre-classified email messages are supplied for training purposes so that they can learn the classification rules. Several works have been found in ML where naive Bayes (NB) [5], support vector machine (SVM) [6], neural network (NN) [7], K-nearest neighbour (KNN) [8], Rough Sets (RS) [9], and random forests (RF) [10].

In ML-based techniques, many significant improvements have been made in spam detection. But these techniques have some drawbacks such as limitations of fault tolerance, low learning capability, hidden messages inside stego images [11]. Also, the majority of spam filters have no ability of incremental learning in real-time [12]. Thus, there is a need to propose spam detection technique using deep learning. This technique has a large number of processing layers and levels of abstraction. In this paper, we have introduced a predictive model for the detection of spam emails based on deep learning techniques [13–15]. Our contribution is summarized below:

- We have reviewed and implemented existing ML-based techniques for the detection of spam emails.
- We have proposed a predictive model for spam email detection based on the deep learning technique.
- The performance of the ML-based and deep learning technique is discussed. From the experimental results, we have found a high detection rate of the proposed deep learning-based model as compared to other previous ML-based techniques.

The rest of the paper is organized as follows: Section 2 presents a review of filtering techniques as well as discusses existing works regarding spam detection. In Sect. 3, we have presented our proposed CNN-based deep learning spam detection model. Section 4 discusses performance and result analysis. Finally, the paper is concluded in Sect. 5.

2 Literature Survey

In this section, we have categorized different existing spam mail filtering techniques found in the literature [1]. They are described below:

- Content-Based Filtering: It works based on some rules and signatures. Rules or signatures are already established in algorithms. Based on the already established signature or pattern match, the classification algorithm defines the email is spam or not. Content-based approach is used to analyse email content, phrases used in email, word occurrences, word distribution, and number of words; then, it used some rules or signature to categorize each incoming email.
- Sample-based filtering: It is another way to filter email. Under this category, take a lot of real-world email-related issues and apply ML algorithms to analysed email issues. According to the analysed email issues, the email legitimacy will be checked.

- Rule-based filtering: It is one of the most important ways to classify emails. In this approach, we have some prior heuristic knowledge of the email. Every email tries to replicate or forward again, and again as parallel, the score is also increased. When the value exceeds the threshold or maximum mark, the email automatically becomes spam. This threshold limit remains unchanged, so the rules need to be continually updated for better classification.
- Adaptive Filtering: Adaptive filtering is one significant form of filtration. In this method, cluster emails are prepared based on certain basic features using ML algorithms. Each incoming mail is compared with the other clusters, and based on the percentage of similarities, the decision will be made. K-means clustering approach is used for clustering of emails.

Lueg [16] discussed various concepts related to the logical and theoretical approaches to implement email spam filtering. But, the practical implementation with ML algorithms is missing in the paper.

Wang [17] proposed different filtering methods and categorized emails into different categories based on some characteristics of email.

In [18], the author addresses numerous filtering strategies for better accuracy using the ML ensemble process. But, the paper did not address the latest email filtering issues.

Bhowmick et al. [19] explored in depth the various approaches of content filtering methods using ML algorithms. It addressed the basic definition of algorithm efficacy and explained the spam's essence such that it is simple to use ML approaches to analyse the spam.

Laorden et al. [20] addressed the identification of anomaly spam. The spam identification gives better precision and reliability. This also solves a multi-class problem present in the email.

Sahel et al. [21] applied methods that are based on a detection approach focused on anomalies behavior of email. Author used negative selection algorithm (NSA) technique to distinguish spam and non-spam emails. NSA is the sub-branch of an artificial immune systems (AIS). AIS methods are behavioral approach motivated with the biological environment by antibodies. NSA detects the antigens and takes toxic antigens down.

This paper [22] focuses on the early detection of spamming accounts (ErDOS) algorithm. The algorithm falls under the content- and feature-based classifications. The author describes the feature with the ratio of incoming emails and outgoing emails, receiving emails from multiple sources. The author uses another ErDOS-LVS principle which is used to classify spammer with a low volume. The author works on real-world data set.

Jain et at. [23] proposed a framework for the application of SMS and twitter spam detection using a deep learning algorithm based on convolution neural network (CNN) and long short-term memory (LSTM). For the twitter spam detection model, the accuracy of the framework is reached 92.80%. The framework does not cover the error rate, which plays a major role in evaluating the performance of the model.

Different email classification works have been done based on ML techniques [24], such as NN [25], SVM [26], and RF [25] in various paper. But, the performance is not up to the mark. So, we have decided to classify email by using deep learning techniques. Here, we have proposed a comparative approach based on CNN model for improving the accuracy of email spam detection model.

3 Proposed Model

This section represents a simplified proposed CNN-based spam detection model (Fig. 1). The working flow of the proposed CNN model is depicted in Fig. 2. In the existing system, most models use the ML approach to detect spam emails. But, due to the changing environment and dynamic nature of the network, the ML approaches are not much more efficient. Hence, the detection system needed a more strong spam detection mechanism. This proposed detection model's primary objective is to enhance the spam detection rate of the incoming emails. The other purpose is to reduce the error rate when the number of incoming emails is spam. Regarding these two issues, a CNN-based spam detection algorithm is proposed, which is applicable most of the cases. The proposed CNN-based model that is shown in Fig. 1 mainly consists of four phases: data input and preprocessing, feature selection and normalization, CNN-based deep learning model and result classification. Data input and preprocessing phase consist of an input layer, embedding, pre-trained weight and embedded vector. This phase is responsible for removing all unwanted data from raw data to achieve accurate and useful information. The feature extraction phase consists of embedding, pre-trained weight and embedded vector. This phase is mainly responsible for the selection of the most essential and relevant features which enhance the

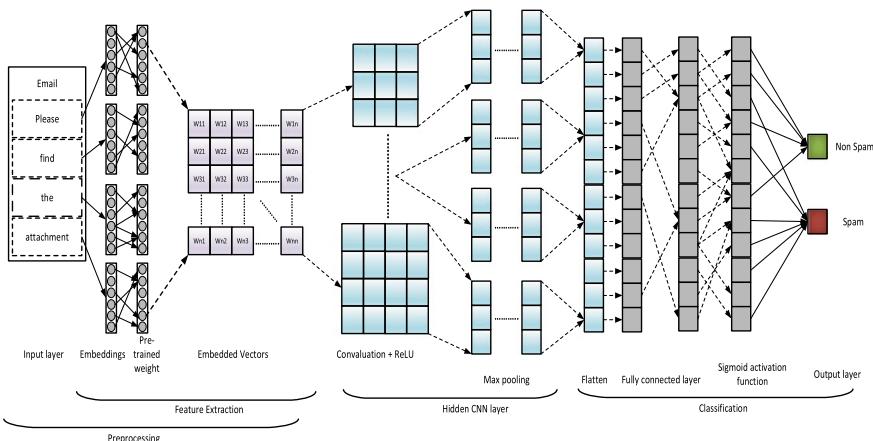


Fig. 1 Framework of the CNN model for spam detection

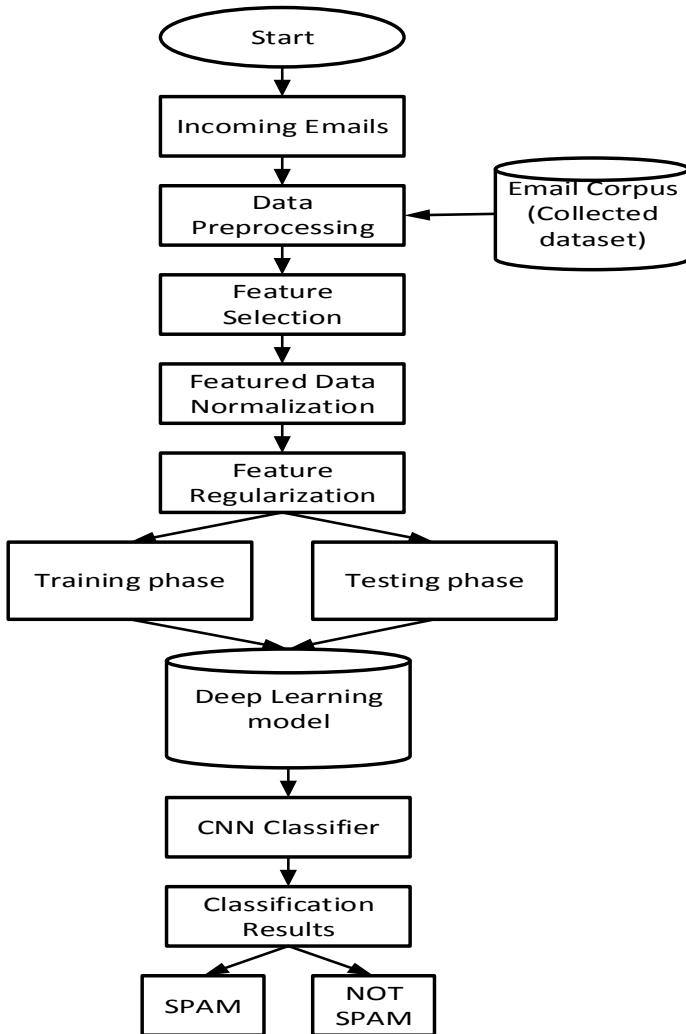


Fig. 2 Working of proposed model

result. The hidden CNN layer consists of convolution and pooling accountable for giving appropriate information from feature data. The classification phase consists of a flatten, fully connected layer, softmax activation function and output layer. This phase is responsible for the classification of results in terms of spam and non-spam emails.

A detailed discussion of the phases mentioned above is as follows.

3.1 Phase 1: Data Input and Preprocessing

Every incoming email is considered input data, for example, “Please find the attachment”, shown in Fig. 1. Data input for the training purpose is email corpus, and incoming mail is considered as data input for the testing purpose.

Preprocessing is the initial phase of the operating every incoming email. This phase includes tokenization in which the mail is transformed into concrete terms and then allocated with unique symbols to this word [27]. Remove punctuation is another work in which many symbols are removed that do not contribute more in email, e.g. comma, semi-comma, etc. Lemmatizing provides a single term of multiple words. For instance, go, going, go, went, gone contain the same meaning.

3.2 Phase 2: Feature Selection and Normalization

The goal of feature extraction is to transform the input data into a set of features. In deep learning model, it would be done automatically to extract the relevant features for better result [28].

Normalization is a process where the multiple values are re-scaled from zero to one for better understanding in a deep learning model. The most common approaches for normalization is z-score and t-normalization [29].

Regularization is the approach to minimize the model’s error rate by assigning appropriate weights and functions using backpropagation. In the proposed work, we initialize the importance after adding the gradient of descent [30].

3.3 Phase 3: CNN-Based Deep Learning Model

There are several steps to perform algorithms for machine learning. One of those is an exaction feature. Machine-leaning feature exaction is manual for email spam detection features such as noun, verb, adjectives, word length and wrong words. When we retrieved the correct function from the data set, it would result better, and some of them would have a lot of information yet hidden features. The extraction of features works fine in CNN-based deep learning for better results. There are several processes involved in CNN spam detection over email: embedding, convolution, ReLU and pooling.

Embedding is the process in which a low dimension is transformed into the space of a large dimension. Some embedding mechanisms are sparse, Word2Vec, WordNet, ConcepNet and dense embedding [31]. In this paper, we used dense embedding. The pre-train vector of words is known as Glove (V) [32]. The email content are embedded as $B(e_1), B(e_2), B(e_3), \dots, B(e_n)$.

Embedded vector (C) is created after the concatenation of these email content given in Eq. 1.

$$C = B(e_1) \cdot B(e_2) \cdot B(e_3) \cdots B(e_n) \quad (1)$$

Email is represented in the form of a matrix (C). It is the set of embedded vectors, and the size of $m*n$ is given by Eq. 2.

$$C = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix} \quad (2)$$

Once embedded matrix is created, the next step is the convolution layer. In this layer, the Kernel size (k_s) slid (stride) through pattern. Stride (s) is a step that is slid over the matrix. The default value of stride is one. It can be 2 or 3 or even 4, which depends on the dimension of the matrix. It is performed multiple times for extracting the features. It contains the multiplication and addition operation while convolving over the matrix given in Eq. 3. CNN layer consists of convolution, followed by the pooling.

$$C = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix} * \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ \vdots & \vdots \\ f_{n1} & f_{n2} \end{bmatrix} \quad (3)$$

where $*$ is the convolutional operator and $K_{i=1,2,\dots,n}$ is the feature vector which is given by Eq. 4.

$$\begin{aligned} K_1 &= e_{11}f_{11} + e_{12}f_{12} + \dots + e_{1n}f_{1n} + e_{21}f_{21} + e_{22}f_{22} + \dots + e_{2n}f_{2n} \\ K_2 &= e_{21}f_{11} + e_{22}f_{12} + \dots + e_{2n}f_{1n} + e_{31}f_{21} + e_{32}f_{22} + \dots + e_{3n}f_{2n} \\ &\vdots \\ K_n &= e_{(m-1)1}f_{11} + e_{(m-1)2}f_{12} + \dots + e_{m1}f_{21} + e_{m2}f_{22} + \\ &\quad e_{22}f_{23} + \dots + e_{mn}f_{2n} \end{aligned} \quad (4)$$

These feature maps (M) are represented in the form of vector. The size of feature map vector is $(n-k_s + 1)*1$ given in Eq. 5, where n is the total number of words, and k_s is the kernel size. Mathematically, the feature map function works as follows: assume that $n = 256$ and $k_s = 5$, then the feature map size is $256 - 5 + 1 = 252$.

$$M = \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ \vdots \\ K_n \end{bmatrix} \quad (5)$$

We are analysing CNN's deep learning model in detail here. CNN algorithm is described in Algorithm refqqqq. There are several functions involved in CNN spam detection over email.

Algorithm 1 CNN algorithm for Email Classification

- 1: INPUT: Acc_x, Acc_y, Acc_z = Acceleration data point in x,y,z co-ordinate and Gry_x, Gry_y, Gry_z = Gyroscope data point in x,y,z co-ordinate.
 - 2: OUTPUT: Class for testing email data sample.
 - 3: **procedure** SCAN WINDOW OF THE TRAINING DATA SAMPLE
 - 4: f_e = Feature extraction
 - 5: n_z = Featured data normalization
 - 6: r_z = Regularization of feature data set
 - 7: **for** each attribute a in data sample D **do**
 - 8: Forward Propagation:
 - 9: c_n = Convolution of extracted feature (f_e)
 - 10: m_x = Max pooling (c_n)
 - 11: f_c = Fully connected neural network (m_x)
 - 12: s_f = Soft max activation function(f_c).
 - 13: Backward Propagation:
 - 14: Backwards propagation with adam optimizer.
 - 15: Repeat weight w_i convergences.
 - 16: **return** Class for the email data sample.
-

ReLU $f(x)$ stands for rectified linear unit. It is the most implemented activation function. It works in the intermediate layer. The range of the function is between zero to infinite. It has less computation as compared to other activation functions. ReLU is expressed in mathematical form by Eq. 6.

$$f(x) = \max(0, x) \quad (6)$$

It gives x as output when the value of x is positive otherwise zero.

Pooling is the process of extracting an essential point from the window which has the necessary information. There are different ways to implement pooling, which is max, min and avg pooling. In this paper, we have used max pooling; it gives better results. For example, if we have feature map (fm) and the window size (wz), then the total number of features is fetched for the next round as shown in Eq. 7.

$$(F_{total})_{next} = \frac{fm}{wz} \quad (7)$$

Mathematically, the pooling function works as follows: suppose, $f_m = 1000$, and $w_z = 4$ then, the pooling function fetched $1000/4 = 250$ features for the next round process.

3.4 Phase 4: Result Classification

Flatten is the process of converting a multidimensional matrix ($n*m$) into a single feature vector (matrix) ($1*nm$). Mathematically, the flatten process is represented by matrix 8.

$$\begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1m} \\ F_{21} & F_{22} & \cdots & F_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1} & F_{n2} & \cdots & F_{nm} \end{bmatrix} \rightarrow [F_{11} \ F_{12} \ \cdots \ F_{nm}] \quad (8)$$

Fully connected is the feed-forward neural network. It is connected with the final classification is known as a fully connected layer. It takes the input from the previous layer and produces the class of the problem, whose size is the same as one-dimensional array.

Final layer activation functions are sigmoid, softmax, etc. Here, we used the sigmoid (S) activation function, which gives a better outcome. The range of a sigmoid function is between zero to one. It works for a two-class problem. Mathematically, sigmoid activation function is representation by Eq. 9.

$$S = \frac{1}{1 + \exp^{-x}} \quad (9)$$

where x is the input value. It ranges in between -2 and $+2$. It means when we slightly change in x , it will also contribute to wide fluctuations in the S .

4 Implementation Results and Performance

We have developed the proposed model for the detection of email spam. This prototype is implemented in a Jupyter Notebook (version 6.0.3) using Python language (version 3.7.7). For this, first of all, we have downloaded the data set. Data set is downloaded from UCI repository [33] named as spam base data set. The data set attribute is multivariate, and the values of those attributes are either numerical or real. The data set is categorized into two parts; first one is spam, and the other one is non-spam. The data set contains 4601 rows (instances) and 57 columns (attributes). In this data set, 978 spam emails and 3623 non-spam emails are shown in Table 1.

Table 1 Data set classification

Data set	No. of Email	% of Email
Spam	978	21.26
Non-Spam	3623	78.74
Total	4601	100

Table 2 Training and testing data values

Data set	Percentage (%)	Value
Training	70	3220
Testing	30	1381
Total	100	4601

Table 3 Confusion matrix of different classifiers

Classifier	T_P	F_P	T_N	F_N
MNB	362	196	608	215
KNN	393	124	680	184
SVM	456	160	644	121
DT	509	68	736	68
RF	512	29	775	65
MLP	499	22	782	78
CNN (Proposed model)	516	30	774	61

For training purposes, we have used 70% (3220) of the data set and the rest of the 30% (1381) data set were used for testing purposes described in Table 2.

The data set is also implemented on the existing ML techniques. The ML models includes multinomial naive Bayes (MNB) [34], k-nearest neighbour (KNN) [35], support vector machine (SVM) [36], decision tree (DT) [37], random forest (RF) [10] and multi-layer perceptron (MLP)[38]. In our proposed model, the true positive rate (T_P) is 516, which is more significant than other ML models. False positive (F_P) rate is 30, which is less than a different ML model. True negative (T_N) value is 774, which is high as compare to other models, and false negative (F_N) is 61, which is low as compare to different ML models described in Table 3.

We have calculated the performance of our model and compared the model with the existing models. The performance is calculated into two terms: performance in terms of detection rate and performance in terms of error rate. The performance in terms of detection rate includes Accuracy (Acc) [Equation (10)], Precision (P) [Equation (11)], Recall (R) [Equation (12)] and F-Score [Equation (13)]. The performance in terms of error rates includes Co-relation coefficient (cr) [Equation (14)], Mean Absolute Error (MAE) [Equation (15)], Root Mean Square Error (RMSE) [Equation

Table 4 Performance measurement metric and their computation

Evaluation matrix	Description	Used formula
Acc	Accuracy is the sum of T_P and T_N divided by total value (T)	$Acc = \frac{T_P + T_N}{T}$ (10)
P	Precision is the Value of T_P divided by actual positive (A_P)	$P = \frac{T_P}{A_P} * 100$ (11)
R	Recall is the percentage of T_P divided by predicted Result (P_R)	$R = \frac{T_P}{P_R} * 100$ (12)
F-Score	F-Score is harmonic mean of precision and recall. It is also called F1 Score or F-Measure	$F - Score = \frac{2*P*R}{P+R}$ (13)
cr	It is the relationship between the two relative variable which values lies in between -1 to +1	$cr_{xy} = \frac{s_{xy}}{s_{xxy}}$ (14)
MAE	It is error among paired observations that convey a certain hypothesis	$MAE = \frac{1}{n} \sum_{i=1}^n Y_i - \hat{Y}_i $ (15)
RMSE	It is the square root mean of the variance between expected results and real ones	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$ (16)
RRSE	It is ratio of RMSE and MAE	$RRSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n Y_i - \hat{Y}_i }}$ (17)

Table 5 Detection rate comparison of proposed model with other models

Classifier	Acc	Precision	Recall	F-Score
MNB	70.23	64.87	62.73	63.78
KNN	77.69	76.01	68.11	71.84
SVM	79.65	74.02	79.02	76.44
DT	90.15	88.21	88.21	88.21
RF	93.19	94.63	88.73	91.59
MLP	92.75	95.77	86.48	90.89
CNN (Proposed model)	93.41	94.50	89.42	91.89

(16)] and Root Relation Square Error (RRSE) [Equation (17)]. All the performance matrices and their computation mechanism are explained in Table 4.

In Table 5, the values of performance show a better detection rate of the proposed model as compared to the previous ML model. The proposed model Acc is 93.41, P is 94.50, R is 89.42, and F-Score is 91.89, and these values are more significant than other ML models. So, we can say that our proposed model finds a better detection rate.

Table 6 Error comparison of proposed model with other models

Classifier	cr	MAE	RMSE	RRSE
MNB	0.38	0.29	0.54	1.10
KNN	0.53	0.22	0.47	0.95
SVM	0.58	0.20	0.45	0.91
DT	0.79	0.09	0.31	0.91
RF	0.86	0.06	0.26	0.52
MLP	0.85	0.07	0.26	0.54
CNN (Proposed model)	0.86	0.06	0.25	0.52

In Table 6, the CNN proposed model cr is 0.86, MAE is 0.06, RMSE is 0.25, and RRSE is 0.52. This shows our model error rate is less as compared to other ML models.

5 Conclusion

Due to the importance of emails in business applications, we have reviewed different literature works to detect spam emails. In the literature, we have found several ML techniques that spam email detection rate is not satisfactory. To overcome such a problem, a CNN model based on deep learning techniques has been developed. For the development of the model, a data set has been downloaded from the UCI machine learning. The proposed models are implemented on the downloaded data set. We have also performed existing ML techniques on the downloaded data set. The achieved results in terms of detection rate as well as the error rate show the CNN model has a high detection rate and less error rate as compared to the existing implemented ML model. However, the proposed model's limitation is lack of plugin notification which needs continual updates for the application to detect zero-hour attack.

References

1. Dada EG, Bassi JS, Chiroma H, Abdulhamid SM, Adetunmbi AO, Ajibuwu OE (2019) Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon* 5(6):01802–01825
2. Fonseca O, Fazzion E, Cunha I, Las-Casas PHB, Guedes D, Meira W, Hoepers C, Steding-Jessen K, Chaves MHP (2016) Measuring, characterizing, and avoiding spam traffic costs. *IEEE Internet Comput* 20(4):16–24
3. Singh A, Chatterjee K (2017) Cloud security issues and challenges: a survey. *J Netw Comput Appl* 79:88–115
4. Chaudhary S, Singh A, Chatterjee K (2019) Wireless body sensor network (wbsn) security and privacy issues: a survey. *Int J Comput Intelligence IoT* 2(2):1–7

5. Androutsopoulos I, Koutsias J, Chandrinou K, Palioras G, Spyropoulos C (2000) An evaluation of naive bayesian anti-spam filtering. In: Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML), vol 0006013, pp 9–17
6. Torabi Z, Nadimi-Shahraki M, Nabiollahi A (2015) Efficient support vector machines for spam detection: a survey. (*IJCSIS*) Int J Comput Sci Information Security 13(1):11–28
7. Ndumiyana D, Magomelo M, Sakala L (2013) Spam detection using a neural network classifier. *Online J Phys Environ Sci Res* 2:28–37
8. Thirumuruganathan S (2010) A detailed introduction to K-nearest neighbor (KNN) algorithm. <http://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knnalgorithm//>. Online; Accessed 19 Mar 2020
9. Pawlak Z (2012) Rough sets: theoretical aspects of reasoning about data, vol 9. Springer Science & Business Media
10. DeBarr D, Wechsler H (2009) Spam detection using clustering, random forests, and active learning. In: Sixth conference on Email and anti-spam. Mountain View, California, pp 1–6
11. Barreno M, Nelson B, Sears R, Joseph A, Tygar D (2006) Can machine learning be secure? In: Proceedings of the 2006 ACM symposium on information, computer and communications security, pp 16–25
12. Kumar S, Arumugam S (2015) A probabilistic neural network based classification of spam mails using particle swarm optimization feature selection. *Middle-East J Sci Res* 23(5):874–879
13. Li D, Yu D (2014) Deep learning: methods and applications. *Found Trends Signal Process* 7(3):197–387
14. Roy PK, Singh JP, Banerjee S (2020) Deep learning to filter sms spam. *Future Generation Comput Syst* 102:524–533
15. Roy PK (2020) Multilayer convolutional neural network to filter low quality content from quora. *Neural Process Lett* 1–17
16. Lueg CP (2005) From spam filtering to information retrieval and back: seeking conceptual foundations for spam filtering. In: Proceedings of the American Society for Information Science and Technology, vol 42, issue 1, pp 1–3
17. Wang X-L, Cloete (2005) Learning to classify email: a survey. In: 2005 international conference on machine learning and cybernetics, vol 9, pp 5716–5719
18. Li W, Zhong N, Yao YY, Liu J, Liu C (2006) Spam filtering and email-mediated applications. In: International workshop on web intelligence meets brain informatics, pp 382–405
19. Alexy B, Shyamanta H (2018) E-mail spam filtering: a review of techniques and trends. In: Advances in electronics, communication and computing, pp 583–590
20. Laorden C, Ugarte-Pedrero X, Santos I, Sanz B, Nieves J, García Bringas P (2014) Study on the effectiveness of anomaly detection for spam filtering. *Information Sci* 277:421–444
21. Saleh A, Karim A, Shanmugam B, Azam S, Kannoopatti K, Jonkman M, Boer F (2019) An intelligent spam detection model based on artificial immune system. *Information* 10:209–225
22. Cohen Y, Gordon D, Hendler D (2018) Early detection of spamming accounts in large-scale service provider networks. *Knowledge-Based Syst* 142:241–255
23. Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. *Ann Math Artif Intelligence* 85(1):21–44
24. Moradpoor N, Clavie B, Buchanan W (2017) Employing machine learning techniques for detection and classification of phishing emails, pp 1–8
25. Kong Y, Yu T (2018) A deep neural network model using random forest to extract feature representation for gene expression data classification. *Sci Rep* 8:1–9
26. Sharma R, Kaur G (2016) E-mail spam detection using svm and rbf. *Int J Modern Educ Comput Sci* 8:57–63
27. Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. *Expert Syst Appl* 36(7):10206–10222
28. Dara S, Tumma P (2018) Feature extraction by using deep learning: a survey. In: 2018 second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp 1795–1801

29. Leeghim H, Seo I-H, Bang H (2008) Adaptive nonlinear control using input normalized neural networks. *J Mech Sci Technol* 22(6):1073–1083
30. Mazilu S, Iria J (2011) L1 vs. L2 regularization in text classification when learning from labeled features. In: 2011 10th international conference on machine learning and applications and workshops, vol 1, pp 166–171
31. Pennington J, Socher R, Manning C (2014) GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1532–1543
32. Jiang K, Feng S, Song Q, Calix R, Gupta M, Bernard G (2018) Identifying tweets of personal health experience through word embedding and lstm neural network. *BMC Bioinformatics* 19:68–84
33. Hopkins GHJS, Reeber E (1999) UCI Machine Learning Repository: Spambase data set. <https://archive.ics.uci.edu/ml/datasets/spambase>. Online; Accessed 01-03-2020
34. Rish I (2001) An empirical study of the naïve bayes classifier. *Int J Comput Intelligence Appl* 2001 Work Empir Methods Artif Intell 3:41–46
35. Firte L, Lemnaru C, Potolea R (2010) Spam detection filter using knn algorithm and resampling. In: Proceedings of the 2010 IEEE 6th international conference on intelligent computer communication and processing, pp 27–33
36. Amayri O, Bouguila N (2010) A study of spam filtering using support vector machines. *Artif Intell Rev* 34:73–108
37. Chakraborty S, Mondal B (2012) Spam mail filtering technique using different decision tree classifiers through data mining approach—a comparative performance analysis. *Int J Comput Appl* 47:26–31
38. Goh KL, Singh A, Lim KH (2013) Multilayer perceptrons neural network based web spam detection application, pp 636–640

Chapter 21

Sentiment Analysis Algorithms: Classifiers and Their Comparison



Shubham Joshi, Rochit Dubey, Aryav Tiwari, and Poonam Jindal

1 Introduction

Online discussion forums, viewer reviews and personal blogs establish communication among consumers and between consumers and firms of the Motion Picture Industry. ‘What to watch next?’ is discovered through the platform’s recommendation system. When people think that they ‘choose’ their next watch content, they are basically choosing from a number of decisions made by machine learning algorithms. A tool which helps uncover people’s opinions and emotions about a particular service or product is called sentiment analysis. Mining out the views, opinions, sentiments, emotions and attitude from the vast pool of data comprising blogs, reviews, tweets, news or comments is opinion mining, also termed as sentiment analysis or subjectivity analysis.

The application of sentiment analysis is extensive in the field of competitive marketing as it minimizes the need for a system of direct feedback, and so is faster. This ensures that the firms may respond timely to fluctuating trends of public opinion. [1] shows the classification of different types of data into positive and negative classes using machine learning algorithms, and this practice of classification is termed as sentiment analysis or emotion detection.

The classifiers and algorithms used in this paper are—K nearest neighbours, logistic regression, support vector machine, Gaussian Naive Bayes, Bernoulli Naive

S. Joshi (✉) · R. Dubey

Electronics and Communication Department, National Institute of Technology, Kurukshetra, India

A. Tiwari

Electronics and Communication Engineering Department, National Institute of Technology, Kurukshetra, India

P. Jindal

Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, India

Bayes and multinomial Naive Bayes. These algorithms can train the machine depending upon the type of data and have different approaches to do so. They are applied in the field of both regression and classification.

We have made improvement upon the work done in [1] and included parameters other than accuracy, namely precision score and training time. The calculation of precision score and accuracy is discussed in later parts of this paper.

2 Dataset and Preprocessing

Further continuing the approach used in [1] for movie reviews, we have used the Internet Movie Database, i.e. IMDb dataset present in the Keras library, and this dataset consists of a total of 50,000 reviews out of which 25,000 reviews are present in the training set while the rest of the 25,000 reviews are present in the test set. Each review consists of a large number of unique words. For easy analysis and comparison, we have limited the number of training set reviews to 6000 and test set reviews to 2500 making a total of 8500 reviews. Each review consists of 1000 unique words that occur most frequently in these reviews. These words will be kept in a vocabulary where each word is mapped to an index. Each sentence is then represented using a vector, and the vector consists of values 1 if the word from vocabulary is present in the sentence whose vector is created, else the value is 0.

Since raw data tend to be full of noise, test classification algorithms are applied over preprocessed dataset, in order to minimize overfitting and give a better generalization accuracy. Stemming, tokenization and stop word removal are examples of frequently applied steps used to preprocess data.

3 Classifiers

3.1 *K-Nearest Neighbours*

KNN is what is described as a lazy learning algorithm, hence no training phase [2]. It gets computationally costly and tedious when data are multidimensional [3]. Being a nonparametric method, KNN suffers from the curse of overfitting, cannot generalize the results well and finally predicts poorly [4]. KNN can perform exceptionally well if the number of features is less as the training time required for it is low as compared to other classifiers.

3.2 *Naive Bayes*

Naive Bayes (NB) as a classifier is generally recommended for datasets with the smaller size as it does not overfit because of the presence of inherent regularization in it. In opinion mining, the Bayesian network is used frequently because of its high computational complexity. We expect it to perform well for the data which has words which are strong (i.e. carry a straightforward meaning and relation with respective class). Naive Bayes also performs well when the features present in the text do not depend on each other; i.e. they are independent [5]. The basis for this classifier is Bayes theorem. It has three types:

Bernoulli NB (BNB): It is based on Bernoulli distribution where each feature is assigned 0 or 1 value for whether it is present or not.

Gaussian NB (GNB): It is based on the normal distribution, and in this data can be modelled using mean and standard deviation.

Multinomial NB (MNB): Unlike Bernoulli distribution in multinomial, instead of 0 or 1 value the frequency of the feature is assigned for computation.

3.3 *Support Vector Machines (SVMs)*

SVM is very effective for various text problems such as classification of news articles and sentiment prediction [6, 7] as they have the capability to deal well with high dimensionality. The underlying principle is classifying the points based on the best hyperplane separating them. Unlike the other classifiers which have a higher capacity to fit the data (training data), SVM generalizes better and is less likely to overfit data [8].

3.4 *Logistic Regression*

Logistic regression is a linear algorithm and is named after a logistic function which forms the core of this algorithm. The logistic function which is also called a sigmoid function is an S-shaped curve which maps a real-valued number to a value between 0 and 1. The difference between linear and logistic regression is that in the case of logistic regression the output value is either 0 or 1. For probability greater than 0.5, output is 1 else 0.

4 Flow Chart

Figure 1 shows the process followed in comparing different classifiers on parameters

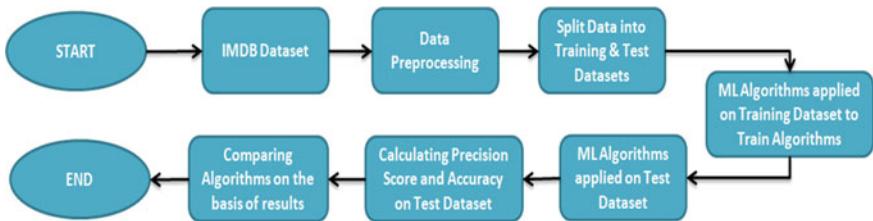


Fig. 1 Flow chart of process followed to establish a comparison between the classifiers

like accuracy, precision score and training time on the IMDb dataset. The process started with the selection of the dataset and preprocessing steps such as tokenization, stop-word removal etc. were performed on the data. Further for training the model and later testing it, data was divided into train and test sets. Then, all the classifiers stated in the paper were applied first on the training data to train the algorithm and then on the test data one after another, results were noted for certain parameters such as accuracy and precision score for the test dataset, and in the end according to the scores obtained previously the algorithms were compared.

5 Simulation Environment

The primary programming language used for much of the work and research in the field of machine learning is Python and that has been used. The environment of Jupyter Notebook developed by ‘Project Jupyter’, a non-profit organization, is preferred. Jupyter Notebook is an open-source Web-based application that allows the user to not only create but also share documents containing code, equations and visualization. We have used Python in it for getting dataset, preprocessing dataset and evaluation of classifiers on various parameters using libraries present in Python.

6 Comparison Parameters

Before defining the parameters used for comparison, we will be discussing the values which are necessary for calculating the parameters on the basis of which different algorithms are compared.

True Positives (TP): This includes the correctly predicted positive values which mean both the actual and the predicted values are yes, e.g. if actual class value tells that the passenger has survived and predicted class tells you the same thing.

True Negatives (TN): This includes the correctly predicted negative values which mean both the actual and the predicted values are no., e.g. if actual class value tells that the passenger has not survived and predicted class tells you the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP): When an actual class has a value no, the predicted class has a value yes, e.g. if the actual class has a value that a particular passenger did not survive while the predicted class has a value that this passenger will survive.

False Negatives (FN): When an actual class has a value yes, the predicted class has a value no, e.g. if the actual class has a value that a particular passenger survived while the predicted class has a value that this passenger will not survive.

6.1 Accuracy

It is one of the most intuitive parameters, and it can simply be defined as the ratio of total correct predictions to the total number of predictions. It may appear that we can get all the information about the classifier using this parameter but this is well and good on the symmetric data when the number of false positives and the number of false negatives are almost the same. It can be characterized by the given formula

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (1)$$

6.2 Precision

This is the parameter which tells us out of all the predicted values what ratio out of them is correctly predicted, so it can be defined as the ratio of true positives to sum of true as well as false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

6.3 Training Time

This can be defined as the time as for which the system has to be trained, i.e. on the training data after which we can start the predictions on it for the test data.

7 Performance Analysis

The classification was made using different algorithms such as KNN, SVM, NB and LR, they are scored based on certain parameters, and the ones used here are accuracy, precision score and training time. There appeared to be no elite algorithm, the results vary on the basis of training and test split, and also a trade-off is present between training time and accuracy. The results obtained from the comparison of these algorithms are stated below.

7.1 Training Time

See Table 1.

Table 1 ‘Training time’ of classifiers for the same training dataset

Classifier	Training time (s)
KNN	0.551
Logistic regression	0.371
Multinomial Naive Bayes	0.029
Gaussian Naive Bayes	0.198
Bernoulli Naive Bayes	0.134
SVM	54.029

7.2 Training Set Results

Table 2 ‘Accuracy’ and ‘precision score’ of classifiers on the training dataset

Classifiers	Accuracy (%)	Precision score (%)
KNN	77.38	71.654
Logistic regression	91.2	90.09
Multinomial Naive Bayes	84.66	84.75
Gaussian Naive Bayes	83.4	81.15
Bernoulli Naive Bayes	83.33	82.16
SVM	92.41	92.09

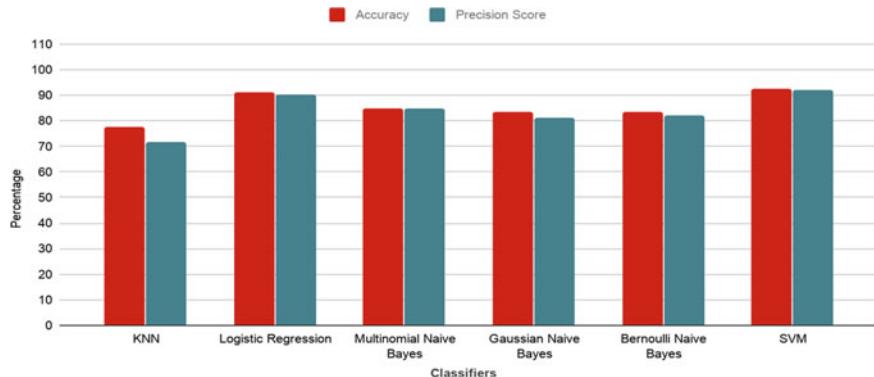


Fig. 2 Comparison of ‘accuracy’ and ‘precision score’ of classifiers on the training dataset

7.3 Test Set Results

The algorithms were trained on a training set which consisted of 6000 movie reviews, while the test set had about 2000 reviews. Accuracy and precision score were calculated for both training set and test set, and the results are shown below both in tabular form in Table 3 as well as in graphical form in Fig. 2. As shown in Table 1, KNN performs much faster with training time 0.551 s but shows an accuracy of just 61.36% on the test set as shown in Table 3. The classifier that was successful in achieving the best out of the trade-off between training time and accuracy score was multinomial Naive Bayes. It achieved a practicable accuracy of 83.96% within training time at 0.371 s. Multinomial Naive Bayes also has the best precision score of 82.76% among all the algorithms (Fig. 3).

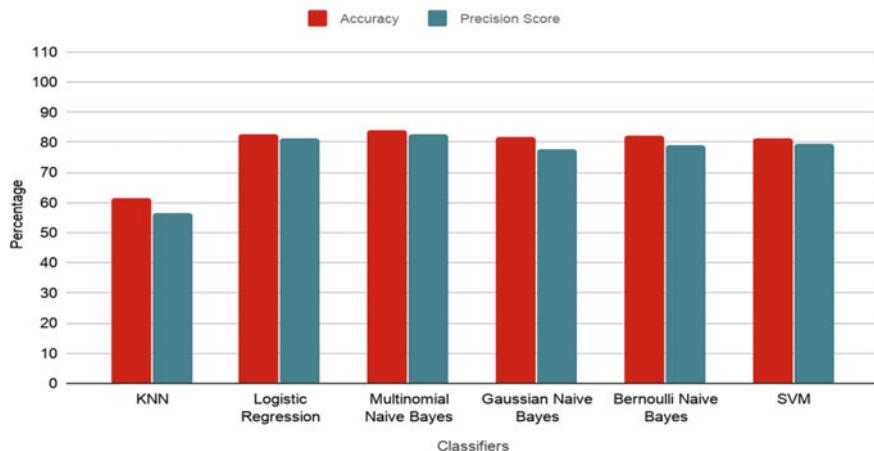


Fig. 3 Comparison of ‘accuracy’ and ‘precision score’ of classifiers on test dataset

Table 3 ‘Accuracy’ and ‘precision score’ of classifiers on test dataset

Classifiers	Accuracy (%)	Precision score (%)
KNN	61.36	56.47
Logistic regression	82.56	81.13
Multinomial Naive Bayes	83.96	82.76
Gaussian Naive Bayes	81.922	77.94
Bernoulli Naive Bayes	82.12	79.12
SVM	81.36	79.46

8 Conclusion

Taking a closer look at the results obtained on comparing different algorithms on different parameters, certain observations can be drawn out; as far as KNN is concerned, it requires no training time being a lazy learning algorithm, but does not have that good of the accuracy and performs poorly on multidimensional data. SVM requires a higher training time and overfits most of the time, hence making a poor generalization, but gives good accuracy in our case but cannot be generalized for future datasets. Logistic regression works in a decent manner for both training and test datasets as shown in Tables 2 and 3, respectively. It has a higher accuracy and precision score when compared to KNN; however, when compared to MNB it takes more time for training with almost the same accuracy and precision score (when compared on test dataset) and this makes it less preferable to MNB. MNB works best for our dataset as it achieves the best out of trade-off between training time and accuracy, and it also gives a good precision score without overfitting data. However, there is no best classifier as different classifiers work differently with different datasets and the performance may vary depending on various properties of dataset like its size or length [9]. So, the classifier to be used has to be selected according to the dataset used as well as the parameter (like accuracy, training time, precision score, etc.) we wish to improve upon. So, we have to decide the classifier to be used depending on the dataset used and the parameter which is most critical for our analysis. A summary of the analysis is given in Table 4.

Table 4 Performance of classifiers on test dataset

Accuracy	KNN performs poorly in accuracy, while the rest of the algorithms have comparable performances. MNB being the best of them all
Precision score	In precision score performance, the gap among classifiers is wider than that of accuracy. MNB is the best performer in this case
Training time	Training time brings in the dimension of practical usability. The use of SVM despite being a good performer in accuracy and precision score is restricted by higher training time

9 Future Scope

Considering all aforementioned classifiers, all lack in one aspect or another. The intricacies of language still baffle the classifiers, for instance, sarcastic comments or triple or quadruple negatives for that matter. The use of certain words pertaining to an industry might convey an altogether meaning in another, malarkey also poses as a hindrance in classification even with stop word removal, and it does not suffice sometimes. The cruciality of this newly emerging field serves as a boon as well as a curse; recently, it has shown the power to influence presidential elections but it poses a serious threat too. There is no predefined way to separate real review from a fake one, and if a fake review is presented masquerading as a real one the consequences can prove to be dire. Until now, sentiment analysis was a problem of duality: people either like it or hate it. But now the traditional approach is being replaced by considering the grey area in between. So, instead of segregating it into two groups of positive and negative, classification is also being done in the spectrum between positive and negative classes. Companies may carry out user feedback as follows:

1. Enjoyed it a lot!
2. It was good.
3. Neutral
4. Not good.
5. Don't want to see that again.

The feedback received will be used to sort the product, movie, etc., on the basis of popularity. This is not a binary classification problem as the user review can be classified in more than two classes. This multi-class classification can also be used to give star ratings to a given product on the basis of the reviews given to it; in this case, the user is not asked to manually select the star rating but the algorithm itself will deduce the number of stars to be given by analysing the reviews provided by various users (Fig. 4).

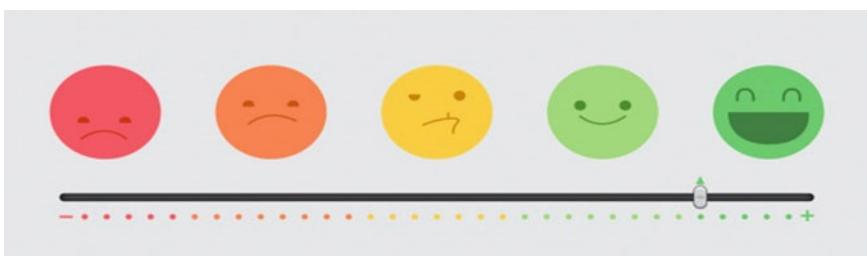


Fig. 4 Scale showing varied customer experiences

References

1. Hartmann J, Huppertz J, Schamp C (2019) Heitmann M (2019) Comparing automated text classification methods. *Int J Res Mark* 36(1):20–38. <https://doi.org/10.1016/j.ijresmar.2018.09.009>
2. Yang Y (1999) An evaluation to statistical approaches to text categorization. *Inf Retrieval*. [\(Springer\)](https://doi.org/10.1023/A:100998220290)
3. Aggarwal CC, Zhai CX (2012) A survey of text classification algorithms. In: Mining text data, vol. 9781461432234. Springer US, pp 163–222. https://doi.org/10.1007/978-1-4614-3223-4_6
4. Bellman RE (1961) Adaptive control processes. Princeton University Press, Princeton, NJ
5. Domingos P, Pazzani M (1997, November) On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.* (1997) [\(Springer\)](https://doi.org/10.1023/A:1007413511361)
6. Pang B, Lee L, Vaithyanathan S (2002, July) Sentiment classification using machine learning techniques. In: EMNLP'02: proceedings of the ACL-02 conference on empirical methods in natural language processing, vol 10, pp 79–86. <https://doi.org/10.3115/1118693.1118704>
7. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (Eds) Machine learning: ECML-98. ECML 1998. Lecture notes in computer science (Lecture notes in artificial intelligence), vol 1398. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/BFb0026683>
8. Bennett KP, Campbell C (2000) Support vector machines: hype or hallelujah? In: ACM SIGKDD explorations newsletter (2000, December). <https://doi.org/10.1145/380995.380999>
9. Bermingham A, Smeaton AF (2010, October) Classifying sentiment in microblogs: is brevity an advantage? In: CIKM'10: Proceedings of the 19th ACM international conference on Information and knowledge management, pp 1833–1836. <https://doi.org/10.1145/1871437.1871741>

Chapter 22

Email Sentiment Classification Using Lexicon-Based Opinion Labeling



Ulligaddala Srinivasarao and Aakanksha Sharaff

1 Introduction

Sentiment analysis (SA), also termed as opinion mining, refers to computational linguistics, text analysis, and natural language processing (NLP) for identity and extracting personal information. SA is extensively applied to social media and reviews for different applications like marketing, customer service, etc. Sentiment analysis is to find out a person's attitude related to a topic or a document [1]. Sentiment analysis classifies the polarity of any given content. Also, the states of emotion like "sad," "happy," fear," etc., can be found out. There are various challenges in sentimental analysis. Opinion word is one which may be considered as positive in one case and harmful in another case. The second issue is that the opinion of people will not be the same all the time. Traditional text processing is mostly dependent on the difference between the two pieces of the word. The sentence "the picture was nice" is not similar to "the picture was not nice" in sentiment analysis. People will contradict their opinions. People give negative and positive comments that can be managed by examining the sentences one at a time. People convey ideas in various informal media such as Facebook, blogs, Twitter, and Amazon that are very hard to understand by the machine but are readable by humans.

In four years between 2011 and 2015, there is a surge in the number of email users by 3%. Out of all the emails, business emails are mostly exchanged every day over 108.7 billion emails. This is because the email is the best way for business. When people communicate through emails, it becomes unavoidable to reveal the

U. Srinivasarao (✉) · A. Sharaff

Department of Computer Science and Engineering, National Institute of Technology Raipur,
Raipur, Chhattisgarh, India

e-mail: usrinivasarao.phd2018.cs@nitrr.ac.in

A. Sharaff

e-mail: asharaff.cs@nitrr.ac.in

feeling sun intentionally or intentionally. The sentimental analysis is performed on a big dataset of business emails; it may reveal the critical information beneficial for business. Various opinion mining and sentiment analysis-based applications have been made in the last decade, where the algorithms and techniques were initially developed. Some improved methods have also proposed applications [2]. Excellent research has been conducted in sentiment analysis, but the research over the email sentiment analysis is minimal. So, email sentiment analysis has been performed for unlabeled data.

In this paper, two prevalent feature extraction techniques, namely count vectorizer and term frequency and inverse document frequency (TF-IDF) techniques, have been used. For the pre-labeling process, the method, including opinion lexicon labeling, has been used to classify sentiment labeling. As for sentiment classification algorithms, XGBoost, gradient boosting, and random forest classifiers have been tested. Out of these two classification techniques, XGBoost performs consistently well when classifying email sentiments.

2 Related Work

In Liu et al. sentiment analysis, a study of extracting and analyzing the implications of attitudes, emotions, and opinions from NLP has attracted researchers from diverse areas [3]. Salah and Gayar [4] proposed a hybrid sentiment analysis technique by combining support vector machine (SVM) classifier algorithms with VADER lexicon labeling. In [5], Bogawar and Bhoyar have classified the sentiments of different data mining methods such as fuzzy c-means clustering and k-means clustering. Liu and Lee [6] have proposed a new direction to solve sentiment analysis tasks/here, the email sentiment pattern has been recognized using a trajectory representation.

In [7], Gupta et al. conveyed that emotion mining for customer care services will be helpful for marketers to gain information regarding satisfaction levels of their customers to improve their services. Nanda et al. [8] have performed sentiment analysis of movie review in the Hindi language by using the random forest (RF) classifier along with SVM. In [9], Rathor et al. illustrated that methodology based on SVM outperforms the other ways when the SA is performed on a dataset of reviews related to Amazon products. Liu and Lee have shown in [10] that SVM is the better choice for sentiment classification. Mamgai et al. developed an automatic question-answering in video lectures [11]. In [12], Sharaff and Nagwan performed a new study to solve the problem in two steps. First, a clustering approach is used to cluster the emails, and then the email threads are identified. Sharaff and Srinivasarao proposed a relationship between the words in the content and subject-based emails used [13]. In this research, including opinion lexicon labeling, the English opinion lexicon has been used to classify sentiment labeling. As for sentiment classification algorithms, XGBoost, gradient boosting, and random forest classifiers have been tested.

3 Methodology

Figure 1 shows the proposed sentiment analysis methodology applied to the unlabelled email dataset. The methodology consists of different techniques, which include data extraction, preprocessing, feature extraction (FE), sentiment lexicon (SL), and sentiment classification (SC). Preprocessing incorporates the following three steps: tokenization, stemming, and stop word removal. Count vectorizer approach and TF-IDF are utilized as features. For the pre-labeling process, the method, including opinion lexicon labeling English opinion lexicon, has been used for classification of sentiment. As for sentiment classification algorithms, XGBoost, gradient boosting, and random forest classifiers are tested. The complete methodology has been explained in the subsequent subsections.

3.1 Preprocessing

Generally, email data is noisy and unstructured. So, preprocessing is very much important to generate the related data and then use the refined emails in the experiments [14]. The general preprocessing steps used in text mining are tokens, stemming, and stop word removal. Email data refining is difficult due to the drawback of noisy data, like quotations and headers that may be refined. The following preprocessing steps are used in this paper: (1). Tokenization: Tokenization breaks the text into

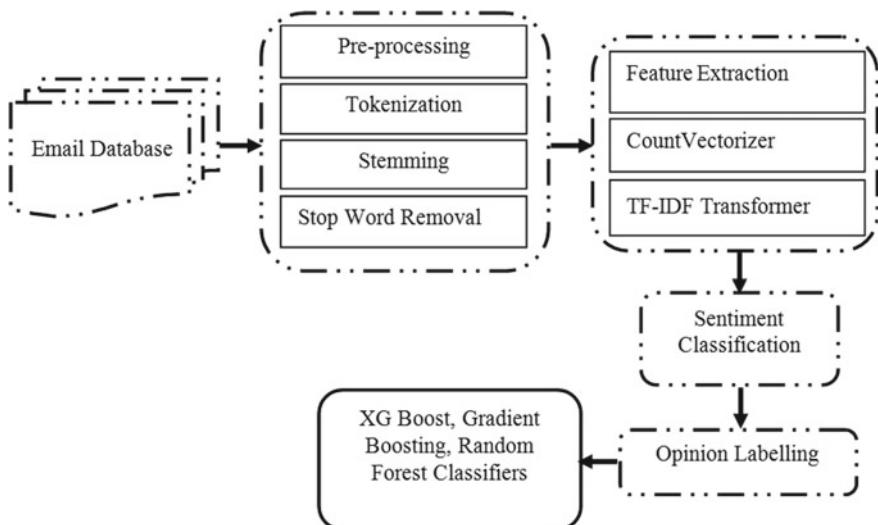


Fig. 1 Proposed method

symbols, phrases, and words. (2). Stemming: Stemming has been used for identifying the stem or root of a word. (3). Stop word removal: Stop words are a part of natural language. The importance of removing the stop words is that they make the text look less important and also heavier for the analysts. This will reduce the term space dimensionality.

4 Feature Extraction

In [15], Fattah proposed feature selection methods that are useful to classify the text and also for sentiment analysis. These methods will rank the features depending upon some measures such that unwanted features will be removed by storing the most important features for improving the accuracy. As the email sentiment analysis-related research is limited, a more useful feature selection method should be explored further. In this paper, the count vectorizer approach and TF-IDF are utilized as feature extraction methods.

4.1 Count Vectorizer

Count vectorizer works on term frequency; i.e., it counts the number of times the tokens occur and then builds a sparse matrix of documents x tokens.

- **TF-IDF Transformer:** The word frequency will be rescaled on the basis of frequency of occurrence of a word in all the documents. Because of this, the scores corresponding to the frequent words are also frequent among all the documents are reduced. This is called TF-IDF. It is the score of the words among all the documents. Further, both TF and IDF are combined for producing a final score of a term t in document d, as shown in Table 1.

Table 1 TF-IDF values

Words	TF * IDF
Access	0.34641
John	0.35641
Gadd	0.23094
Susan	0.23094
Name	0.11547
Thank	0.11647
Task	0.10547

Table 2 Opinion words list

Positive words	Negative words
Large, clean, cool, strong, safe, full, nice, young	Dad, danger, rundown, dark, random, slow, noisy, torn

5 Sentiment Classification

Li et al. proposed [16] sentiment classification, based on a manually built dictionary that consists of thousands of positive and negative sentimental words and then adopts a term-counting approach to incorporate polarity shifting information. As the dataset considered is unlabeled, it is required to have a pre-labeling method to provide training and testing data for performing the classification task. Opinion lexicon labeling approach has been considered.

5.1 Opinion Lexicon Labeling

In opinion lexicon, words are normally defined and are used for opinion expressions (OE). Positive words include like, excellent, happy, nice, etc. The negative words list included are like, hated, bad, dark, etc. This paper uses the English opinion lexicon, particularly email SA. The list consists of 20,006 and 4784 positive and negative words, respectively, as shown in Table 2. Let PL be collection of positive word list, NL be set of negative word list, and A be set of attribute generated from E , comprising feature $a_1, a_2, a_3, \dots, a_i$.

$$a_i = \text{frequency} * e_n (e_n \in PL \cup NL) \quad (1)$$

For each feature attribute a_i , a minimum–maximum normalization is

$$\text{Normalized}_{(a_i)} = \frac{a_i - \text{Min}(a_i)}{\text{Max}(a_i) - \text{Min}(a_i)} \quad (2)$$

6 Experimentation Results

Enron email corpus is largely adapted by different researchers to conduct experiments to classify the emails. This is a large, publicly available dataset. The Enron corpus dataset available at [1] consists of greater than 0.5 millions email messages that are retrieved from around 160 users. The basic idea of this study is classifying the sentiments in the email content. A total of 7000 email messages are considered as dataset. Next, features are extracted using count vectorizer and TF-IDF techniques.

The sentiment features are labeled using opinion lexicon for classifying into positive and negative or neutral polarities. These are further fed as inputs to different classifiers. For classifying, three models were used, namely XGBoost, gradient boosting, and random forest classifier. The classification report is generated using confusion matrix. The performance evaluation of different classifiers has been done by using precision, recall, *F*-measure, and accuracy. Precision is defining as the ratio of actual value (AV) neutral and predicted value neutral cases. Recall is defined as the ratio of actual value (AV) neutral and predicted value neutral cases. *F*-measure can be calculated by taking the harmonic mean precision and recall. Correctly classified rate is defined as accuracy.

$$\text{Precision} = \frac{\text{AvNe PvNe}}{\text{AvNe PvNe} + \text{AvP PvNe} + \text{AvN PvNe}}$$

$$\text{Recall} = \frac{\text{AvNe PvNe}}{\text{AvNe PvNe} + \text{AvNe PvP} + \text{AvNe} + \text{PvN}}$$

$$F\text{- measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{\text{AvNe PvNe} + \text{AvP PvP} + \text{AvN PvN}}{\text{Total messages}}$$

The best accuracy has been observed by XGBoost classifier, i.e., 78%. The next better results are produced by gradient boosting, i.e., 76%, and random forest 66%. The accuracy, precision, recall, and *F*-measure have been computed for the different classifiers, and XGBoost outperforms all. Final comparisons of various classification algorithms have been shown in Table 3.

Table 3 Comparison of classification algorithms

Classifiers	Sentiment	Accuracy	Precision	Recall	<i>F</i> -measure
XGBoost	Positive	0.78	0.81	0.88	0.84
	Negative		0.81	0.65	0.72
	Neutral		0.72	0.76	0.74
Gradient boosting	Positive	0.76	0.73	0.82	0.78
	Negative		0.77	0.65	0.71
	Neutral		0.79	0.76	0.78
Random forest	Positive	0.66	0.62	0.75	0.68
	Negative		0.62	0.69	0.65
	Neutral		0.78	0.53	0.63

7 Conclusion

The most commonly used communication medium in business is email. To understand the hidden information in email communication has a larger priority in business intelligence. This paper introduces an email sentiment analysis using count vectorization and TF-IDF methods as a feature extraction approach and combined opinion labeling method to classify the sentiment. A predefined opinion word list has been utilized. The evaluation of the proposed methodology has been performed using two experiments. A comparison between the two feature extraction methods, i.e., count vectorization and TF-IDF, confirms the option of the TF-IDF method as the default feature extraction method. A comparative study of three different classifiers, including XGBoost, gradient boosting, and random forest, justifies that XGBoost algorithm performs more accurately in the sentiment classification task.

References

1. Jagdale RS, Shirsat VS, Deshmukh SN (2016) Sentiment analysis of events from Twitter using open source tool. *Int J Comput Sci Mobile Comput* 5(4):475–485
2. Ravi K, Ravi V (2015) A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowl Based Syst* 89:14–46
3. Liu S, Lee I, Cai G (2016, October) Sentiment clustering with topic and temporal information from large email dataset. In: Proceedings of the 30th Pacific Asia conference on language, information and computation: posters, pp 363–371
4. Salah R, El Gayar N (2019) Sentiment analysis using unlabeled email data (No. 2080). EasyChair
5. Bogawar PS, Bhojar KK (2016, August) Soft computing approaches to classification of emails for sentiment analysis. In: Proceedings of the international conference on informatics and analytics, pp 1–7
6. Liu S, Lee I (2018) Discovering sentiment sequence within email data through trajectory representation. *Expert Syst Appl* 99:1–11
7. Gupta N, Gilbert M, Fabbriozio GD (2013) Emotion detection in email customer care. *Comput Intell* 29(3):489–505
8. Nanda C, Dua M, Nanda G (2018, April) Sentiment analysis of movie reviews in hindi language using machine learning. In: 2018 international conference on communication and signal processing (ICCPSP). IEEE, pp 1069–1072
9. Rathor AS, Agarwal A, Dimri P (2018) Comparative study of machine learning approaches for Amazon reviews. *Procedia Comput Sci* 132:1552–1561
10. Liu S, Lee I (2018) Email sentiment analysis through k-means labeling and support vector machine classification. *Cybern Syst* 49(3):181–199
11. Mamgai D, Brodiya S, Yadav R, Dua M (2019) An improved automated question answering system from lecture videos. In: Proceedings of 2nd international conference on communication, computing and networking. Springer, Singapore, pp 653–659
12. Sharaff A, Nagwani NK (2016) Email thread identification using latent Dirichlet allocation and non-negative matrix factorization based clustering techniques. *J Inf Sci* 42(2):200–212
13. Sharaff A, Srinivasarao U (2020, January) Towards classification of email through selection of informative features. In: 2020 first international conference on power, control and computing technologies (ICPC2T). IEEE, pp 316–320

14. Tang G, Pei J, Luk WS (2014) Email mining: tasks, common techniques, and tools. *Knowl Inf Syst* 41(1):1–31
15. Fattah MA (2017) A novel statistical feature selection approach for text categorization. *J Inf Process Syst* 13(5):1397–1409
16. Li S, Lee SYM, Chen Y, Huang CR, Zhou G (2010, August) Sentiment classification and polarity shifting. In: Proceedings of the 23rd international conference on computational linguistics. Association for computational linguistics, pp. 635–643

Chapter 23

Real-Time Facial Emotion Recognition Using Deep Learning



Shruti Chand, Apoorva Singh, Ria Bhatia, Ishween Kaur, and K. R. Seeja

1 Introduction

Facial emotions are critical requirements for non-verbal communication between human beings. It is only possible because human beings are able to distinguish feelings relatively accurately and effectively. The automated facial emotional recognition system is an essential component of human–computer interaction. Apart from the commercial uses of this automatic facial emotion recognition system, the model can also be used to develop cognitive processing insights of our brain. The process of emotional recognition from facial images depicting different and unique expressions is complex. Moreover, the inconsistency and changes in images due to various identities impede the efficiency of the model.

Deep learning methods have performed very well in the case of MNIST digit recognition dataset. The emotion recognition problem can be considered as similar to digit recognition mission. In this problem, corresponding digit labels, we have emotion labels. But the process of emotional recognition is much more complex, because digit images are much simpler than facial images depicting different and unique expressions. More so, the inconsistency and changes in images due to various identities impede the efficiency of the model. The objective of this research is to analyse different emotions represented by the human being in real time. This is done using a laptop webcam, where the person’s image is captured and emotions are recognized in real time. Emotions can be divided into 7 classes—happiness, sadness, fearful, disgust, angry, neutral and surprise. The key emphasis is on improving the

S. Chand (✉) · A. Singh · R. Bhatia · I. Kaur · K. R. Seeja

Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi 110006, India

K. R. Seeja

e-mail: seeja@igdtuw.ac.in

consistency of the method and using the most effective architecture for the various sets of images available in the dataset.

2 Literature Review

In the literature, there are various techniques of emotion analysis, especially deep learning-based techniques like deep belief networks [1] and convolutional neural network [2, 3]. The pretrained models like VGGNet, ResNet, GoogLeNet, and AlexNet [4] were also proposed for facial emotion recognition. There are both subject-dependent and subject-independent emotion recognition models [5]. Barros et al. [6] have used representation of the hierarchical features to deal with spontaneous emotions. Azcarate et al. [7] have highlighted a model to detect emotions using live video streams and sequences. They explained in detail about the motion units of faces in different expressions and their difference with each other. They have used tree augmented Naive Bayes (TAN) and Naive Bayes classifiers. Hybrid deep learning approach [8] that includes the convolutional neural network (CNN) for each frame and the long short-term memory (LSTM) for the next frames is also proposed. The combination provides analysis for individual as well as consecutive frames. Zhan et al. [9] developed an expression recognition system and included it into Multiplayer Online Games (MOGs) to monitor the facial expressions of humans. They classified the output into good mood, bad mood and surprise. Zhang et al. [10] used fuzzy multi-class support vector machines (FSVMs) as classifiers and compared it with multiclass SVM. The author focused on handling outliers and noises present in human faces by analysing distance between eyes, eyebrows, nose and mouth on Japanese Female Facial Expression (JAFFE). Pons et al. [11] via this paper explained an idea to find human expressions using a convolution neural network (CNN). The paper identifies how the thought process of humans changes and contributes to a change in expressions. Models on discrete Hopfield neural networks [12] were also proposed to identify facial expressions. This technique implied how the memory function of the Hopfield neural networks can optimally classify the predefined common expressions among a variety of facial expressions. Ma et al. [13] identified a new technique for human emotion analysis that used a 2D distinct cosine transformation over the whole facial image as a function to detect expressions. A proactive feed forward neural network having one layer is used as a facial expression classifier. Esau et al. [14] explained about a video-based emotion recognition system known as VISBER. It used video sequences to analyse the emotions. Kiran et al. [15] have used the Viola–Jones method for face detection, LBP for feature extraction and GRNN for classification.

This research proposes a deep learning model to analyse different emotions expressed by human being from images captured in real time.

3 Dataset

The dataset used for building model is the Recognition of Facial Expression (FER2013) [16] data collected from Kaggle. The dataset has photographs of faces with a greyscale (of 48 pixels). The training set has 35,887 data points. Train.csv has 2 columns, “emotion” and “pixels”. For the facial emotion, the “emotion” column contains a number (0–6), and inclusive denotes the 7 basic emotions (zero: angry, one: disgust, two: fear, three: happy, four: sad, five: surprise and six: neutral). For each image, the column “pixels” includes a string surrounded by quotes. The values of this string are separated spatially as pixel values in a row in a large order. FER has more variety in the images compared to other datasets, including facial occlusion (mostly with hand), partial faces, low-contrast images and eyeglasses.

4 Architecture of the Model—Xception (Extreme Version of Inception-V3)

The deep learning model used in this research is Xception developed by Google.

It is a modification of the separable convolution which is depthwise convolution. Due to this, it is superior to Inception-V3. The Xception [17] architecture is comparatively small in comparison with regular CNNs and is able to achieve high emotion efficiency in classification on the FER dataset. The Xception model implements pointwise convolution followed by depthwise convolution as shown in Fig. 1.

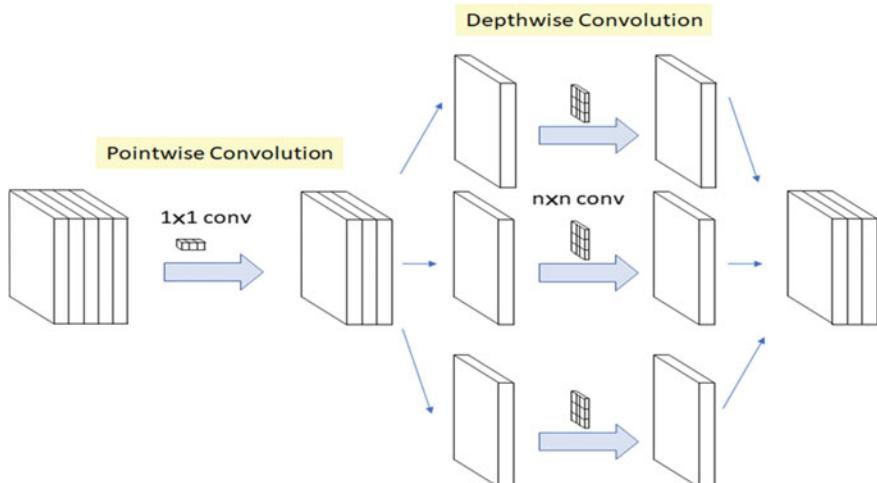


Fig. 1 Depthwise separable convolution (modified) in Xception

Our final architecture is a fully convolutional neural network. It has four separable residual depthwise convolutions. This is followed by a batch normalization operation which further improves efficiency. It then goes on to implement ReLU enabling features for each convolution. The last layer applies an average global pooling function to produce a prediction and a softmax activation function. This template has about 60,000 parameters; this correlates to a 10-min decrease compared to the standard implementation.

5 Implementation

5.1 Face Detection

Detection of faces from photographs is the first step involved. Face extraction from photographs can be achieved with various algorithms. Haar cascades [18] discovered by Viola and Jones are used to detect or remove the image. This algorithm operates by measuring Haar features rather than pixel values, which uses AdaBoost as the boosting algorithm. Every Haar feature has some kind of resemblance for recognizing a part of the face. Instead of measuring 2500 functions for each frame, we use the definition of cascades. We sample 2500 features into x different cascades. Now, we can detect linearly whether or not there is a face in multiple cascades. This moves the picture to the next cascade if the cascade finds a face in an image. If no face is found, we turn to the next browser. That reduces the time limit.

We used a conventional approach to image pre-processing by scaling them down from -1 to 1 . We scaled the images by dividing them by 255 to $[0, 1]$. Subtraction by 0.5 and multiplication by 2 are performed additionally. This is done to modify the range to $[-1, 1]$. $[-1, 1]$ has a greater variety when it comes to computer vision problems for neural network.

5.2 Training the Model

We trained our model using the Keras deep learning library. During preparation, the categorical loss of cross-entropy is reduced at a learning rate of 0.0001 , with stochastic gradient descent. The model has been equipped with 100 epochs, with 64 batch size. The model's accuracy on training and validation sets is improving as losses diminish and as training moves further as shown in Fig. 2. It can also be seen that in training and validation, their values are quite nearby.

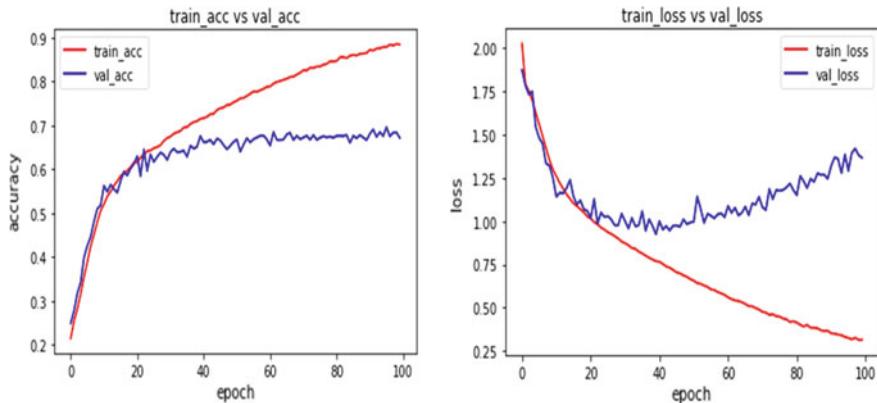


Fig. 2 Accuracy versus epoch (left) and loss versus epoch (right)

5.3 Prediction in Real Time

The live video feed is first recorded using CV2 on Python to predict in real time and fed into a face detection network, which can achieve superior performance in real time. Afterwards, the detected face is fed into our trained network, where the model generates the prediction.

6 Results and Discussion

The proposed CNN-based framework for facial emotion recognition was first trained on the training data at a learning rate of 0.0001 for 100 epochs, with a batch size of 64. Thereafter, this model was tested on a set of test data.

We have used a confusion matrix to describe the performance of our model on test data, and it is shown in Fig. 3. The average accuracy achieved by our model and proposed framework on the FER2013 dataset is 68.57%.

The proposed method is compared with the state-of-the-art techniques and is shown in Table 1.

The probabilities of each of seven emotions (major probabilities are highlighted) generated by the proposed model in real time from the images captured using webcam are shown in Fig. 4.

Fig. 3 Confusion matrix for seven emotional expressions

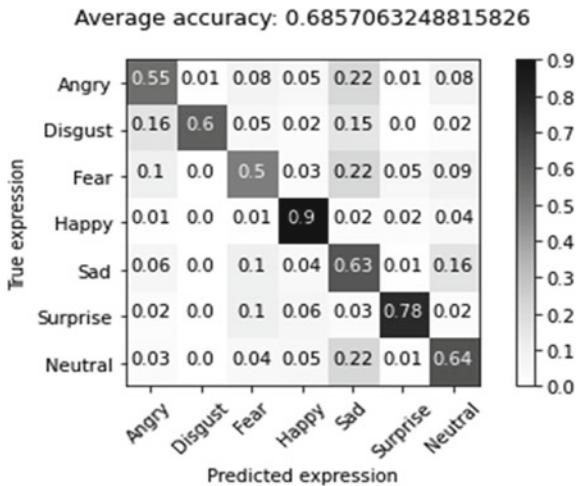


Table 1 Performance comparison

Method	Dataset	Accuracy (%)
Bag of words [19]	FER 2013	67.4
SVM [20]	FER 2013	66.31
GoogLeNet [16]	FER 2013	65.2
Proposed model	FER 2013	68.57

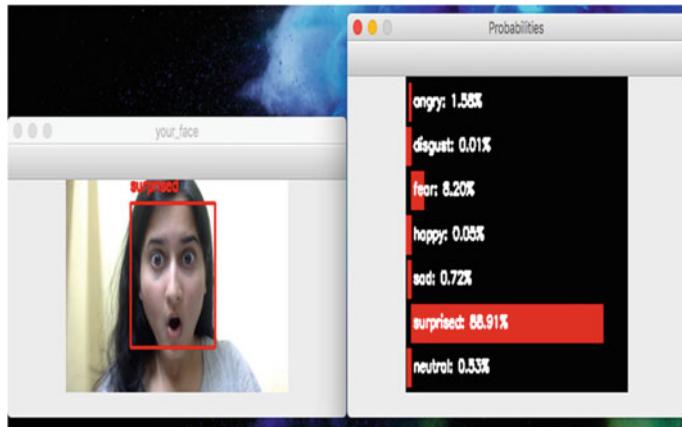


Fig. 4 Sample output

7 Conclusion and Future Scope

We have proposed and implemented a real-time facial emotion recognition model using a convolutional neural network (CNN) model—Xception. Our model achieved an average recognition accuracy of 68.57% for seven emotions. Moreover, it is found that the proposed model performs well in recognizing emotions on faces that it has seen before. The proposed model is a real-time model which detects the facial expressions from the images captured using webcam and generates the predicted emotion along with the percentage of each of the seven facial emotions.

References

1. Terusaki K, Stiglianii V (2014) Emotion detection using deep belief networks
2. Alizadeh S, Fazel A (2017) Convolutional neural networks for facial expression recognition
3. Minaee S, Abdolrashidi A (2019) Deep-emotion: facial expression recognition using attentional convolutional network. arXiv preprint [arXiv:1902.01019](https://arxiv.org/abs/1902.01019)
4. Gan Y (2018) Facial expression recognition using convolutional neural network. In: Proceedings of the 2nd international conference on vision, image and signal processing (ICVISP 2018). Association for Computing Machinery, New York, NY, USA, Article 29, pp 1–5
5. Neagoie VE, Barar A, Sebe NICU, Robitu PAUL (2013) A deep learning approach for subject independent emotion recognition from facial expressions. Recent Adv Image Audio Signal Process 93–98
6. Barros P, Jirak D, Weber C, Wermter S (2015) Multimodal emotional state recognition using sequence-dependent deep hierarchical features. Neural Netw 72:140–151
7. Azcarate A, Hageloh F, Van de Sande K, Valenti R (2005) Automatic facial emotion recognition. Universiteit van Amsterdam 1–6
8. Ko BC (2018) A brief review of facial emotion recognition based on visual information. Sensors 18(2):401
9. Zhan C, Li W, Ogunbona P, Safaei F (2008) A real-time facial expression recognition system for online games. Int J Comput Games Technol 2008
10. Zhang YD, Yang ZJ, Lu HM, Zhou XX, Phillips P, Liu QM, Wang SH (2016) Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation. IEEE Access 4:8375–8385
11. Pons G, Masip D (2017) Supervised committee of convolutional neural networks in automated facial expression analysis. IEEE Trans Affect Comput 9(3):343–350
12. Yoneyama M, Otake A, Iwano Y, Shirai K (1997, October) Facial expressions recognition using discrete hopfield neural networks. In: Proceedings of international conference on image processing, vol 1. IEEE, pp 117–120
13. Ma L, Khorasani K (2004) Facial expression recognition using constructive feedforward neural networks. IEEE Trans Syst Man Cybern Part B (Cybern) 34(3):1588–1595
14. Esau N, Wetzel E, Kleinjohann L, Kleinjohann B (2007, July) Real-time facial expression recognition using a fuzzy emotion model. In: 2007 IEEE international fuzzy systems conference. IEEE, pp 1–6
15. Talele K, Shirsat A, Uplenchwar T, Tuckley K (2016, December) Facial expression recognition using general regression neural network. In: 2016 IEEE Bombay section symposium (IBSS). IEEE, pp 1–6
16. Giannopoulos P, Perikos I, Hatzilygeroudis I (2018) Deep learning approaches for facial emotion recognition: a case study on FER-2013. Advances in hybridization of intelligent methods. Springer, Cham, pp 1–16

17. Saravanan A, Perichetla G, Gayathri DK (2019) Facial emotion recognition using convolutional neural networks. arXiv preprint [arXiv:1910.05602](https://arxiv.org/abs/1910.05602)
18. Rajesh KM, Naveenkumar M (2016) A robust method for face recognition and face emotion detection using support vector machines. In: IEEE, 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)
19. Ionescu RT, Popescu M, Grozea C (2013) Local learning to improve bag of visual words model for facial expression recognition. In: Workshop on challenges in representation learning, ICML
20. Georgescu MI, Ionescu RT, Popescu M (2019) Local learning with deep and handcrafted features for facial expression recognition. IEEE Access 7:64827–64836

Chapter 24

Clustering of Single-Cell Transcriptome Data Based on Evolutionary Algorithm in Assimilation with Fuzzy C-Means



Amika Achom and Ranjita Das

1 Introduction

The breakthrough in the technology of RNA sequencing (RNA-seq), microarrays, transcriptomics, single-cell RNA sequencing (scRNA-seq) technology, etc. has generated an enormous amount of transcriptomics, and epigenomics data. It has been able to inspect the expression patterns of genes in a cell population. This enhances the scope to investigate the heterogeneity nature of gene expression profiles. scRNA-seq technology provides complete information about the RNA expressed by each cell in a population. To identify such patterns of gene expression, genes can be organized into similar and dissimilar groups. scRNA-seq data are arranged as an expression count matrix of size $g \times c$ in 2-D, where every row represents genes (g) and column (c) represents the cell.

Clustering technique can group cells according to their transcriptomic profile, and genes can be grouped based on the similarities in their expression levels or pattern. The information about the transcriptomics factor in a cell can reflect the activity of a subset of genes in a population. In the literature, several authors have used the traditional clustering approaches such as hierarchical clustering [9] and K-means [11] for clustering scRNA-seq data. But the problem is that it is inappropriate to use the above approaches especially when the dimension of the data is very large and clusters are of changeable structures and density.

In recent years, researchers have explored graph-based partitioning method [5, 9], model-based clustering method [10], neighbor distance-based method [4], trajectory inference method [3], etc. In Ref. [9], clustering approach employs smart local moving (SLM) algorithm to detect clusters over large networks. SLM algorithm itself employs a modularity optimizer, the Louvain algorithm to construct a community network. But this algorithm tends to work by setting different resolution parameters

A. Achom · R. Das (✉)

National Institute of Technology, Mizoram, Aizawl, India

e-mail: rdas@nitmz.ac.in

which are a limitation to the above method. One limitation of the construction of trajectory topology is that the chosen trajectory method depends on the dimension of the data. So a developed trajectory method does not apply to a wide range of datasets.

To the best knowledge, no evolutionary nature-inspired algorithm has been applied to group transcriptomics profile of each gene. In this work, newly introduced weighted distance is implemented and uses as a similarity measure. In the proposed clustering framework, appropriate weights are assigned to the best solution in the optimization process so that it could explore and exploit the search space in a refined manner. FCM algorithm is assimilated with GWO since scRNA-seq data are generally affected by noise and outliers. Application of the FCM algorithm in the current work is seen to model the noise/outliers as separate clusters. The efficacy of the developed clustering algorithm is also demonstrated to identify the cell types and marker gene/transcriptomes in each cluster in the spleen tissue of mus musculus.

2 Weighted Distance as a Similarity Measure

This paper utilizes weighted distance measure as the underlying proximity measure. The weighted distance measure between any two gene points, (g_x, g_y) , is defined as:

$$D_{\text{wtDist}}(g_x, g_y) := \sqrt{\sum_{j=1}^D \{(g_x^j - g_y^j) * \text{weight}_j\}^2} \quad (1)$$

Here, the factor weight_j (in Eq. 1) is used to assign weight to the j^{th} expression value. Let us assume there are D features in a dataset. This D represents the length of a component vector for a set of weight. Initially, each of the component vector is initialized with a random number [0,1]. This is explained as: $\text{weight}_j := \text{random()}/\text{RANDOM-MAX}$. random() function generates a random number from [0, RANDOM-MAX]. After initializing each component of the feature vector with some weight, now the weighted feature vector is normalized as: $\text{weight}_j = \frac{\text{weight}_j}{\sum_{j=1}^d \text{weight}_j}$. This concept of assigning important weight to significant features introduced in ref. [8] is implemented in the paper.

3 Gray Wolf Optimizer Algorithm

Gray wolf optimizer (GWO) [6] is a nature-based evolutionary algorithm. The algorithm models the hunting and leadership relationship of gray wolves. The hunting mechanism of gray wolves is composed of three strategy (i.e., tracking, encircling, and attacking the prey) and all operate cooperatively. The encircling mechanism of GWO algorithm is modeled as:

$$\vec{D} = |\vec{C} * X_p(t) - X(t); \vec{X}_{t+1} = \vec{X}_p(t) - \vec{A} * \vec{D}| \quad (2)$$

Here, $X_p(t)$ and $X(t)$ represent the location vector of prey and gray wolves. \vec{A} and \vec{C} are updated as: $\vec{A} = 2\vec{a} * \vec{R1} - \vec{a}$; $\vec{C} = 2 * \vec{R2}$ \vec{a} is a linearly decremented from 2 to 0 over the optimization process and a is set as: $a = 2 - 2 * (\frac{t}{MaxIteration})$. $\vec{R1}$ and $\vec{R2}$ denote the random component vector from the range of [0,1]. The mathematical equation for the position updation is expressed as:

$$\vec{D}_\alpha = |\vec{C}_1 * X_\alpha(t) - X(t)|; \vec{D}_\beta = |\vec{C}_2 * X_\beta(t) - X(t)|; \vec{D}_\delta = |\vec{C}_3 * X_\delta(t) - X(t)| \quad (3)$$

$$\vec{X}_1 = X_\alpha(t) - \vec{A}_1 * (\vec{D}_\alpha); \vec{X}_2 = X_\beta(t) - \vec{A}_2 * (\vec{D}_\beta); \vec{X}_3 = X_\delta(t) - \vec{A}_3 * (\vec{D}_\delta) \quad (4)$$

$$\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (5)$$

The component vector \vec{a} controls the exploration and exploitation phase of the algorithm. When $|\vec{A}| > 1$, solution diverge and explore the search space to find a better prey and $|\vec{A}| < 1$ solution converges and update the position in the best direction of α , β , and δ solutions.

4 Proposed Method

This section outlines the detailed workflow for the proposed clustering algorithm and is described as follows:

4.1 Population Initialization

The population pool consists of a set of chromosome vectors. These chromosome vectors are initialized with a unique real number randomly from the dataset and represent the coordinate of the cluster centers. All vectors are of the same length.

4.2 Fuzzy C-Means Clustering Algorithm

Fuzzy C-means clustering (FCM) [2] algorithm is based on the minimization of the intra cluster variance and formulated as an objective function (J -measure) as:

$$J(u, K) = \sum_{i=1}^N \sum_{j=1}^K u_{ij}^m D_{\text{wtDist}}^2(g_i, K_j), 1 \leq m < \infty \quad (6)$$

$D_{\text{wtDist}}(g_i, K_j)$ represents the weighted Euclidean distance of i th data point (d_i) to j th cluster center. u_{ij} indicates the membership value of i th data point (d_i) in cluster j . K_j represents the center of cluster j . m is the fuzzifier constant and $m = 2$. FCM algorithm iteratively minimize $J - \text{measure}$ and at the same time update the membership value (u_{ij}) and cluster centers (K_j). The equation for updating the membership value and cluster center is described as:

$$u_{ij} = \frac{1}{\sum_{l=1}^K \left(\frac{D_{\text{wtDist}}(g_i, K_l)}{D_{\text{wtDist}}(g_i, K_j)} \right)^{\frac{2}{m-1}}}; K_j = \frac{\sum_{i=1}^N (u_{ij})^m * g_i}{\sum_{i=1}^N (u_{ij})^m} \quad (7)$$

The step for executing the FCM algorithm is briefly outlined as follows:

1. Firstly, cluster center encoded in the chromosome vector is extracted out.
2. Weighted distance of every gene point from the extracted cluster center is calculated.
3. Now a gene, (g_i) is allocated to the cluster having the minimum weighted distance from all the cluster center in step (1).
4. Then membership value and cluster centers are recomputed or updated using Eq. (7).

4.3 Generation of Candidate Solution

After executing the FCM algorithm, it generates a set of candidate solutions for the optimization algorithm. To establish a hierarchical structure, the fitness value is calculated using Eq. (7). The solution with minimal fitness value is considered to be the best solution and so on.

4.4 Updating Position and Weighting of Search Agents

The position updation of the GWO algorithm is improved in the paper by assigning dynamic weight to α , β , and δ solution instead of the average position of these three solutions. These dynamic weights are assigned from the weight matrix and are updated at every step of the GWO algorithm. Rodriguez et al. [7] instigate a hierarchical operator to implement a dynamic pyramidal structure for adjusting weights to α , β , and δ solution according to crisp fuzzy logic. The above concept can be expressed in the following equation as:

$$\vec{X}(t+1) = \frac{\vec{X}_1 * W_\alpha + \vec{X}_2 * W_\beta + \vec{X}_3 * W_\delta}{(W_\alpha + W_\beta + W_\delta)} \quad (8)$$

$$W_\alpha = \frac{(f_\alpha + f_\beta + f_\delta)}{f_\alpha}; W_\beta = \frac{(f_\alpha + f_\beta + f_\delta)}{f_\beta}; W_\delta = \frac{(f_\alpha + f_\beta + f_\delta)}{f_\delta} \quad (9)$$

f_α, f_β and f_δ are the fitness value obtained from $J - measure$.

4.5 Terminating Criteria

The proposed algorithm (FCM and GWO) is executed for a maximum number of generations and finally returns an optimal cluster center for clustering. The pseudocode for the proposed clustering approach is given below in algorithm 1.

Algorithm 1: Proposed clustering algorithm (Fuzzy C-means and GWO)

1. **Input:** Pre-processed single cell transcriptomics data
 2. **Output:** Optimum cluster center or centroid for clustering
 3. **begin (procedure)**
 4. Population $P \in [1 : NP]$ contain NP number of solution vector
 5. Initialize control parameters: a , A , and C
 6. Evaluate fitness value (Equation: 6)
 - $X_\alpha :=$ Best solution vector encoded as α
 - $X_\beta :=$ Second best solution vector represented as β
 - $X_\delta :=$ Third best solution vector represented as δ
 7. **while** ($t < MaxIteration$)
 8. **do**
 9. **for** (each vector in P)
 10. **do**
 11. Update current position (Equation 8)
 12. **end for loop**
 13. Update control parameters: a , A , and C
 14. Compute and update fitness values of all vector in P
 15. Calculate and assign weight to α, β, δ (Equation 9)
 16. Update the positions of $X_\alpha, X_\beta, X_\delta$ (Equation 3)
 17. $t = t+1$
 18. **end of while loop**
 19. return X_α (optimal cluster center)
 20. **end of procedure**
-

5 Experiment and Result

5.1 Data Preprocessing, Normalization, and Feature Selection

To preprocess the expression, matrix (N row, i.e., gene \times column) is considered. This matrix records the total number of molecules (UMI) for each gene detected in every cell. The following steps are executed:

1. Feature selection: Cells having a unique feature count of over 2500 or more than 200 are selected.
2. Data normalization: The expression value of each cell is normalized by the total expression value and multiply by a factor of 10,000.
3. Gene selection: To identify subsets of genes that are highly expressive, ratio of mean–variance (i.e., normalized dispersion) is calculated.
4. Data scaling: Expression value of each gene is scaled so that mean and variance across each cell are 0 and 1, respectively.
5. Dimension reduction: To map high-dimensional data to a 2-D form, principal component analysis (PCA) is applied to the scaled data.

After preprocessing, clustering algorithm is executed on the principal components. Bladder tissue, kidney tissue, limb muscle, and spleen tissue (GSE109774) of mus musculus are downloaded from Tabula Muris¹ and GSM2230757 pancreatic islet human1 sample [1], GSM2230758 pancreatic islet human2 sample [1], GSM2230759 pancreatic islet human3 sample [1], GSM2230760 pancreatic islet human4 sample [1], and GSM2230761 pancreatic islet mouse1 sample [1] are downloaded from National Center for Biotechnology Information (NCBI) (Fig 1).²

5.2 Analysis of Clustering Result

A comparative evaluation is conducted considering SC3 [4], CIDR [5], Seurat [9], t-SNE and K -means [11], Ensemble technique [10], DE-based FCM, and proposed clustering algorithm (FCM and GWO) on different sample of human and mouse pancreatic islets. A good score of adjusted rand index (ARI) on all these samples in Table 1. Table 1 indicates that FCM and GWO can also give good clustering results. Figure 2 display the 2-D representation of the clustering results. For human pancreatic islets sample1, sample2, sample3, and sample4, 14 different clusters are obtained. For mouse pancreatic islets sample1, 13 different clusters are obtained.

¹<https://tabula-muris.ds.czbiohub.org/>.

²<https://www.ncbi.nlm.nih.gov/>.

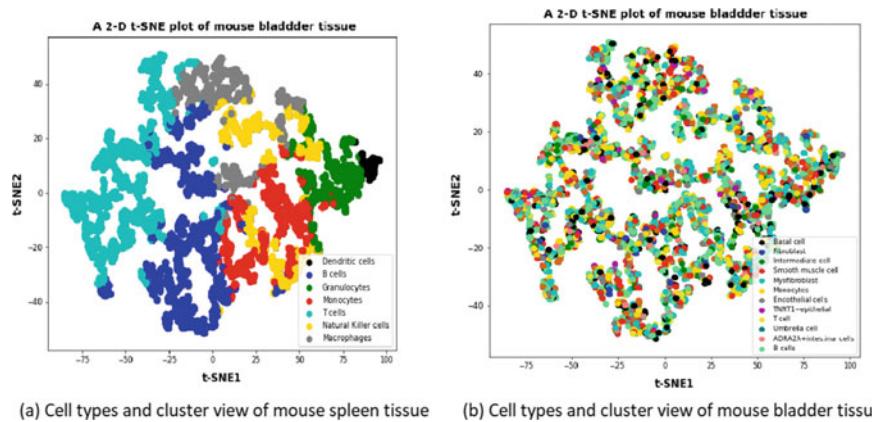


Fig. 1 2-D graphical view of obtained clusters for mouse spleen and bladder scRNA transcriptomes data

Table 1 Comparative performance evaluation of proposed (FCM-based GWO) with some clustering algorithm wrt adjusted rand index (ARI) on human and mouse sample

Clustering	Human1	Human2	Human3	Human4	Mouse1
SC3	0.35	0.50	0.24	0.499	0.45
CIDR	0.50	0.25	0.432	0.600	0.377
Seurat	0.37	0.150	0.389	0.575	0.52
t-SNE and K-means	0.52	0.75	0.763	0.75	0.62
Ensemble technique	0.625	0.85	0.74	0.772	0.58
DE based FCM	0.5248	0.4900	0.5990	0.6896	0.6275
Proposed (FCM and GWO)	0.6500	0.8003	0.7325	0.7550	0.6220

5.3 Identification of Cell Types and Transcriptomes in Spleen Tissue of *Mus Musculus*

To analysis the clustering result, FCM and GWO clustering algorithm is extended to identify the cell types and transcriptomes or genes present in the gene cluster. For this relative frequency, % of total gene data point in each cluster is calculated. Marker genes or transcriptome in each cluster correspond to that gene that has a high expression level. Table 2 depicts the identified cell types and marker genes. The obtained cluster is shown in Fig. 2.

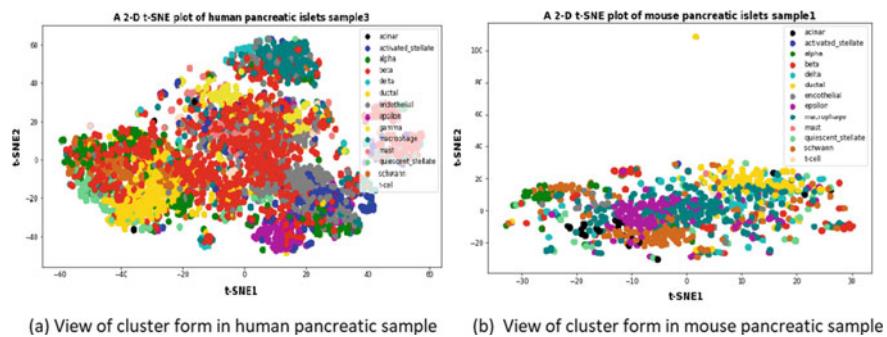


Fig. 2 2-D view of obtained clusters for pancreatic islet of human and mouse [1]

Table 2 Identification of types of cell in spleen tissue of mus musculus

Cell types	Frequency %	Markers Gene or transcriptomes
T cells	23.87596899	CD4+, Cdc40, CD8+, Trmt11, Emc10
B cells	27.68992248	CDCA3, Cdk12, CD19, C030006K11, CD39, CD73
Monocytes	13.58139535	Ly6e, Lmf2, COA3, COX8A
Granulocytes	9.798449612	CD11b, Ly6e, Ppp2ca, Ppp2r1a, Rab5if
Dendritic cells	11.81395349	Mdc1, Cd11c(Gid8), Clec4a2, CD86
Natural Killer Cells	11.62790698	Cd244, KlrB1c, Klf13, Kif5b
Macrophages	1.612403101	Fth1, Cct7, Csf1

6 Conclusion and Future Work

The current paper attempts to present a new form of clustering method for scRNA transcriptome data. The ability of the developed clustering technique to identify cell types and transcriptomics factor/marker gene in a cluster demonstrated its robustness nature to adapt to the bioinformatics field. Also, the work demonstrated its capability to detect subpopulation or rare populations in a cluster or population. Good J-measure and ARI scores motivated to apply different evolutionary search operators. As a part of future work, an automated multiobjective framework could be developed to enhance its functionality.

Acknowledgements Dr. Ranjita Das acknowledges Sunrise Project with Ref: NECBH/2019-20/178 under North East Centre for Biological Sciences and Healthcare Engineering (NECBH) Twinning Outreach Programme hosted by Indian Institute of Technology Guwahati (IITG), Guwahati, Assam funded by Department of Biotechnology (DBT), Ministry of Science and Technology, Govt. of India with number BT/COE/34/SP28408/2018 for providing necessary financial support.

References

1. Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, Ryu JH, Wagner BK, Shen-Orr SS, Klein AM et al (2016) A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Syst* 3(4):346–360
2. Hathaway RJ, Bezdek JC (2001) Fuzzy c-means clustering of incomplete data. *IEEE Trans Syst Man Cybern Part B Cybern* 31(5):735–744
3. Ji Z, Ji H (2016) Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic Acids Res* 44(13):e117–e117
4. Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, Natarajan KN, Reik W, Barahona M, Green AR et al (2017) Sc3: consensus clustering of single-cell rna-seq data. *Nat Methods* 14(5):483
5. Lin P, Troup M, Ho JW (2017) Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biol* 18(1):59
6. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
7. Rodríguez L, Castillo O, Soria J, Melin P, Valdez F, Gonzalez CI, Martinez GE, Soto J (2017) A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Comput* 57:315–328
8. Saha S, Acharya S, Kavya K, Miriyala S (2017) Simultaneous clustering and feature weighting using multiobjective optimization for identifying functionally similar mirnas. *IEEE J Biomed Health Inf* 22(5):1684–1690
9. Satija R, Farrell JA, Gennert D, Schier AF, Regev A (2015) Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 33(5):495
10. Yang Y, Huh R, Culpepper HW, Lin Y, Love MI, Li Y (2018) Safe-clustering: Single-cell aggregated (from ensemble) clustering for single-cell rna-seq data. *Bioinformatics* 35(8):1269–1277
11. Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu J et al (2017) Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 8:14049

Chapter 25

Waste Segregation to Ease Recyclability



Rahul Kumar Verma and Suneeta Agarwal

1 Introduction

Billion tonnes of waste is generated in the world daily, and most of the countries face challenges to manage it. In most countries, the waste is segregated into dry and wet waste at the house itself, but municipal authorities still face difficulties to recycle the waste; as a result, huge volume of waste is left untreated. In some countries, a lot of manpower is used to segregate the dry waste which is unhygienic and cumbersome. Our motivation is to efficiently segregate the dry waste to ease the recycling process. This will not only help to reduce environmental pollution but also be economically beneficial. Here, we manage the solid waste by segregating it to recycled or non-recycled waste to properly dispose the waste completely. Paper, metal and plastic can be recycled, and other waste can be decomposed as a landfill. The only way to simplify the solid waste management is to automate the process of classification.

Earlier number of techniques has been devised to automate the process of waste segregation. Some of them have used image processing techniques to segregate the waste. The RGB pixel, texture and their intensity are used for the classification. In today's world of artificial intelligence and machine learning, supervised machine learning techniques are being used for image classification. In this technique, a labeled dataset is created and the devised model predicts the class as output based on the input data. Recently, deep learning architectures like convolution neural network have become popular in image classification. It has been proved to be more efficient as compared to conventional machine learning techniques in face recognition, speech recognition, image analysis in health care, object detection, etc. The CNN model learns the features from the input data and passes them to the subsequent layers to abstract the feature more precisely. Finally, the architecture predicts the output based

R. K. Verma · S. Agarwal (✉)

Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India

e-mail: suneeta@mnnit.ac.in

on the feature gathered during learning. However, the CNN model requires a large dataset to work efficiently. To overcome this disadvantage, the researcher moved toward capsule neural network which gives results even with small training dataset. Since we are using the dataset of only 4956 images, we will employ capsule neural network for waste segregation.

2 Related Works

Previously, many efforts have been made to manage the waste efficiently. In early days, many researchers have used image sensors to detect the category of waste. Thung and Yang [1] used support vector machine and CNN-based AlexNet neural network [2] to classify waste images into six different categories. They focused on comparing the result obtained by different machine learning techniques. Support vector machine yielded an accuracy of 63% compared to 25% of CNN-based model with the 70/30 training/test data split.

Sreelakshmi et al. [3] implemented capsule neural network to segregate waste into plastic and non-plastic categories. They compared two different datasets. One dataset included images collected from different Web sites, while other consists of real-time images. They have achieved an accuracy of around 96% for both the datasets when trained for 1000 epochs.

3 Capsule Neural Network

CNN proved to be the pillar of image classification, but its performance degrades when the images are rotated and tilted or have a different orientation. CNN ignores orientation and spatial relationship between the components of the image. It requires a large amount of data to train a CNN model. Due to these limitations, capsule neural network came into existence. Capsule neural network is also a CNN-based model which consists of capsules. Capsules are vector which specifies the important features of the image along with its likelihood. These features include pixel, texture, hue and color intensity, and with capsules it also stores features like pose, size, orientation, deformation, etc. Capsule maintains equivariance by changing the output according to the changes made in the input data keeping the information intact.

Capsule neural network is a layered neural network. Several convolution layers are used to implement the lower layer capsule which fetches the small features of input image using rectified linear unit (ReLU) activation function. More complex features like orientation, pose, tilt and shift are gathered by upper layer capsules known as routing capsules. Using squashing function, the output of lower capsule is converted to vector which has values 0 or 1 to represent probability while preserving direction. At the higher layer capsule, dynamic routing [4] is implemented to gather complex features of the image. Routing consists of routing weights for each interconnection

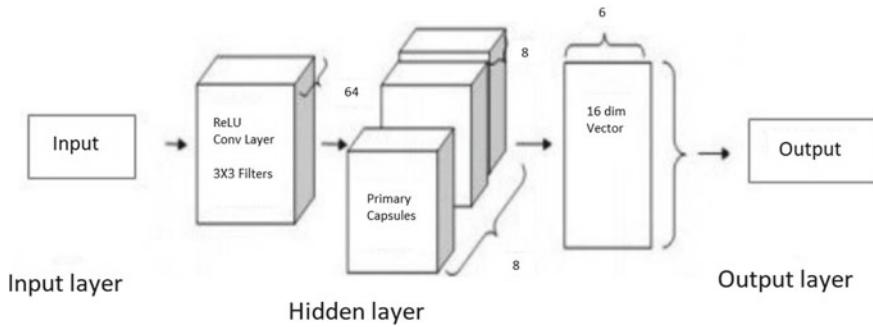


Fig. 1 Capsule neural network

between capsules. Routing weight increases with high possibility of agreement or decreases otherwise (Fig. 1).

4 Dataset Description

The existing dataset of Thung and Yang [1] of 2500 images is used, and later it is improved by adding more images to the dataset. The dataset of 4956 images is labeled as cardboard, glass, metal, paper, plastic and trash with 403, 966, 815, 1532, 907 and 333 images, respectively. Figure 2 displays example of the dataset used. The



Fig. 2 Example of dataset

Table 1 Details of dataset

Classes	Training data	Test data
Cardboard	323	80
Glass	773	193
Metal	652	163
Paper	1226	306
Plastic	726	181
Trash	267	66

dataset is divided into 80:20 training and test data to implement it on the capsule neural network (Table 1).

5 Proposed Architecture

Figure 1 gives an overview of the capsule neural network architecture. It consists of three layers which are input layer, hidden layer and output layer. In the input layer, the images are resized to 64×64 to ease the computation of the network. Convolution layer is implemented with 64 filters of 3×3 filter size. It uses ReLU activation function. Primary capsules are formed after the convolution layer. In the hidden layer, input vectors are multiplied with weight matrix to gather important spatial relationship between low-level features and high-level features within the image. Squash nonlinearity function is used to squash the vector to 0 or 1 along with its direction. Routing by agreement takes place in the high-layer capsules depending upon the weighted input vectors. A 16D vector outputs after the routing which contain all the instantiation parameters required to reconstruct the image. In the final layer, lambda loss function is used to determine how similar the reconstructed image compared to the actual image the network is being trained from. The length of the vector determines the presence of probability of a particular object and classifies it as one of the six classes mentioned in the dataset.

6 Performance Parameters

The performance of a multi-class model is based on confusion matrix. The parameters used in confusion matrix are true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Table 2 gives an overview of the confusion matrix with predicted values on horizontal axis and actual values on vertical axis.

The values of confusion matrix are defined as follows:

- True Positive (TP): Image predicted as positive and it is true.
- True Negative (TN): Image predicted as negative and it is true.

Table 2 Confusion matrix

	Predicted-yes	Predicted-no
Actual-yes	TP	FN
Actual-no	FP	TN

- False Positive (FP): Image predicted as positive and it is false. It is a type1 error.
- False Negative (FN): Image predicted as negative and it is false. It is a type2 error.

7 Results and Observations

Capsule neural network achieved better result compared to that of the Thung and Yang CNN-based model [1]. Thung and Yang faced difficulty in training the network as the dataset was very small. They even discarded the trash category of the dataset due to very less images in that category. They are barely able to achieve the test accuracy of 22%. Confusion matrix of Thung and Yang CNN model is shown in Fig. 3. Table 3 reports the detailed result of Thung and Yang.

On the other hand, capsule neural network manages to train the network and achieves the validation accuracy of 79.27% when trained for 100 epochs with batch size 32. Confusion matrix is shown in Fig. 4. Table 4 gives the detailed report of capsule neural network. Figures 5 and 6 show the training-validation loss and accuracy curve, respectively.

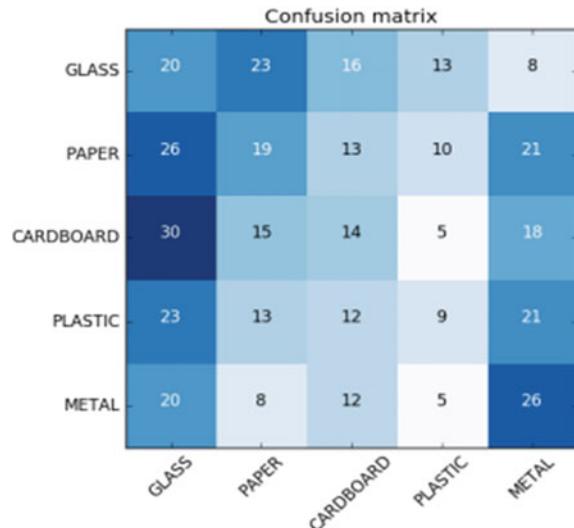
Fig. 3 Confusion matrix for Thung and Yang CNN model

Table 3 Classification report for Thung and Yang CNN model

Material	Precision	Recall
Cardboard	0.25	0.17
Glass	0.21	0.29
Metal	0.17	0.21
Paper	0.12	0.21
Plastic	0.37	0.28

	Cardboard	Glass	Metal	Paper	Plastic	Trash
	0	1	2	3	4	5
Cardboard	31	18	11	2	13	5
Glass	14	85	30	25	20	19
Metal	7	41	49	27	23	16
Paper	24	28	45	152	48	17
Plastic	14	38	37	20	57	15
Trash	7	15	7	6	7	24

Fig. 4 Confusion matrix for capsule neural network

Table 4 Classification report for capsule neural network

Material	Precision	Recall	F1-score
Cardboard	0.36	0.38	0.37
Glass	0.39	0.44	0.41
Metal	0.27	0.30	0.28
Paper	0.65	0.49	0.55
Plastic	0.34	0.31	0.35
Trash	0.20	0.36	0.25

Fig. 5 Training and validation loss for capsule neural network

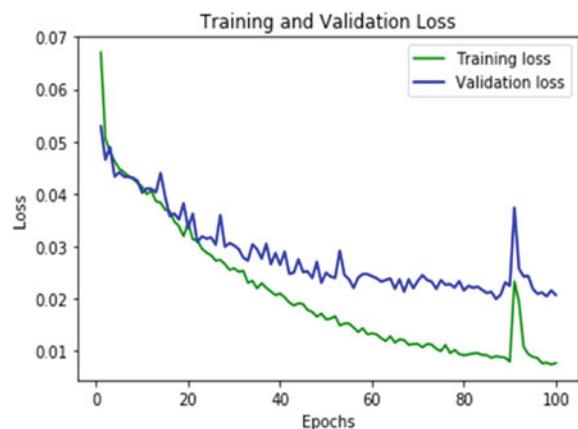
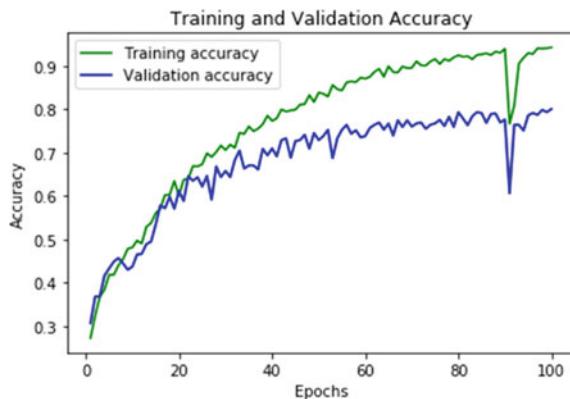


Fig. 6 Training and validation accuracy for capsule neural network



8 Conclusion and Future Work

Capsule neural network is capable of classifying images efficiently. It is shown to be more accurate compared to other CNN-based model. Capsule neural network consists of capsule which stores more relevant and complex information of the images. Capsule neural network is implemented on six different categories of solid waste. Capsule neural network achieved an accuracy of 79.27% on the given dataset as compared to previous CNN-based models. We have found that capsule neural network performs better and is able to obtain better accuracy even with the small dataset of 4956 images. In the future, we would like to extend this model to implement it on complex images of solid waste. The efficiency of the model can be further improved by balancing the dataset. We can improve our model by detecting and classifying multiple objects. This would help in classifying a stream of recycling materials instead of single object. Finally, we would add more classes to the dataset to improve the classification categories of recycling materials and hence we can retain the same model.

Reference

1. Yang M, Thung G (2016) Classification of trash for recyclability status. CS229 Project Report 2016 (2016)
2. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
3. Sreelakshmi K, Akarsh S, Vinayakumar R, Soman KP (2019) Capsule neural networks and visualization for segregation of plastic and non-plastic wastes. In: 2019 5th international conference on advanced computing & communication systems (ICACCS), pp 631–636. IEEE, 2019.
4. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Advances in neural information processing systems, pp 3856–3866

5. Patrick MK, Adekoya AF, Mighty AA, Edward BY (2019) Capsule networks—a survey. *J King Saud Univ-Comput Inf Sci*
6. Adedeji O, Wang Z (2019) Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manuf* 35:607–612
7. Chu Y et al (2018) Multilayer hybrid deep-learning method for waste classification and recycling. *Comput Intell Neurosci* 2018 (2018)
8. Aral RA et al (2018) Classification of trashnet dataset based on deep learning models. In: 2018 IEEE international conference on Big Data (Big Data). IEEE
9. Bircanoglu C et al (2018) Recyclenet: intelligent waste sorting using deep neural networks. In: 2018 innovations in intelligent systems and applications (INISTA). IEEE
10. Xi E, Bing S, Jin Y (2017) Capsule network performance on complex data. arXiv preprint [arXiv:1712.03480](https://arxiv.org/abs/1712.03480)
11. Zhihong C et al (2017) A vision-based robotic grasping system using deep learning for garbage sorting. In: 2017 36th Chinese control conference (CCC). IEEE (2017)
12. Awe O, Mengistu R, Sreedhar V (2017) Smart trash net: waste localization and classification. arXiv preprint (2017)

Chapter 26

Comparative Analysis of Intelligent Systems using Support Vector Machine for the Detection of Diabetic Retinopathy



G. Sri Venkateswara Reddy, Dolly Das, Saroj Kumar Biswas,
B. Sai Prashanth, B. Praful Bhargav, T. Vinay Kumar, Monali Bordoloi,
Biswajit Purkayastha, and Tohida Rehman

1 Introduction

Diabetic Retinopathy (DR) is a form of diabetic eye disease (DED) which occurs because of diabetes mellitus. DR affects patients who have had diabetes from a very long time. Diabetes occurs when human body fails to secrete enough insulin, which results in growth of blood glucose level. The retinal blood vessels may get damaged due to increase in blood glucose level, causing vision loss. The prolonged diabetes is particularly dangerous as it increases the danger of blindness, if it is left undetected and without treatment at some earlier point of time. It has affected 80% of diabetic patients who are suffering from diabetes for a period of 10 years or more [1]. DR causes blindness because of retinal damage caused due to leakage of fluid from retinal blood vessels, which results in the formation of different kinds of retinal lesions and/or DR features such as microaneurysms, hemorrhages, exudates, ruptured retinal blood vessels, fovea avascular zone, cotton wool spots, etc. DR features are obtained from the assessment and analysis of retinal fundus images. The fundus image is the rear view of the human eye that consists of the optic disc, the macula region, the fovea and the retinal blood vessels. It is captured using a digital fundus camera with or without pupil dilation. Based on the presence of these features, DR is classified into the following categories, namely 0-no DR, 1-mild Non-Proliferative DR (NPDR), 2-moderate NPDR, 3-severe NPDR and 4-proliferative DR, for the purpose of identification of early stages of DR. DR is alarmingly increasing and

G. Sri Venkateswara Reddy · D. Das (✉) · S. K. Biswas · B. Sai Prashanth · B. Praful Bhargav ·

T. Vinay Kumar · M. Bordoloi · B. Purkayastha

Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

T. Rehman

Information Technology, Jadavpur University, Kolkata, India

mostly affecting the younger generation of the population. Even though manual screening is a feasible solution, but it has certain constraints which makes it very unreliable for early exposure and diagnosis of the disease, to a great extent. All these factors have signified the importance and need of a cost-optimized intelligent system for early prediction and detection of DR. Various techniques are proposed to design such an intelligent system using Support Vector Machine (SVM) [2–6], Naive Bayes (NB) classifier [4, 7], K-Nearest Neighbour (KNN) classifier [4, 5], Gaussian Bayes classifier [2, 3], genetic algorithm [8], neural network [9] and various other ML techniques [5, 7, 8, 10, 11, 9, 12, 13, 14]. Thus, an intelligent system named Intelligent System for DR using Support Vector Machine (ISDRSVM) for initial exposure of DR is proposed. The objective of this proposed system is to distinguish DR features extracted from the retinal fundus images acquired from the Kaggle dataset and categorize the severe stages of DR using SVM classifier. The SVM classifier helps in addressing non-linear classification tasks especially using larger dataset. They are large margin classifiers with larger boundaries to perform effective classification. Image processing is another fundamental phase in medical imaging for better feature extraction and image classification [15]. The main objective of this model is to implement the model proposed by Carrera et al. [16] using Kaggle dataset. The model is trained on four different SVM kernels, namely linear kernel, polynomial kernel, sigmoid kernel and Radial Basis Function (RBF) kernel. The performance of the sigmoid kernel is compared, and it can be found that there exists a significant difference in the performances of the models. In the following sections, various models proposed earlier for detection of DR, the proposed methodology and the results obtained from it and related discussions are elaborated.

2 Literature Survey

DR is a chronic disease that causes mutilation to the eye if the person is diabetic and is suffering from it for a very long time, causing severe blindness. In order to introduce an intelligent automated system to such a problem, where manual clinical analysis and examination are a time-consuming task, various ML algorithms are proposed using SVM for classification of fundus images and identify DR features, for the early diagnosis of the disease. Some of these works are discussed in this section.

Aravind et al. [17] have proposed a method which classifies 105 fundus images using SVM for detection of DR. The images are preprocessed initially, and then morphological operations are applied for elimination of optic disc and vessels, for the extraction of microaneurysms and associated statistical features. The methodology has classified DR as normal, mild and severe with a sensitivity and a specificity of 92% and 80%, respectively. Zohra et al. [18] have proposed a methodology for the recognition of retinal blood vessels and hard exudates, through texture analysis, feature extraction and SVM classification, for the detection of DR. The proposed method has used statistical approaches such as Spatial Gray Level Dependency (SGLD) matrix, for texture analysis and morphological techniques for feature extraction. The

proposed system has achieved a sensitivity of 97.5% and a specificity of 100%. Jaya et al. [19] have anticipated a recognition system which engages a fuzzy membership function to evaluate the degree of membership of the structures which comprises or resembles with exudates and non-exudates, for the detection of DR. The model has used an imbalanced dataset of 200 retinal images to extract the features and classify them as exudates or non-exudates, based on pixels. The Fuzzy SVM (FSVM) model takes a total of 29 features to perform texture analysis. The FSVM has achieved a high performance of A_z of 0.9552, an accuracy of 93%, sensitivity of 94.1%, specificity of 90% and a misclassification rate of 0.07. Carrera et al. [16] have proposed a model for automatic classification of NPDR, a severe case of DR for the detection of DR. The model extracts blood vessels, microaneurysms and hard exudates from 400 retinal images. They have obtained a sensitivity of 95% and a predictive capacity value of 94%, using SVM. Sopharak et al. [20] have proposed a methodology for the exposure of exudates, in 39 non-dilated retinal fundus images and 231,734 samples, for the identification and primary uncovering of DR. The images are processed, features are selected, and classification is performed using Naive Bayes (NB) and SVM classifier, respectively. The proposed model has used 15 features for exudates detection. Naïve Bayes classifier has obtained a PR (average of precision and sensitivity) of 65.78% on six features. SVM classifier has a sensitivity of 92.28%, specificity of 98.52%, precision of 53.05%, PR of 72.67%, and an accuracy of 98.41%. Using the feature set of NB classifier, the NN classifier has obtained a PR of 61.54% and 61.81%, for Euclidean and Mahalanobis distance, respectively. Again, using the feature set of SVM classifier, the NN classifier has obtained a PR of 65.15% and 64.99%, for Euclidean and Mahalanobis distance, respectively. The NB classifier has achieved an overall per-pixel sensitivity of 93.38%, specificity of 98.14%, precision of 47.51%, PR of 70.45%, and an overall accuracy of 98.05% on a test set of 42,909 exudate pixels and 2,374,201 non-exudate pixels. Priya et al. [21] have proposed a comparative methodology to establish the performances of Probabilistic Neural Network (PNN), Bayesian classification and SVM, for early detection and diagnosis of DR. The model performs image processing, morphological processing and boundary detection for the extraction of features such as retinal blood vessels, hemorrhages and exudates, using 350 fundus images. The experimental results have shown that PNN has an accuracy of 89.6%, Bayes classifier has an accuracy of 94.4%, and SVM has an accuracy of 97.6%. Additionally, the model also uses 130 images from ‘DIARETDB0: Evaluation Database and Methodology for Diabetic Retinopathy’ upon which PNN has an accuracy of 87.69%, Bayes classifier has an accuracy of 90.76%, and SVM has an accuracy of 95.38%. Foeady et al. [22] have proposed a method to detect microaneurysms and hemorrhages through morphological operations followed by image classification using SVM, for the detection of DR. The fundus images are acquired from DIARETDB which are preprocessed initially, and then features are extracted using gray level co-occurrence matrix (GLCM). This study has achieved an accuracy of 82.35% on classification of normal eye and DR and 100% accuracy for NPDR and Proliferative DR (PDR).

3 Proposed Methodology

A computerized system for the recognition and classification of the stages of DR is proposed and implemented. Figure 1 depicts the proposed methodology for detection of DR by performing data acquisition, data analysis, image enhancement, extraction of features such as blood vessels, microaneurysms, exudates and hemorrhages and classification using SVM.

3.1 Dataset Acquisition and Analysis

The dataset of retinal fundus images is acquired from Kaggle DR detection competition, captured on the basis of various imaging conditions. The images are labeled as left and right field of the human retina, for every DR patient. Experts have labeled the DR fundus images on a scale of 0–4 such as 0—no DR, 1—mild DR, 2—moderate DR, 3—severe DR and 4—proliferative DR, based on the presence and absence of DR features. Figure 2 depicts the different grades of DR fundus images present in the Kaggle dataset. The Kaggle dataset consists of 35,126 fundus images. The dataset is highly skewed. Among these images, 25,810 images show no DR, i.e., grade 0; 5292 images have mild DR, i.e., grade 1; 2443 images have moderate DR, i.e., grade 2; 873 images have severe DR, i.e., grade 3, and 708 images have proliferative DR, i.e., grade 4. The dataset is highly imbalanced.

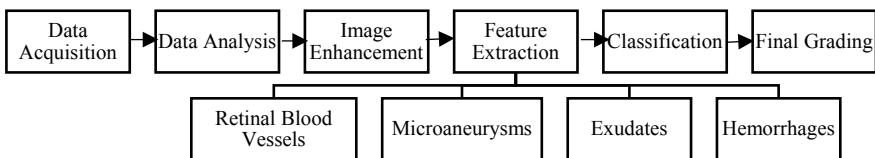
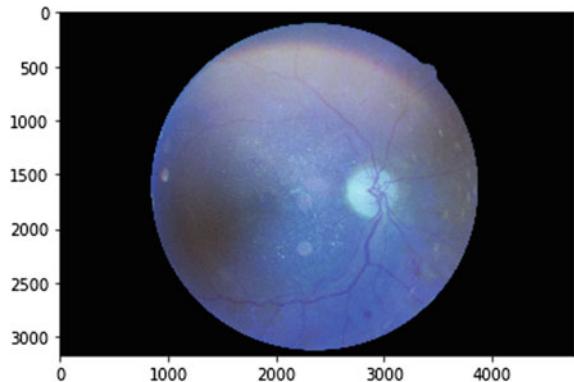


Fig. 1 Proposed methodology for detection of DR using SVM



Fig. 2 Grades of DR fundus images present in the Kaggle dataset: (i) Normal eye. (ii) Mild DR eye. (iii) Moderate DR eye. (iv) Severe DR eye. (v) Proliferative DR eye

Fig. 3 Enhanced blood vessels



3.2 Image Enhancement

Image enhancement is the procedure of fine-tuning the digital images for effective image analysis. These techniques are used to extract higher image details which are concealed and to highlight features of interest such as change in brightness, contrast, removing noise, filtering and stretching. Different approaches in image enhancement are filtering, morphological processing such as dilation and erosion, histogram equalization, noise removal using a Wiener filter, linear contrast adjustment, unsharp mask filtering and decorrelation stretch. To perform image enhancement, the Probability Mass Function (PMF) of the pixels in the image is computed. In the next step, the Cumulative Mass Function (CMF) is computed which is multiplied with the Cumulative Distribution Function (CDF) value with gray levels (minus). The new gray level values obtained are mapped onto the number of pixels which gives the final equalized image.

3.3 Feature Extraction

In the proposed model, DR is classified based on four features extracted from the fundus images. They are blood vessels, microaneurysms, exudates and hemorrhages. The features are extracted using images from the Kaggle dataset.

3.3.1 Blood Vessels

In the proposed method, the density of the blood vessels in the fundus image is determined. Initially, the color space model conversion takes place from RGB (Red Green Blue) to CMY (Cyan Magenta Yellow) representation. The magenta component is secluded. Morphological operations such as erosion, opening and dilation are

Fig. 4 Segmented blood vessels

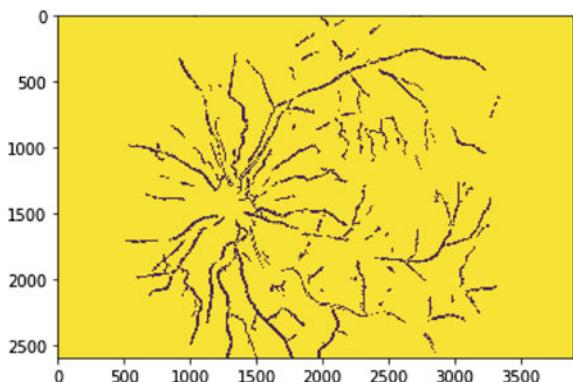
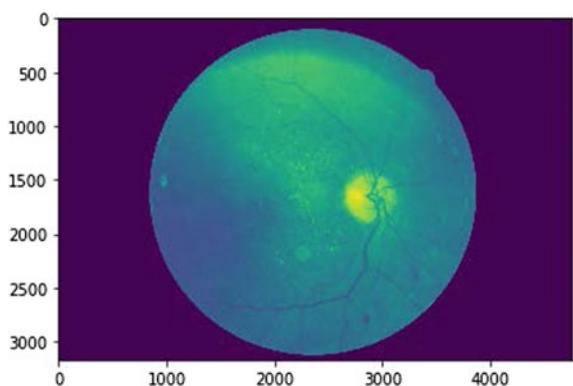


Fig. 5 Enhanced microaneurysms

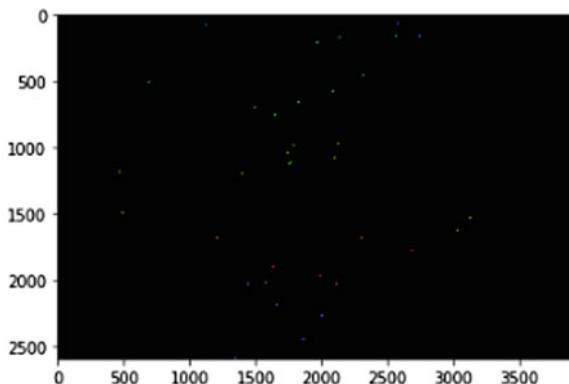


performed to eliminate the blood vessels. The transformation between the magenta component and the resultant image, obtained after morphological processing, is binarized. Contrast Limited Adaptive Histogram Equalization (CLAHE) is used to enhance the image contrast. Figure 3 depicts the enhanced blood vessels. The noise in the binarized image is reduced through dilation and erosion. Noise is reduced using mean filtering. The density of blood vessels (white pixels), in the resultant image, is computed by subtracting the blurred image from the enhanced image. The resultant images are compared with the thresholding operator, and the final set of images are formed. Figure 4 depicts the segmented blood vessels obtained from the enhanced image in Fig. 3.

3.3.2 Microaneurysms

Microaneurysms are minor swellings in the blood vessels of the retina, insignificant and circular-shaped dots near the blood vessels. They are identified as the

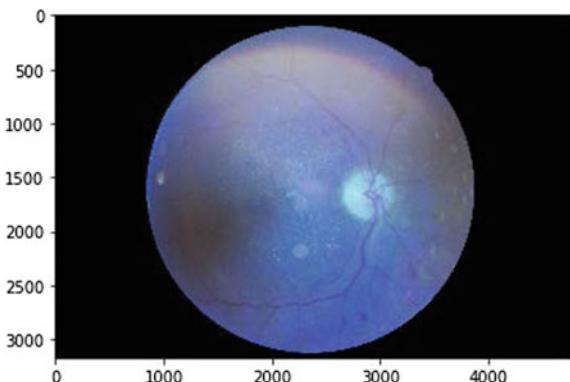
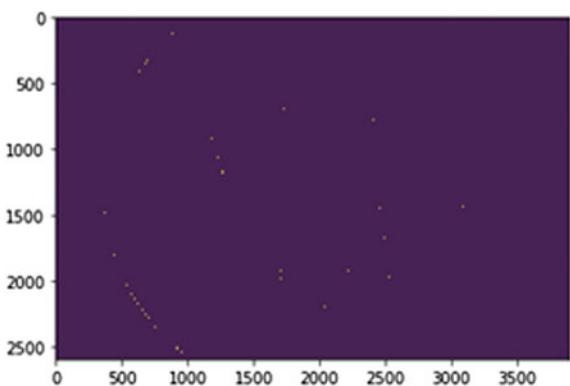
Fig. 6 Segmented microaneurysms



earliest symptom for the diagnosis of DR. They are the localized capillary dilations, saccular in structure, appearing in the form of clusters as small red dots or in isolation. They are 1–3 pixels in diameter [23]. The number of microaneurysms is determined by extracting the green component and subtracting the blood vessels. In the next step, contrast stretching is performed using Adaptive Histogram Equalization (AHE). The corresponding vessel pixels are painted with the average retina color. In the next step, a disc-based dilation operation highlights the microaneurysms. Edge detection is performed on the image using Canny edge detection method. On detection of boundaries, certain gaps and holes are identified which are filled using hole filling algorithms, for detection of possible microaneurysms. The images without boundaries are obtained, and edge detected images are then subtracted from images with boundaries. The microaneurysms are extracted based on their arrangement and dimensions using morphological operations and maximum number of pixels, to get the actual count of microaneurysms. The consequential imageries of this progression are shown in Fig. 5 and Fig. 6.

3.3.3 Exudates

Exudate is a fluid that filters from the blood vessels to form lesions or area of inflammation [1]. Exudates are discrete yellow-white intra-retinal deposits which can differ from small specks to larger patches and may develop into ring-like structures called circinate, with a diameter of 1–6 pixels [23]. They are situated in the posterior pole of the fundus. They appear as bright patterns and are well contrasted [22]. The shape and size of EXs vary significantly thus exhibiting irregular boundaries. EXs are largely made up of extracellular lipid resulting as a leakage from abnormal retinal capillaries. The recognition of exudates is vital for diagnosis of DR, and their rich contrast is helpful for their recognition. The optic disc and exudates are bright features. Figure 8 depicts the enhanced exudates. Canny edge detection procedure is used for edge recognition of exudates. The magenta component is removed from the CMY image to detect hard exudates. Binarization is performed depending on

Fig. 7 Enhanced exudates**Fig. 8** Segmented exudates

the standard deviation of the magenta component, using a particular threshold value. The binarized image is enhanced through transformation of the contour of the retina to white color and then adding the optic disc mask, which is extracted using the green and cyan components. Using the Hough transform, optic disc is eliminated, for exudate detection. Finally, an erosion operation is implemented beforehand the computation of the density of hard exudates. Figures 7 and 8 depict the enhanced exudates and segmented exudates, respectively.

3.3.4 Hemorrhages

Hemorrhages are structural deformations in the walls of the blood vessels with an increasing risk of blood leaking from the vessels [21]. The blood leaking expands in the area with low pressure producing asymmetrical, irregular shapes. They are usually 3–10 pixels in diameter but may expand based on the leakage caused [23]. In the extraction of hemorrhages, the green component of the image is extracted to intensify the image contrast. CLAHE is applied to upsurge the dynamic range

Fig. 9 Enhanced hemorrhages

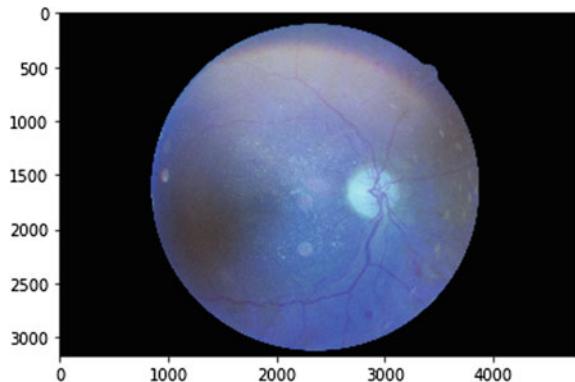
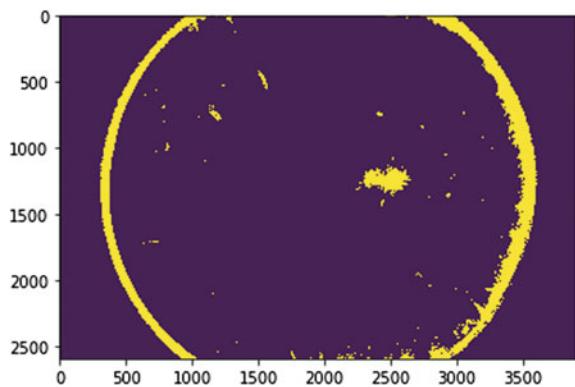


Fig. 10 Segmented hemorrhages



of contrast. The image is thresholded accordingly, and morphological operations are performed. This is followed by performing the image complement operation. Figures 9 and 10 depict the enhanced and segmented hemorrhages, respectively.

4 Results and Discussions

The proposed model using SVM is compared with a model previously proposed by Carrera et al. [16] which has also used SVM. The proposed model is also trained on four different kernels, namely linear kernel, polynomial kernel, sigmoid kernel and radial basis function (RBF) kernel. The model proposed by Carrera et al. [16] has considered three features of DR, namely blood vessels, microaneurysms and hard exudates, from 400 fundus images. The model proposed in this paper is a remodeling of Carrera et al. [16] upon Kaggle dataset which consists of 33,792 fundus images. On implementing the previous model, the system achieved a sigmoid kernel classification accuracy of 64.27% with a regularization, $C = 1$, using 4 DR features, namely

blood vessels, microaneurysms, exudates and hemorrhages. On the other hand, the proposed model has used three DR features such as blood vessels, microaneurysms and exudates and have achieved a sigmoid kernel accuracy of 69.09%. The regularization parameter is used to regulate the trade-off between low training error and a low testing error which indicates the ability of the classifier to generalize unseen data. Figure 11 depicts confusion matrix for four features using sigmoid kernel. Figure 12 depicts confusion matrix for three features using sigmoid kernel. Table 1 depicts a comparison on the performance of the models using sigmoid kernel and different features obtained from MESSIDOR dataset (test dataset), for DR detection.

From these results, it can be inferred that the proposed model which is a remodeling of Carrera et al. [16] has given better results due to the use of a larger dataset. The proposed model has experimentally shown better results and is capable of improving its classification accuracy by 5% on remodeling of Carrera et al. [16], using only three DR attributes and a larger dataset. In comparison to all the models reviewed in the literature which have used a very small and imbalanced dataset ranging from 100 to 1200 fundus images, which may lead to biasness in performance of the detector even while using an efficient and large margin classifier such as SVM, the proposed model has lower accuracy due to an imbalanced large dataset and not due to the use

Fig. 11 Confusion matrix for four features using sigmoid kernel

Actual	Predicted				
	0	1	2	3	4
[5251,	0,	849,	71,	16] 0	
[491,	0,	78,	5,	3] 1	
[1105,	0,	176,	14,	8] 2	
[181,	0,	30,	2,	1] 3	
[150,	0,	12,	4,	1] 4	
0	1	2	3	4	

Fig. 12 Confusion matrix for three features using sigmoid kernel

Actual	Predicted				
	0	1	2	3	4
[5751,	57,	272,	13,	94] 0	
[546,	6,	16,	1,	8] 1	
[1206,	5,	75,	2,	15] 2	
[197,	1,	13,	0,	3] 3	
[144,	0,	18,	0,	5] 4	
0	1	2	3	4	

Table 1 Comparison between the previous model [16] and the proposed model for DR detection

Sl. No	Models	Total No. of attributes/features	Sigmoid kernel accuracy (%)
1	Previous model [16]	4	64.27
2	Proposed model	3	69.09

of a small dataset. The proposed model makes promising improvements, through less overfitting and better generalization, and is hopeful of achieving much higher accuracy with the incorporation of various other DR features and advanced ML techniques such as Deep Learning (DL), in the near future.

5 Conclusion

DR is a chronic disease affecting mostly patients suffering from prolonged diabetes. It causes formation of DR features due to the rupture of blood vessels, causing abnormalities and retinal lesions which leads to blindness. The importance of a reliable and intelligent detection system, which can detect DR at a very early stage and identify various features reflecting possible signs of DR for early treatment, is very essential. In the process of detecting DR using an intelligent system, image enhancement and segmentation, feature extraction plays an important role in amplifying the performance of SVM classification. The SVM classifier implemented here achieves its highest classification accuracy using a sigmoid kernel, among all other kernels such as linear kernel, polynomial kernel and RBF kernel. The proposed model in comparison with a previously proposed intelligent system for DR detection has shown significant performance. However, the overall accuracy is less but the results are encouraging for a future clinical assessment to assimilate the existing algorithms in a tool for improved diagnosis of DR. Besides, more features are required to be extracted for texture analysis to improve accuracy and sensitivity of the proposed detector. Additionally, the SVM can be finely tuned by varying different parameters such as C value, gamma and kernels to obtain higher accuracy. Since, the training time of SVM increases with increase in the number of features, some other classification techniques or ML algorithms or in fact Deep Learning (DL) models could be introduced. This will lead to the reduction in training time, enhance faster extraction of features and help in efficient image classification such that larger datasets and feature sets could be adopted, thereby minimizing the rate of false responses.

Acknowledgements The authors would like to express their gratitude to the Department of Computer Science and Engineering, National Institute of Technology Silchar, for providing infrastructural facilities and support. The authors would also like to express their gratitude to Technical Education Quality Improvement Program (TEQIP-III) cell of National Institute of Technology, Silchar, for providing financial support and facilities.

References

1. Gandhi M, Dhanasekaran R (2013) Diagnosis of diabetic retinopathy using morphological process and SVM classifier. In: 2013 international conference on communication and signal processing, pp 873—877

2. Senapati RK (2016) Bright lesion detection in color fundus images based on texture features. *Bull Electr Eng Inf* 5(1):92–100
3. RAJA SS, Vasuki S (2015) Screening diabetic retinopathy in developing countries using retinal images. *Appl Med Inf* 36(1):13–22
4. Amin J, Sharif M, Yasmin M, Ali H, Fernandes SL (2017) A method for the detection and classification of diabetic retinopathy using structural predictors of bright lesions. *J Comput Sci* 19:153–164
5. Lachure J, Deorankar AV, Lachure S, Gupta S, Jadhav R (2015) Diabetic retinopathy using morphological operations and machine learning. In: 2015 IEEE international advance computing conference (IACC), pp 617–622
6. Vaishnavi J, Ravi S, Devi MA, Punitha S (2016) Automatic diabetic assessment for diabetic retinopathy using support vector machines. *Int J Control Theor Appl* 9(7):3135–3145
7. Asha PR, Karpagavalli S (2015) Diabetic retinal exudates detection using machine learning techniques. In: 2015 international conference on advanced computing and communication systems, pp 1–5
8. Decencière E, Cazuguel G, Zhang X, Thibault G, Klein JC, Meyer F, Marcotegui B, Quellec G, Lamard M, Danno R, Elie D (2013) TeleOphtha: machine learning and image processing methods for teleophthalmology. *Irbm* 34(2):196–203
9. Kaur P, Chatterjee S, Singh D (2019) Neural network technique for diabetic retinopathy detection. *Int J Eng Adv Technol (IJEAT)* 8(6):440–445
10. Porwal P, Pachade S, Kamble R, Kokare M, Deshmukh G, Sahasrabuddhe V, Meriaudeau F (2018) Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research. *Data* 3(3):25
11. Li B, Li HK (2013) Automated analysis of diabetic retinopathy images: principles, recent developments, and emerging trends. *Curr DiabRep* 13(4):453–459
12. Malathi K, Neduncelian R (2018) A recursive support vector machine (RSVM) algorithm to detect and classify diabetic retinopathy in fundus retina images. *Biomed Res.* <https://doi.org/10.4066/biomedicalresearch.29-16-2328>
13. Roychowdhury S, Koozekanani DD, Parhi KK (2013) DREAM: diabetic retinopathy analysis using machine learning. *IEEE J Biomed Health Inform* 18(5):1717–1728
14. Long S, Huang X, Chen Z, Pardhan S, Zheng D (2019) Automatic detection of hard exudates in color retinal images using dynamic threshold and SVM classification: algorithm Development and Evaluation. *Biomed Res Int* 2019:1–14
15. Sisodia DS, Nair S, Khobragade P (2017) Diabetic retinal fundus images: preprocessing and feature extraction for early detection of diabetic retinopathy. *Biomed Pharmacol J* 10(2):615–626
16. Carrera EV, Gonzalez A, Carrera R (2017) Automated detection of diabetic retinopathy using SVM. In: 2017 IEEE XXIV international conference on electronics, electrical engineering and computing (INTERCON), pp 1–4
17. Aravind C, PonniBala M, Vijayachitra S (2013) Automatic detection of microaneurysms and classification of diabetic retinopathy images using SVM technique. *Int J Comput Appl* 975(8887):1–5
18. Zohra BF, Mohamed B (2009) Automated diagnosis of retinal images using the support vector machine (SVM), Faculte des Science, Department of Informatique, USTO, Algerie
19. Jaya T, Dheeba J, Singh AN (2015) Detection of hard exudates in colour fundus images using fuzzy support vector machine-based expert system. *J Digit Imaging* 28(6):761–768
20. Sopharak A, Dailey MN, Uyyanonvara B, Barman S, Williamson T, New KT, Moe YA (2010) Machine learning approach to automatic exudate detection in retinal images from diabetic patients. *J Mod Opt* 57(2):124–135
21. Priya R, Aruna P (2013) Diagnosis of diabetic retinopathy using machine learning techniques. *ICTACT J Soft Comput* 3(4):1–13

22. Foeady ZA, Novitasari RCD, Asyhar HA, Firmansjah M (2018) Automated diagnosis system of diabetic retinopathy using GLCM method and SVM classifier. *Proc Electr Eng Comput Sci Inf* 5(1):154–160
23. Ege BM, Hejlesen OK, Larsen OV, Møller K, Jennings B, Kerr D, Cavan DA (2000) Screening for diabetic retinopathy using computer based image analysis and statistical classification. *Comput Methods Programs Biomed* 62(3):165–175

Chapter 27

Case-Based Expert System for Early Detection of Diabetic Retinopathy



Rahul Barman, Saroj Kumar Biswas, Dolly Das, Biswajit Purkayastha, and Malaya Dutta Borah

1 Introduction

Diabetic retinopathy (DR) is a medical condition in which the human retina is affected and can cause permanent blindness. According to the study, it has been found that DR affects most of the people who have been suffering from prolonged diabetes since 15–20 years [1]. DR has been a major medical condition due to diabetes mellitus. DR has been found to be prevalent mostly amongst the working generation. Thus, such kind of a medical condition needs to be detected and treated at an early stage such that blindness can be prevented from occurring. There are four stages in DR, and these are mild Non-Proliferative DR (NPDR), moderate NPDR, severe NPDR, and proliferative DR (PDR). DR can be detected on the basis of the existence of retinal lesions such as microaneurysms, foveal avascular zone, exudates, and hemorrhages [2–6], in the rear view of the human eye, i.e., the fundus. All these features have their respective stages of occurrence such as hemorrhages are significant features in the NPDR stage. The existence and identification of each of these features are significant and help the experts to know, from which stage of DR the patient has been suffering from and accordingly conduct the necessary treatment. Besides, the importance of having an expert system is highly necessitated to fulfill the absence or unavailability of experts if there exists a huge population suffering from diabetes mellitus or diabetes. In such cases, manual analysis may be feasible but time-consuming and hence early detection of DR may not be possible.

Thus, for such an efficient detection, it is important to model an expert system that can help in the identification of the features and can detect DR at an early stage. Thus, the purpose of diagnosing the disease can be fulfilled by using an expert system through proper acquisition of the fundus images. This fundus image can be

R. Barman · S. K. Biswas · D. Das (✉) · B. Purkayastha · M. D. Borah
Department of Computer Science and Engineering, National Institute of Technology Silchar,
Silchar, India

obtained from the human retina by performing a comprehensive eye examination. In this examination, the human retina is dilated using certain medically identified contrasting agents. The fundus images are then captured using a digitized fundus camera called Ophthalmoscope. The Ophthalmologist examines these fundus images to identify features such as microaneurysms, foveal avascular zone, exudates, hemorrhages, ruptured retinal blood vessels, cotton wool spots, intraretinal microvascular abnormalities for detection of DR. Hence, an expert system could be proposed that takes fundus images as input and classifies them into various categories of DR using these features. Different kinds of works have been done since decades, proposing various expert systems intelligent enough for the detection and smart computer-assisted systems which can perform automated analysis of the disease. Various techniques have been employed earlier for the detection and diagnosis of DR, such as Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM) [2], Neural Network (NN) [2], Support Vector Machine (SVM) [3–6], K-Nearest Neighbour (KNN) classifier [5, 6, 8], genetic algorithm [9], and various other ML techniques [3–7, 9–11].

Recently, a new Artificial Intelligence (AI) paradigm Case Based Reasoning (CBR) is widely used for image processing. CBR is a technique which solves a new problem by the experiences of the previous problems stored in a case base. The working principle of CBR is very similar to the human brain, which compares the previous problems with the new problem to propose a solution to the new one. CBR can also learn by acquiring new cases through use. There is no need to create any particular rule for solving a new problem in CBR as it can give a solution to a new problem through precedents. This makes CBR different than other AI techniques for problem-solving. Therefore, this paper proposes a method named Case Based Expert System for Diabetic Retinopathy (CBESDR) for early detection of DR which takes retinal fundus images as input and classifies them into five categories: 0-No DR, 1-Mild Non-Proliferative DR (NPDR), 2-Moderate Non-Proliferative DR, 3-Severe Non-Proliferative DR, and 4-Proliferative DR (PDR) to determine the severity of the disease. The proposed CBESDR extract features such as blood vessels, microaneurysms, exudates, and hemorrhages from the retinal fundus images with class labels and represent them as cases to store into the case base. When a new retinal image is given as input, CBESDR retrieves similar cases from the case base using Euclidean distance similarity measure, and reuse and revise them to classify the new case.

2 Literature Review

Kumar and Kumar [12] have discussed a solution for detection of DR by extracting the area and number of microaneurysms from the fundus images. The color fundus images are preprocessed to remove light variation, poor contrast, and noise. The methodology has considered features such as exudates, microaneurysms, and blood vessels, extracted from fundus images and uses Support Vector Machine (SVM) for

classification. SVM has classified images into two classes: healthy eye and DR eye. The sensitivity and specificity of the proposed system are 96% and 92%, respectively. The proposed system is only able to detect the disease in terms of yes and no, but could not detect the severity of the disease. Koul et al. [13] have proposed a model which detects DR using Neural Network (NN). In the first phase, the RGB retinal images are converted into grayscale images. Blood vessels and optic disc are extracted using edge detection techniques and removed from the fundus images. The training set is prepared based on the color feature of the input image. The system has been trained by the training set, and it can classify the diabetes and non-diabetes part of the image. The system produced 100% sensitivity, 95.83% specificity, and 96.67% accuracy, which is better than SVM. Chakrabarty [14] has used a deep convolutional neural network for detecting the DR using color fundus images. The network contains an input layer which takes preprocessed grayscale images as input followed by a 3-set combination of convolution. Each set consists of a convolution layer, a Rectified Linear Unit (ReLU) layer and a maxpooling layer. The system has achieved an accuracy of 91.67%. Surjandy et al. [15] have stated a solution for the traffic congestion problem using CBR. The pictures from the CCTV camera are converted into binary patterns (cases) using DECODE. The case is described as either empty(low), smooth(medium), or solid(high). The new pictures captured by the CCTV cameras are treated as a new case and are compared with the previous patterns. Lee et al. [16] have used the function of mask algorithm (Fmask) and CBR to improve the simple image simulation algorithm. Fmask is used to detect clouds, shadow of clouds, and water areas, and CBR is used to select appropriate reference imagery to compare with the target imagery, selected by image simulation. There are four steps of an improved simulation algorithm which are preprocessing, solar radiation modeling, selection of reference imagery and top-of-reflection simulation. The third step uses the CBR method to predict the result of new cases by considering the result of past cases. The improved simulation algorithm creates a better result with 95.59% accuracy.

3 The Proposed Case-Based Expert System for Diabetic Retinopathy (CBESDR)

The layout diagram of the proposed CBESDR is shown in Fig. 1. The CBESDR consists of five stages which are image enhancement, feature extraction, case representation, retrieve, and reuse and revise. In Fig. 1, solid line represents the creation of the case base and dotted line represents reasoning process of the model once a new image is given as a new input. The detailed of the stages is discussed below.

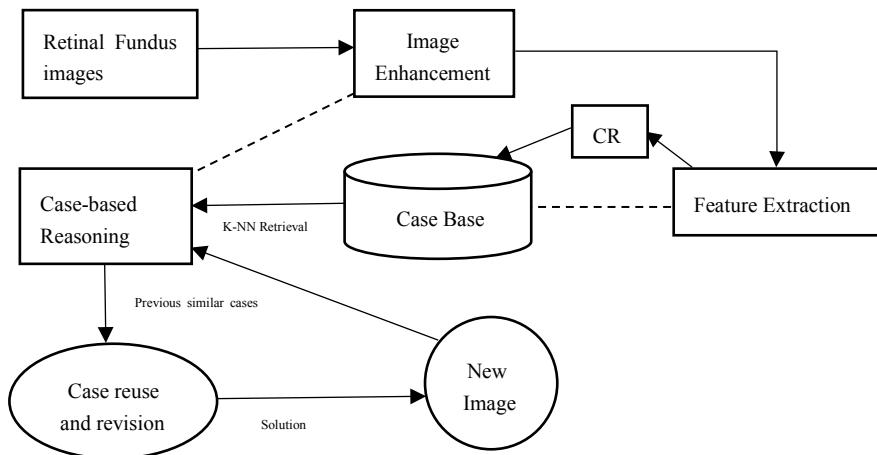


Fig. 1 Case-Based Expert System for DR (CBESDR)

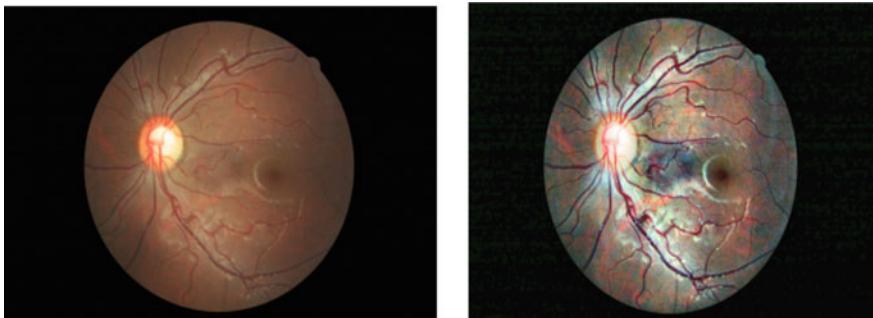


Fig. 2 Retinal image (left) and after using CLAHE retinal image (right)

3.1 Stage 1: Image Enhancement

Contrast Limited Adaptive Histogram Equalization (CLAHE) [17] is used for image enhancement which helps to denoise the image and enhance the contrast of the green component of the image. Image enhancement makes the feature extraction process effective. Figure 2 shows retinal image before and after image enhancement.

3.2 Stage 2: Feature Extraction

Feature extraction is an essential part of the proposed CBESDR system. The prominent features blood vessels, microaneurysm, exudates, and hemorrhages for DR

Table 1 Performance of CBESDR system for different k values

k	Accuracy (%)	k	Accuracy (%)
3	81.60	12	84.78
4	83.40	13	84.86
5	82.88	14	82.63
6	83.04	15	81.91
7	83.09	16	82.54
8	82.78	17	82.90
9	82.88	18	83.60
10	84.22	19	83.58
11	82.37	20	83.70

are extracted. Blood vessels are extracted by using the sequential filter which is a combination of morphological opening and closing operation. Microaneurysms are extracted by the morphological top-hat and opening operation on the gamma adjusted green channel of the retinal image. Exudates are the brighter spots in a retinal image. Hence, it has been extracted by using the thresholding. On finding the mean intensity of an image and using it as threshold value, hemorrhages are extracted from the retinal image.

3.3 Stage 3: Case Representation

A case stored in the case base consists of two parts: a problem part and a solution part. In CBESDR, the problem part is created by the features which are extracted from the fundus images, i.e., blood vessels, microaneurysms, exudates, and hemorrhages; and the solution part is the classification of the retinal image. The classifications of the images are 0-No DR, 1-Mild Non-Proliferative DR, 2-Moderate Non-Proliferative DR, 3-Severe Non-Proliferative DR, and 4-Proliferative DR. Table 1 shows the schematic representation of a case stored in case base.

3.4 Stage 4: Case Retrieval of KNN

Case retrieval is an important part of CBR technique because on the basis of retrieved cases the solution is given to a new problem. CBESDR retrieves k most similar cases using KNN mechanism where Euclidean Distance (ED) is used for similarity measurement. ED takes two inputs for similarity measurement, where one is query case q and another is stored case x . There are n numbers of features in the query case $q = \{q_1, q_2, \dots, q_n\}$, where feature values are numeric. The stored case x is represented as $x = \{x_1, x_2, \dots, x_n, x_c\}$ where x_1-x_n are attribute values and x_c is the

class value of x . The description of ED is given in Eq. (1).

$$\text{distance } (x, q) = \sqrt{\sum_{i=1}^n \text{diff}(x_i, q_i)^2} \quad (1)$$

where

- x_i is the feature value of i^{th} feature of the case stored in case base.
- q_i is the feature value of i^{th} feature of the query case

$$\text{diff } (x_i, q_i) = |x_i - q_i|$$

3.5 Stage 5: Case Reuse and Revision

CBR uses the top-most retrieved similar case as a solution to the new query problem. But it may happen that the top most similar case is an outlier, and it gives a false result to the query case. To avoid such a situation, more than one similar case is retrieved by CBESDR system. The CBESDR system classifies the query case using majority of voting by the retrieved cases. The performance of the CBESDR system is measured by the different values of the similar cases ($k = 3 \dots 20$). Therefore, the value of k is taken up to 30. The final value of k is the value which performs the best among all.

4 Results and Discussion

The method proposed in this manuscript uses Kaggle dataset, for experimental purpose. It consists of 35,126 fundus images. The pie chart below in Fig. 3 depicts the analysis of the dataset based on the distribution of DR grades. Tenfold cross-validation is executed to measure the performance. Classification accuracy is the performance measure for the proposed CBESDR system. 17,843 retinal images with type 0-No DR, type 1-Mild DR, type 2-Moderate DR, type 3-Severe DR, and type 4-Proliferative DR are used for the experimentation. Since the dataset is highly imbalanced, approximately 53% of the dataset is used with equal number of classes of all the grades of DR, to eliminate false results. The proposed methodology in this manuscript works on extraction of features such as density of blood vessels, microaneurysms count, density of exudates and density of hemorrhages. Table 1 shows the accuracies for different case (k) values. A bar chart is also shown in Fig. 4 to

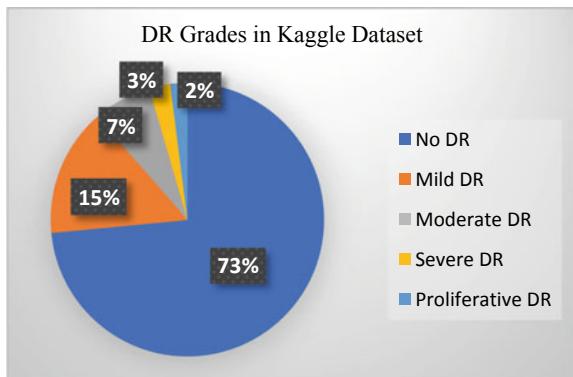


Fig. 3 Kaggle dataset analysis

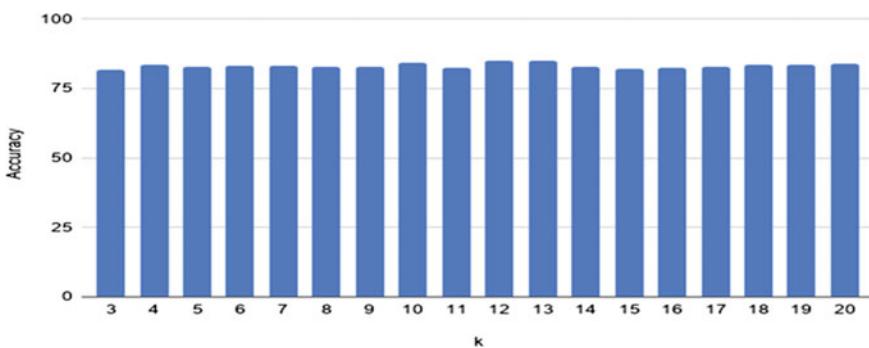


Fig. 4 Performance of CBESDR system for different k values

have better understanding of performances for different values of k . The CBESDR system produces the highest accuracy, 84.86% when the value of k is 13. In CBR, the performance of the system basically depends on the retrieved cases from the case base. During the case retrieval, all the features are treated with equal importance. In this manuscript, the performances of various ML techniques such as SVM, NN and Convolutional Neural Network (CNN) are analyzed in comparison with CBR performance for different case (k) values, in relation to detection of DR. CBR as a significantly traditional AI technique stores data in the form of cases, for solving new problems. Unlike SVM, NN, CNN which can only be used for the purpose of classification, CBR can be used for both regression and classification, which makes it an exceptional approach to DR detection in terms of ranges, during computation of enlargement in the area of retinal abrasions, which might possess continuous values.

5 Conclusion

Today, image data is widely used because of its availability. CBR gives acceptable performance over image data for classification. This paper proposes CBESDR system to classify DR into five classes: type 0, type 1, type 2, type 3, and type 4. The raw retinal images are taken, and image enhancement is done to have better feature extraction. Different feature extraction techniques are used to extract 4 prominent features namely blood vessels, exudates, microaneurysms, and hemorrhages responsible for DR. A case is represented using these features for disease classification. The experimental results of the proposed CBESDR system produces an acceptable performance of 84.86% accuracy. This implies that the proposed system can be used for early detection of DR. In the future, the performance of the system may be improved using feature weighting algorithm while retrieving similar cases from case base.

Acknowledgements The authors would like to express their gratitude to the Department of Computer Science and Engineering, National Institute of Technology Silchar for providing infrastructural facilities and support. The authors would also like to express their gratitude to Technical Education Quality Improvement Program (TEQIP-III) cell of National Institute of Technology Silchar for providing financial support and facilities.

References

1. Fong DS, Aiello L, Gardner TW, King GL, Blankenship G, Cavallerano JD, Ferris FL, Klein R (2004) Retinopathy in diabetes. *Diab Care* 27(suppl1):s84–s87
2. Asha PR, Karpagavalli S (2015) Diabetic retinal exudates detection using machine learning techniques. In: 2015 international conference on advanced computing and communication systems. IEEE, pp 1–5
3. Senapati RK (2016) Bright lesion detection in color fundus images based on texture features. *Bull Electr Eng Inf* 5(1):92–100
4. Ss RAJA, Vasuki S (2015) Screening diabetic retinopathy in developing countries using retinal images. *Appl Med Inf* 36(1):13–22
5. Amin J, Sharif M, Yasmin M, Ali H, Fernandes SL (2017) A method for the detection and classification of diabetic retinopathy using structural predictors of bright lesions. *J Comput Sci* 19:153–164
6. Lachure J, Deorankar AV, Lachure S, Gupta S, Jadhav R (2015) Diabetic retinopathy using morphological operations and machine learning. In: 2015 IEEE international advance computing conference (IACC), pp 617–622
7. Asha PR, Karpagavalli S (2015) Diabetic retinal exudates detection using machine learning techniques. In: 2015 international conference on advanced computing and communication systems, IEEE, pp 1–5
8. Hsiao HK, Liu CC, Yu CY, Kuo SW, Yu SS (2012) A novel optic disc detection scheme on retinal images. *Expert Syst Appl* 39(12):10600–10606
9. Decencière E, Cazuguel G, Zhang X, Thibault G, Klein JC, Meyer F, Marcotegui B, Quellec G, Lamard M, Danno R, Elie D (2013) TeleOphta: machine learning and image processing methods for teleophthalmology. *Irbm* 34(2):196–203

10. Porwal P, Pachade S, Kamble R, Kokare M, Deshmukh G, Sahasrabuddhe V, Meriaudeau F (2018) Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research. *Data* 3(3):25
11. Li B, Li HK (2013) Automated analysis of diabetic retinopathy images: principles, recent developments, and emerging trends. *Curr Diab Rep* 13(4):453–459
12. Kumar S, Kumar B (2018) Diabetic retinopathy detection by extracting area and number of microaneurysm from colour fundus Image. In: 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE pp 359–364
13. Kaur P, Chatterjee S, Singh D (2019) Neural Network technique for diabetic retinopathy detection. *Int J Eng Adv Technol (IJEAT)* 8(6) (2019). ISSN: 2249–8958
14. Chakrabarty N (November, 2018) A deep learning method for the detection of diabetic retinopathy. In: 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp 1–5
15. Anindra F, Soeparno H, Napitupulu TA (2018) CCTV traffic congestion analysis at Pejompong using case based reasoning. In: 2018 international conference on information and communications technology (ICOIACT)
16. Lee SB, La HP, Kim Y, Dalgeun L, Kim J, Park Y (2017) Improvement on image simulation from multitemporal Landsat images. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp 4874–4877
17. Setiawan AW, Mengko TR, Santoso OS, Suksmono AB (2013) Color retinal image enhancement using CLAHE. In: International conference on ICT for smart society (2013). <https://doi.org/10.1109/ICTSS.2013.6588092>

Chapter 28

Decision Making Using Interval-Valued Pythagorean Fuzzy Set-Based Similarity Measure



G. Punnam Chander and Sujit Das

1 Introduction

The dubiety of emergency environments affirms that the decision-makers often find it difficult to provide precise decision-making evaluation in the presence of different alternatives. To surmount this difficulty in decision making, it is essential to explore uncertain decision-making techniques to progress the efficiency that curtails the environmental damage. In medical sciences, the medical experts are regularly facing the problem to provide treatment based on the incomplete and imprecise information regarding the patient. Moreover, the medical experts are found to be bound to handle the problem approximately and concluding this limited and imprecise information. The experts in the domain of pattern recognition facing the similar problem of uncertainty to get the correct classification based on the similarities between unknown samples and patterns.

Managing uncertainty using a fuzzy set unpacked a new research direction in the decision-making studies. Fuzzy set theory was introduced by Zadeh [1] and since then it has been applied in various decision-making methods. Atanassov [2] defined IFS, which is advanced compared to the fuzzy set, which expresses uncertainty using the degree of membership, non-membership, and indeterminacy. As an extension to IFS, Yager and Abbasov [3] proposed PFS describes the vagueness proficiently in making decisions by the degree of membership and non-membership along with the indeterminacy, where the summation of squares of membership and non-membership grade is less than or equal to one. For example, assume if 0.6 and 0.8 are the membership and non-membership degrees, respectively; it is not valid under IFS and valid under

G. P. Chander · S. Das (✉)

Department of Computer Science and Engineering, NIT Warangal, Warangal, India

e-mail: sujit.das@nitw.ac.in

G. P. Chander

e-mail: punnamchander@student.nitw.ac.in

PFSs. So PFSs' are more generalized by its greater space of the membership degree which enhances wider applicability compared to space of the membership degree in IFS. Hence, PFS solves the indeterminate problems easily. Interval-valued intuitionistic fuzzy set (IVIFS) was introduced by Atanassov [4], which characterizes the membership, non-membership and indeterminacy degrees in an interval $[0,1]$. It expresses feature characteristics accurately. It is a tough job for the experts to accurately predict or evaluate an alternative by an exact membership value in uncertain situations, where experts prefer intervals rather than exact values. This idea motivated Peng and Yang [5] to introduce IVPFS, a generalization to PFS. It has been proved to be proficient and able to express and handle uncertainty, vagueness, and imprecision are represented by the degree of membership, non-membership, and indeterminacy which are defined in intervals.

In the literature, the idea of the similarity measure has a vital role in identifying the degree of likeliness between two entities in the FS theory. From the last few decades, researchers are paying their attention to measure the similarities and distances between two objects and applying these measurements in several domains like pattern recognition, multi-criteria decision making, medical diagnosis, and financial services. The commonly used distance measures are Euclidean distance [6], Hamming distance [6], and Hausdorff metric [7]. Recently, some research has been started in solving the distance measure in IVPFS [8, 9]. To mention as a note here, similarity measures have become the proficient mechanism over the distance measure for computing various fuzzy decision-making models. The research work on the similarity measures has been increased significantly, and many similarity measures for IFS, PFS, and IVIFS have been studied in the literature [10–12]. An adequate count of similarity and dissimilarity in a distance measure-based similarity measure of IFS is given by szimidt and kacprzyk [10] and further extended to a group of similarity measures and analyzed with the existing models. Peng [11] et al. developed a distance measure-based similarity measure for the PFSs. The transformation between similarity measures and entropies of an interval-valued fuzzy set (IVFS) based on the set of defined axioms, which is the relationship shown by Zeng and Li [12]. By extending the axioms formulated for entropy of IFS [13], Liu et al. [14] has defined entropy measures for IVIFSs based on a set of axioms. Xu [15] generalized and proposed some similarity measures from IFS to IVIFS. A similarity measure of IVIFS to solve problems in the domains of medical diagnosis, multi-criteria fuzzy decision making, and pattern recognition is proposed by Wei [16]. Singh [17] proposed a cosine-based similarity measure of IVIFS for pattern recognition. Luo [18] proposed a novel similarity measure for IVIFSs formulated from the transformed interval-valued intuitionistic triangle fuzzy numbers. The above similarity measures are represented in IVIFSs to perform medical diagnosis and pattern recognition. However, the similarity measure [15] is having a drawback that may not get the correct classification and leads to counterintuitive results shown in Table 1.

However, few research works are found in medical diagnosis and pattern recognition for measuring the distance and similarity between two IVPFSs. Recently, the study on entropy and similarity measures of IVIFSs in medical diagnosis, pattern recognition, multi-criteria decision making, and other areas getting good attention

from researchers. Due to the lack of information regarding the domain of the problems and inclining socioeconomic environment's intricacy, the conclusions may be provided by IVPFSs in the interval form characterized by its membership and non-membership function. As the IVPFS is more generalized than IVIFS, the work on entropies and similarity measures between IVPFSs in the above-mentioned areas need to be considered for the feasible, proficient, and intuitive results.

Many researchers have been contributing to the effectiveness of distance and similarity measures in the domains of multi-criteria decision making, medical diagnosis, and pattern recognition. However, none of the researchers have applied IVPFS-based similarity measures to accomplish medical diagnosis and pattern recognition. The objective of this paper is to present a decision-making approach using IVPFS-based similarity measure based on weighted distance measure and multiple parameters for choosing the right alternative under uncertain emergency decision making. This approach has been explained with numerical examples, and the results effectively classify and determine the best alternatives which are compared with existing models.

The remaining paper has been structured as: Sect. 2 presents few fundamental concepts relevant to IVPFSs and few existing similarity measures are discussed. A similarity measure based on weighted distance measure using multiple parameters is presented in Sect. 3. Section 4 shows the algorithmic approach. A detailed comparative study between the IVPFS-based similarity measure and existing similarity measures is illustrated by numerical examples in Sect. 5. Section 6 concludes this study.

2 Preliminaries

PFS and IVPFS are briefly reviewed in this section.

Definition 2.1 [19] A PFS F in a finite universe of discourse Z is stated as

$$F = \{(z, \mu_F(z), \nu_F(z)) | z \in Z\} \quad (1)$$

with a condition that $0 \leq (\mu_F(z))^2 + (\nu_F(z))^2 \leq 1$, where $\mu_F, \nu_F: Z \rightarrow [0, 1]$, respectively, represent membership and non-membership degree of the object $z \in Z$ for the set F and $\pi_F(z) = \sqrt{1 - (\mu_F(z))^2 - (\nu_F(z))^2}$ is the degree of indeterminacy (hesitation). Zhang and Xu [20] termed $(\mu_F(z), \nu_F(z))$ be a pythagorean fuzzy number (PFN) denoted by $F = (\mu_F, \nu_F)$.

Definition 2.2 [21] An IVPFS E in a finite universe of discourse Z and $\text{Int}[0,1]$ be the set of all closed subintervals of $[0,1]$, then E is given by

$$E = \{(z, \mu_E(z), \nu_E(z)) | z \in Z\}, \quad (2)$$

with a condition $0 \leq \sup\{(\mu_E(z))^2\} + \sup\{(\nu_E(z))^2\} \leq 1$ for every $z \in Z$, and $\mu_E(z)$, $\nu_E(z)$ are the closed intervals and their lower and upper bounds are represented by $\mu_E^-(z)$, $\mu_E^+(z)$, $\nu_E^-(z)$, $\nu_E^+(z)$, respectively. Here $\mu_E(z)$, denotes the membership degree with function $\mu_E(z): \text{Int}[0,1](z \in Z \rightarrow \mu_E(z) \subseteq [0,1])$ and $\nu_E(z)$ denotes non-membership degree with function $\nu_E(z): \text{Int}([0,1])(x \in X \rightarrow \nu_E(z) \subseteq [0,1])$ for $z \in Z$ to the set E , respectively. Therefore, E can also be stated as:

$$E = \{(z, [\mu_E^-(z), \mu_E^+(z)], [\nu_E^-(z), \nu_E^+(z)]) | z \in Z\}, \quad (3)$$

where $0 \leq \mu_E^-(z) \leq \mu_E^+(z) \leq 1$, $0 \leq \nu_E^-(z) \leq \nu_E^+(z) \leq 1$, $0 \leq (\mu_E^+(z))^2 + (\nu_E^+(z))^2 \leq 1$. As similar to PFS, for every $z \in Z$, the degree of hesitancy $\pi_E(z) = [\pi_E^-(z), \pi_E^+(z)] = [\sqrt{1 - (\mu_E^+(z))^2 - (\nu_E^+(z))^2}, \sqrt{1 - (\mu_E^-(z))^2 - (\nu_E^-(z))^2}]$. For an IVPFS E , the pair $([\mu_E^-, \mu_E^+], [\nu_E^-, \nu_E^+])$ is termed as interval-valued pythagorean fuzzy number (IVPFN) [21]. Assume K , V , and J be the IVPFSs in the universe of discourse Z and $S(K, V)$ be the similarity measure on the IVPFSs, where $S(K, V)$ is described by the mapping $S: \text{IVPFS}(Z) \times \text{IVPFS}(Z) \rightarrow [0,1]$ which satisfy the following properties [21]: (1) $0 \leq S(K, V) \leq 1$; (2) $S(K, V) = S(V, K)$; (3) $S(K, V) = 1$, if and only if $K = V$; (4) $S(K, K^C) = 0$, if and only if K is a crisp set; (5) If $K \subseteq V \subseteq J$, then $S(K, J) \leq S(K, V)$ and $S(K, J) \leq S(V, J)$.

3 Similarity Measure for Interval-Valued Pythagorean Fuzzy Sets

This section presents the similarity measure for IVPFSs between two attributes based on weighted distance measure with multiple parameters.

Definition 3.1 [8] Given two IVPFS $Q = \{(z_i, [\mu_Q^-(z_i), \mu_Q^+(z_i)], [\nu_Q^-(z_i), \nu_Q^+(z_i)])| z_i \in Z\}$ and $T = \{(z_i, [\mu_T^-(z_i), \mu_T^+(z_i)], [\nu_T^-(z_i), \nu_T^+(z_i)])| z_i \in Z\}$, in a finite universe of discourse Z , where $Z = \{z_1, z_2, \dots, z_n\}$. Then the similarity measure for Q and T is computed as

$$S_p^w(Q, T) = 1 - p \sqrt{\frac{1}{2^{p+1} t_k^p} \sum_{i=1}^n w_i \begin{pmatrix} |(t_k - 1)((\mu_Q^-(z_i))^2 + (\mu_Q^+(z_i))^2 - (\mu_T^-(z_i))^2 - (\mu_T^+(z_i))^2)|^p + \\ |((\nu_Q^-(z_i))^2 + (\nu_Q^+(z_i))^2 - (\nu_T^-(z_i))^2 - (\nu_T^+(z_i))^2)|^p + \\ |(t_k - k)((\nu_Q^-(z_i))^2 + (\nu_Q^+(z_i))^2 - (\nu_T^-(z_i))^2 - (\nu_T^+(z_i))^2)|^p \\ - k((\mu_Q^-(z_i))^2 + (\mu_Q^+(z_i))^2 - (\mu_T^-(z_i))^2 - (\mu_T^+(z_i))^2)|^p \end{pmatrix}} \quad (4)$$

where the three parameters $t_1, t_2, p \in [1, \infty]$. p is the L_p – norm; the level of uncertainty related to k is identified by t_k on a condition $t_k \geq k + 1$ and $k \geq 0$; k is the slope. Assume that F and H be two IVPFSs in a finite unit of discourse $Z =$

$\{z_1, z_2, \dots, z_n\}$. The weighted similarity measure for IVPFSs F and H is, $S_p^w(F, H)$ which is defined by

$$S_p^w(F, H) = 1 - \sqrt{p} \frac{1}{2^{p+1} t_k^p} \sum_{i=1}^n w_i \left(\begin{array}{l} |(t_k - 1)((\mu_F^-(z_i))^2 + (\mu_F^+(z_i))^2 - (\mu_H^-(z_i))^2 - (\mu_H^+(z_i))^2)|^p + \\ |(v_F^-(z_i))^2 + (v_F^+(z_i))^2 - (v_H^-(z_i))^2 - (v_H^+(z_i))^2|^p + \\ |(t_k - k)((v_F^-(z_i))^2 + (v_F^+(z_i))^2 - (v_H^-(z_i))^2 - (v_H^+(z_i))^2)|^p + \\ |-k((\mu_F^-(z_i))^2 + (\mu_F^+(z_i))^2 - (\mu_H^-(z_i))^2 - (\mu_H^+(z_i))^2)|^p \end{array} \right) \quad (5)$$

where w_i is the weight of the element z_i with $\sum_{i=1}^n w_i = 1$ and $w_i \geq 0$; $t_k \geq k + 1$, $k \geq 0$. The relevant properties of the similarity measure between IVPFSs F, H , and I are:

(1) $0 \leq S_p^w(F, H) \leq 1$, (2) $S_p^w(F, H) = 1$ if and only if $L = H$, (3) $S_p^w(F, H) = S_p^w(H, F)$, (4) $S_p^w(F, F^C) = 0$ if and only if F is a crisp set; (5) Let $F \subseteq H \subseteq I$ then $S_p^w(F, I) \leq S_p^w(F, H)$ and $S_p^w(F, I) \leq S_p^w(H, I)$.

4 Algorithmic Approach

In this section, we present an algorithmic approach for decision making using IVPFS-based similarity measure.

Step 1 Assume that there exists M^i samples defined by IVPFSs in the universe of discourse Z , $Z = \{z_1, z_2, \dots, z_n\}$ and $M^i = \{< z_j, [\mu_{M^i}^-(z_j), \mu_{M^i}^+(z_j)], [v_{M^i}^-(z_j), v_{M^i}^+(z_j)] > | z_j \in Z \}$ $i, j = 1, 2, \dots, p$ and a test sample $N = \{< z_j, [\mu_N^-(z_j), \mu_N^+(z_j)], [v_N^-(z_j), v_N^+(z_j)] > | z_j \in Z \}$ which is defined by IVPFS to be classified.

Step 2 Compute the similarity measure $S(M^i, N)$ for M^i and N using (4).

Step 3 Choose the largest measure, i.e., $S(M^k, N)$ from $S(M^i, N)$, where $i = 1, 2, \dots, p$ and k is the k^{th} sample which belongs to $[1, p]$.

Example 4.1 Consider three medical patterns P^i , $i = 1, 2, 3$, and two medical samples S_1 and S_2 expressed by IVPFSs in a feature space $X = \{x_1 | x_1 \in X\}$ with a weight vector $w = 1$. The medical patterns P^i , $i = 1, 2, 3$ expressed using IVPFSs are: $P_1 = \{\langle x_1, [0.6, 0.7], [0.1, 0.2] \rangle\}$; $P_2 = \{\langle x_1, [0.8, 0.9], [0.0, 0.1] \rangle\}$; and $P_3 = \{\langle x_1, [0.7, 0.8], [0.1, 0.2] \rangle\}$. The two medical samples are expressed in IVPFSs which are given as $S_1 = \{\langle x_1, [0.4, 0.5], [0.3, 0.4] \rangle\}$; and $S_2 = \{\langle x_1, [0.7, 0.8], [0.2, 0.4] \rangle\}$; The implementation of the similarity measure procedure will classify the samples S_1 and S_2 to one of the patterns P_1 , P_2 , and P_3 based on the similarity degree between them. The respective degree of similarity for S_1 and S_2 for patterns P_1 , P_2 , P_3 are measured by (4) and are $S_p^w(P_1, S_1) = 0.82$, $S_p^w(P_2, S_1) = 0.613$, $S_p^w(P_3, S_1) = 0.726$; $S_p^w(P_1, S_2) = 0.931$, $S_p^w(P_2, S_2) = 0.861$, $S_p^w(P_3, S_2) = 0.975$; The maximum degree of similarity is obtained between S_1 and P_1 , and between S_2 and P_3 , respectively, be

$S_p^w(P_1, S_1) = 0.82$ and $S_p^w(P_3, S_2) = 0.975$. Therefore, the classification of S_1 and S_2 can be determined by: $S_1 \leftarrow P_1$, $S_2 \leftarrow P_3$.

5 Applications and Comparative Analysis

IVPFSs can model and process the uncertain information in more precise manner. In this section, the extant similarity measure, presented in Eq. (4), is adopted to perform medical diagnosis and pattern recognition. Numerical problems are solved under IVPFSs, and the results obtained by this similarity measure have been compared with some existing models to demonstrate the practicability.

Example 5.1 [22] Consider four building block materials, $A_i = 1, 2, 3, 4$ and a pattern B, expressed by IVPFSs in a feature space $X = \{x_1, x_2, x_3, \dots, x_{12}\}$ with a weight vector $w = (0.1, 0.05, 0.08, 0.06, 0.03, 0.07, 0.09, 0.12, 0.15, 0.07, 0.13, 0.05)$. The objective is to find the building material to which the pattern B belongs. We consider the dataset as given in [22]. The degree of similarity $S(A_i, B)$ is computed for each IVPFSs A_i and B by the Eq. (4). We can determine the pattern B to one of the four blocks by the largest similarity value. Since $S_p^w(A_1, B) = 0.603$, $S_p^w(A_2, B) = 0.56$, $S_p^w(A_3, B) = 0.83$, $S_p^w(A_4, B) = 0.97$, evidently $S_p^w(A_4, B)$, the similarity value between A_4 and B leads with the highest value. So the pattern B belongs to pattern A_4 . A comparative study of the presented approach with the existing approaches is given in Table 1.

Medical Diagnosis: To perform medical diagnosis, many theories have been presented in the literature to solve problems in medical sciences by using various methods in various ways [23, 24]. In this subsection, we solve medical diagnosis problems using IVPFS-based similarity measure as presented in Sect. 4.

Example 5.2 [25] For medical diagnosis of headache, the example uses the patient's degree $M_Q(p, s)$, $N_Q(p, s)$, and conformability degree $M_R(s, d)$, $N_R(s, d)$, where $M_Q(p, s)$ is membership degree, $N_Q(p, s)$ is the non-membership degree of

Table 1 Pattern recognition results compared with the presented similarity measure with $p = 1$, $t = 2$, $t_k = 3$ in S_p^w

	$S(A_1, B)$	$S(A_2, B)$	$S(A_3, B)$	$S(A_4, B)$	Classification
S_1 [15]	0.59	0.58	0.81	0.97	A_4
S_2 [15]	0.53	0.53	0.79	0.94	A_4
S_W [16]	0.48	0.47	0.74	0.94	A_4
S^P [18]	0.60	0.58	0.85	0.97	A_4
S_p^w (proposed)	0.60	0.56	0.83	0.97	A_4

Table 2 Similarity measure for patient P_1

T	Migraine	Tension	Cluster
P_1	0.8793	0.7996	0.7957

patient–symptom matrix, and $M_R(s, d)$ is membership degree, $N_R(s, d)$ is the non-membership degree of symptom–disease matrix. Assume that the symptoms ($M_5, M_8, M_{12}, M_{15}, M_{18}, M_{19}$) of patient P_1 are related to migraine, (T_3, T_6, T_{10}) related to tension headache and (C_4, C_{11}) related to cluster headache. For dataset please refer to [25]. Aim is to decide the possible disease for patient P_1 . Here we find the similarity measures of patient P_1 with the mentioned diseases like migraine, tension headache, cluster headache using the procedure mentioned in Sect. 4 and the computed result is given in Table 2.

The largest value of similarity measure obtains proper diagnosis for the patient P_1 . From the result set obtained in Table 2, which is precise and experts can diagnose the patient P_1 suffers from migraine.

Example 5.3 [18] Given four classes of diagnoses $D_i = 1, 2, 3, 4$, namely Viral fever, Typhoid, Pneumonia and stomach problem, and $X = \{s_1, s_2, s_3, s_4\} = \{\text{Temperature, Cough, Headache, Stomach pain}\}$ are the symptoms. The representation of disease–symptom matrix by IVPFSSs is given in Table 3.

We assume that the symptoms associated with patient P is represented as $P = \{\langle s_1, [0.4, 0.5], [0.1, 0.2] \rangle, \langle s_2, [0.7, 0.8], [0.1, 0.2] \rangle, \langle s_3, [0.9, 0.9], [0.0, 0.1] \rangle, \langle s_4, [0.3, 0.5], [0.2, 0.4] \rangle\}$. By using the similarity measures, our objective is to decide the possible disease for the patient P . Table 4 shows the obtained similarity measures opted by the existing methods and the presented method.

Based on the results computed in Table 4, it can be distinguished that D_2 has the largest degree of similarity by using the presented similarity measure for IVPFSSs, which is similar to the other methods. Thus, the patient P is classified into the disease D_2 by the similarity measure. Therefore, the patient P is diagnosed to be suffering with typhoid. Nevertheless, the proposed similarity measure generates intuitive results by adequately considering membership and non-membership grade as

Table 3 Disease–symptom matrix

s_1	s_2	s_3	s_4
$D_1 [0.8, 0.9], [0.0, 0.1]$	$[0.7, 0.8], [0.1, 0.2]$	$[0.5, 0.6], [0.2, 0.3]$	$[0.6, 0.8], [0.1, 0.2]$
$D_2 [0.5, 0.6], [0.1, 0.3]$	$[0.8, 0.9], [0.0, 0.1]$	$[0.6, 0.8], [0.1, 0.2]$	$[0.4, 0.6], [0.1, 0.2]$
$D_3 [0.7, 0.8], [0.1, 0.2]$	$[0.7, 0.9], [0.0, 0.1]$	$[0.4, 0.6], [0.2, 0.4]$	$[0.3, 0.5], [0.2, 0.4]$
$D_4 [0.8, 0.9], [0.0, 0.1]$	$[0.7, 0.8], [0.1, 0.2]$	$[0.7, 0.9], [0.0, 0.1]$	$[0.8, 0.9], [0.0, 0.1]$

Table 4 Computed results

	$S(D_1, P)$	$S(D_2, P)$	$S(D_3, P)$	$S(D_4, P)$	Classification
S_1 [15]	0.81	0.89	0.86	0.84	D_2
S_2 [15]	0.73	0.80	0.78	0.73	D_2
S_W [16]	0.82	0.80	0.79	0.77	D_2
S^P [18]	0.83	0.89	0.87	0.85	D_2
S_p^w (proposed)	0.88	0.94	0.91	0.89	D_2

a representation in IVPFSs and hence proves that it can deal in medical diagnosis problems in practical.

6 Conclusion

Researchers proposed wide variety of similarity measures based on IVIFSs in the field of medical diagnosis. To contrast, no similarity measures based on IVPFS has been applied in this field specifically. This study has presented a decision-making approach using IVPFS-based similarity measure and applied it to medical diagnosis and pattern recognition problem. Furthermore, a comparative analysis has been demonstrated, which provides promising results to solve problems in the fields of medical diagnosis. In the future, researchers can extend this study to investigate entropy-based similarity measures for IVPFS and other extended fuzzy sets.

References

1. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353
2. Atanassov KT (1986) Intuitionistic fuzzy sets. *Fuzzy Sets Syst* 20(1):87–96
3. Yager RR, Abbasov AM (2013) Pythagorean membership grades, complex numbers, and decision making. *Int J Intell Syst* 28(5):436–452
4. Atanassov KT, Gargov G (1989) Interval-valued intuitionistic fuzzy sets. *Fuzzy Sets Syst* 31:343–349
5. Peng X, Yang Y (2016) Fundamental properties of interval-valued pythagorean fuzzy aggregation operators. *Int J Intell Syst* 31(5):444–487
6. Szmidt E, Kacprzyk J (2000) Distances between intuitionistic fuzzy sets. *Fuzzy Sets Syst* 114(3):505–518
7. Grzegorzewski P (2004) Distances between intuitionistic fuzzy sets and/or interval-valued fuzzy sets based on the hausdorff metric. *Fuzzy Sets Syst* 148(2):319–328
8. Peng X, Li W (2019) Algorithms for interval-valued pythagorean fuzzy sets in emergency decision making based on multiparametric similarity measures and WDBA. *IEEE Access* 7
9. Wang YW (2018) Interval-valued pythagorean fuzzy TOPSIS method and its application in student recommendation. *Math Pract Theor* 48(5):108–117
10. Szmidt E, Kacprzyk J (2004) A concept of similarity for intuitionistic fuzzy sets and its application in group decision making, vol 2. In: Proceedings of international joint conference on

- neural networks and IEEE international conference on fuzzy systems. Budapest, Hungary, pp 25–29
11. Peng X, Yuan H, Yang Y (2017) Pythagorean fuzzy information measures and their applications. *Int J Intell Syst* 32(10):991–1029
 12. Zeng WY, Li HX (2006) Relationship between similarity measure and entropy of interval-valued fuzzy sets. *Fuzzy Sets Syst* 157:1477–1484
 13. Szmidt E, Kacprzyk J (2001) Entropy for intuitionistic fuzzy sets. *Fuzzy Sets Syst* 118(3):467–477
 14. Liu XD, Zhang SH (2005) Entropy and subsethood for general interval-valued intuitionistic fuzzy sets, vol LNAI 3613. In: Wang L, Jin Y (eds) FSKD 2005. Springer-Verlag, Berlin Heidelberg, pp 42–52
 15. Xu ZS, Chen J (2008) An overview of distance and simiarity measures of intuitionistic fuzzy sets. *Int J Uncertain Fuzziness Knowl-Based Syst* 16:529–555
 16. Wei CP, Wang P, Zhang YZ (2011) Entropy, similarity measure of interval-valued intuitionistic fuzzy sets and their applications. *Inf Sci* 181:4273–4286
 17. Singh P (2012) A new method on measure of similarity between interval-valued intuitionistic fuzzy sets for pattern recognition. *J Appl Comput Math* 1
 18. Luo M, Liang J (2018) A novel similarity measure for interval-valued intuitionistic fuzzy sets and its applications. *Symmetry* 10:441
 19. Yager RR (2014) Pythagorean membership grades in multicriteria decision making. *IEEE Trans Fuzzy Syst* 22(4):958–965
 20. Zhang X, Xu Z (2014) Extension of TOPSIS to multiple criteria decision making with pythagorean fuzzy sets. *Int J Intell Syst* 29(12):1061–1078
 21. Zhang X (2016) Multicriteria pythagorean fuzzy decision analysis: a hierarchical QUALIFLEX approach with the closeness index-based ranking methods. *Inf Sci* 330:104–124
 22. Xu ZS (2007) On similarity measures of interval-valued intuitionistic fuzzy sets and their application to pattern recognitions. *J Southeast Univ* 23:027
 23. Das S, Roy BK, Kar MB, Kar S, Pamukar D (2020) Neutrosophic fuzzy set and its application in decision making. *J Ambient Intell Human Comput*. <https://doi.org/10.1007/s12652-020-01808-3>
 24. Si A, Das S, Kar S (2019) Extension of TOPSIS and VIKOR method for decision-making problems with picture fuzzy number, vol 1112. In: Mandal J, Mukhopadhyay S (eds) Proceedings of the global AI congress 2019. Advances in intelligent systems and computing. Springer, Singapore, pp 563–577. https://doi.org/10.1007/978-981-15-2188-1_44
 25. Sanchez E (1979) Medical diagnosis and composite fuzzy relations. *Advances in fuzzy set theory and applications*. Elsevier Science Ltd., pp 437–444

Chapter 29

Forest Combustion Recognition Using Deep Learning



Kota Jahnavi, Hari Kishan Kondaveeti, and Asish Kumar Dalai

1 Introduction

As the span in the business of forest conservation is briskly increasing to encounter human needs, preserving rare species of plants are increasingly important in addition to endure our life in this extensive world. Human greediness is playing a crucial role in the result of forest combustion, and due to their selfish nature of desiring unessential products for leading a luxurious life, they are deliberately or unwillingly becoming the main cause for combusting forests.

Forest combustion impacts the environment and ruins our lives and future generations may have a greater effect due to the extinction of forests. Incapacitation, i.e., the state of being disabled or unable to function, is one of the most key factors in the overall activities we do for preventing the forest combustion from the occurrence. We do follow the news day to day and get the update saying that typically, we have about more than 1,000,000 wildfires in the US every year, and over 10 a million acres of land in India have been destroyed due to this combustion. UPADHYAY detailed and summarized to understand easily, took a review for forest fires in this journal [1].

However, it is much more difficult to predict/detect the combustion in a thinly dispersed forest area and also by using few ground-based methods like using camera modules or through video, etc. So, here, we are going to train the model in such a

K. Jahnavi (✉) · H. K. Kondaveeti · A. K. Dalai

Department of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

e-mail: jahnavi.kota@vitap.ac.in

H. K. Kondaveeti

e-mail: kishan.kondaveeti@vitap.ac.in

A. K. Dalai

e-mail: asish.d@vitap.ac.in

way that if you gave an image/picture of the forest, it will detect whether the forest is with fire or without fire, and if it recognizes the fire picture, it puts on an alarm sound. This helps us to find out the forest combustion in an easy way by staying at home.

The *convolution neural networks (CNN)* [2] are a class of deep neural networks and aim at focusing on the images which we give the model to train and test with appropriate techniques so that it captures the image and provides us with better and accurate results.

Through this project, we can meet the main intention of the client. Here, the existing system does not make any detection of the fire on its own unless we place/have any sensors to identify the fire. So, for this sake, in turn humans need to waste their time to go and check around the forest in search of fire. Without the approach through convolution neural networks algorithm, detecting fire might be very tedious and time consuming, resulting in users to refraining from using these applications for long periods. To overcome this problem, we need to develop a new system that can predict or detect the fire by using images developed by CNN (analyzing visual imagery). So, by using deep learning techniques—CNN, we can achieve the goal of our project.

In this paper, we proposed an approach to detect the fire which would be entirely different when compared to other methods for recognition of combustion in the forest. The technique which we work on here is very easy to use, efficient, low budget, and trouble-free access to implement. The main objective of our project is to control forest fires and strengthen forest protection and create a user-friendly environment where anyone should be able to use it. Also, they should be able to access the software from anywhere and anytime if needed. More specifically,

1. We propose a CNN algorithm which is a deep learning technique to recognize the combustion through images in a facile manner.
2. We train and test all the images to get the perfect accuracy so that the model which we use will not give any false predictions.
3. Finally, we demonstrate how automatic code generation and model implementation techniques can be used to detect the images whether it comes under the category of with fire or without fire.

Section 2 consists of overview of the approach, and Sect. 3 consists of the proposed model. Experimental analysis is presented in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 An Overview of the Approach

Nature is a vital property to enjoy in our life. Because of that alluring environment, it provides us with the needful things we want. Now, it is our responsibility to save, protect, and prevent them from combustion. At the same time, it is dangerous for us because when the forest is under fire, it emits a lot of CO₂ which leads to climate

change and global warming. So, forest fire should be detected at an early stage. N. SHAHAM has gone through the restoration of forests and worked on fire prevention methods [3]. In the same way, I tried to implement an easy way of preventing fire by building a model using CNN (Fig. 1). Due to solitary, inaccessibility, sturdy weather, early recognition of fire in the forest is a difficult task.

So, we have developed a model to detect forest fire combustion. Anyways to do it manually, we need a lot of manpower so to overcome that we are implementing the *convolutional neural network (CNN)*. Model, which is deep learning the technique to predict the fire in a forest.

Model analysis Deep learning will permit us to instruct a particular task despite making the system or machine how to learn. We can make use of different datasets with images to train a specific model, or we can use a simple training sets and ask it to learn by itself. The results produced by them are quick and can learn overtime in just a flash of seconds. It can be done with images in the form of datasets and can be trained, tested, and predicted accurately, and they can even crucially recognize patterns through camera modules, satellites, sensors.

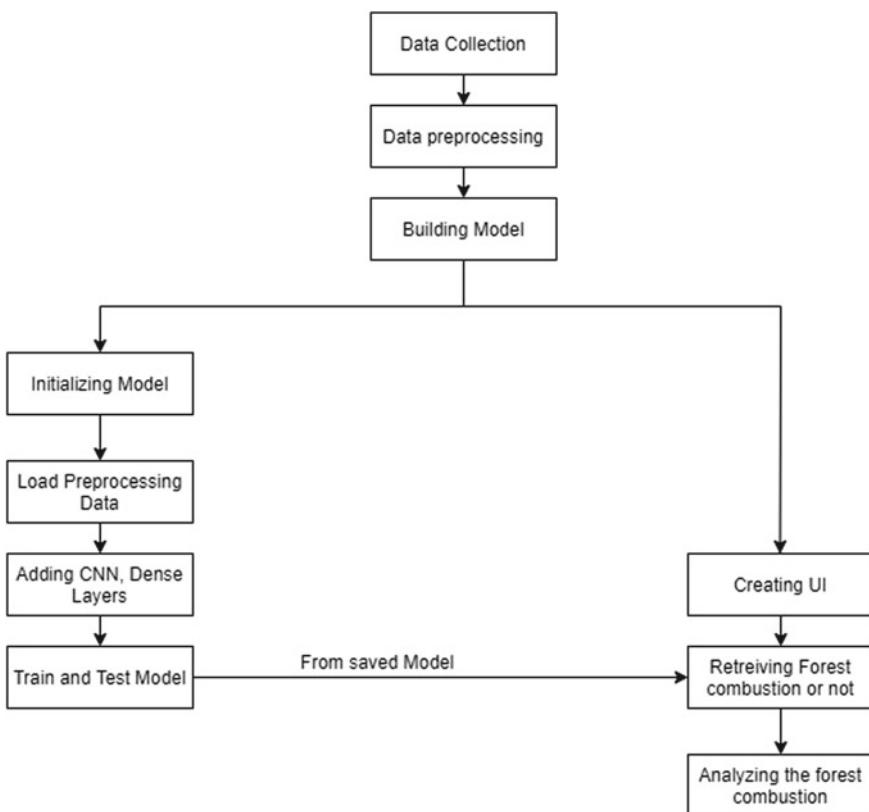


Fig. 1 Architecture overview

The toolkit This integrated model driven the Web development framework is based on the standard, open interfaces as provided by Anaconda Navigator which is a desktop *graphical user interface (GUI)*, and it includes Jupyter notebook an open-source Web application, and Spyder a powerful scientific environment written in Python and Flask a Web framework provides tools, libraries, and technologies that allow building a Web application, which we use it as the main tool to develop the interface.

Existing methods and recent works Forest fire recognition uses many of the existing methods like satellite-based systems, optical sensors, digital camera wireless sensor networks, and GPS few of them are discussed by Ahmad A. A. Alkhatib according to this journal [4]. This system is useful for all the people around the globe who want to detect the fire when occurred in forest areas. Our system is a user-friendly Web application that is used to make human life easier just by giving the image they can recognize whether it is with fire or not with zero effort. There are a few recent forest fire disaster management works published by NIDM, Government of India [5]. Some of the effects around the globe are no letup in global rainforest loss as coronavirus brings new danger, Wildfires rage in Brazil's Amazon Rain forests. A lot of works are being done to prevent such types of disasters across the globe. Now, we are working on this project to recognize combustion within a fraction of seconds.

3 Proposed Model

3.1 Method

The algorithm which we have used is based on a deep learning technique named CNN, and there are many other techniques in deep learning with AI but we choose CNN because we are working with images, and it is the best way to predict images accurately.

A convolution neural network is a deep learning algorithm by which we can take the image in the form of input and give priority to various types of objects in the image and classify/differentiate image by image. This algorithm gives major importance to the color of the image/objects [6].

CNN is a combination of convolution layers and neural network. We use neural network for image processing, and it consists of four layers—input layer, convolution layer, pooling, and dense layer. The dense layer contains three layers—input, hidden, and output layers.

3.2 Speculative Approach

While testing the accuracy following a specific approach is very important so the approach which we followed is using activation function rectified linear units (ReLU). This is used to activate neurons at a time, and if we do not know which approach to use we can use ReLU, epochs should be mentioned appropriately to get the accuracy we have given 100 epochs and steps per epochs, and validations steps will be calculated according to the samples and batch size. So, during this duration, we need to experimentally test by changing the epochs until we get perfect or good accuracy as this is the main important step to predict our model. We also used max pooling, flatten, convolution, Adam optimizer to resize, correcting the edges of images and to reduce loss in the model to increase the accuracy.

3.3 Solution

The proposed forest fire prediction model is based on neural networks (CNN) incorporated into an alert system. The developmental approach of the proposed system includes two modules:

1. Forest fire identification: Identification of fire-affected areas.
2. Fire management: Remedial measures for forest fire are all about detecting where the fire is initiated in the forest.

In our proposed system, we made an effort to classify the images of the forest with fire and forest without fire for further by using convolution neural networks [7]. The forest dataset of train and test has been selected as the working GUI for this approach. Here, we select a sample of few images (approx. 100) with fire and without fire segregate them into train and test and model predicts them perfectly depending upon accuracy.

An investigation of our problem's decision methods of process and anomaly of the problem illustration and initial data impact on obtained outputs. When we train and test the data, we have model loss occurring at that time which is nothing but a prediction error of the neural network.

Model loss is used to calculate the gradients, and these gradients are used to update weights of *neural networks*. The main goal of training a model is to find a set of weights and biases that have low loss, i.e., model can predict images with accurate results. If the model's prediction is ideal, the loss is said to be zero; or else, the loss is greater (Fig. 2).

Application building Image is given as input to the model, preprocessed, train the CNN classifier, test the given data (images), and know the accuracy of the model through a loss so that we can predict the images more precisely. Once this process is done, we are ready with the application building using a Web framework *Flask*, an image is given to the model to determine whether the forest is with fire or without fire with an accurate prediction (Fig. 3).

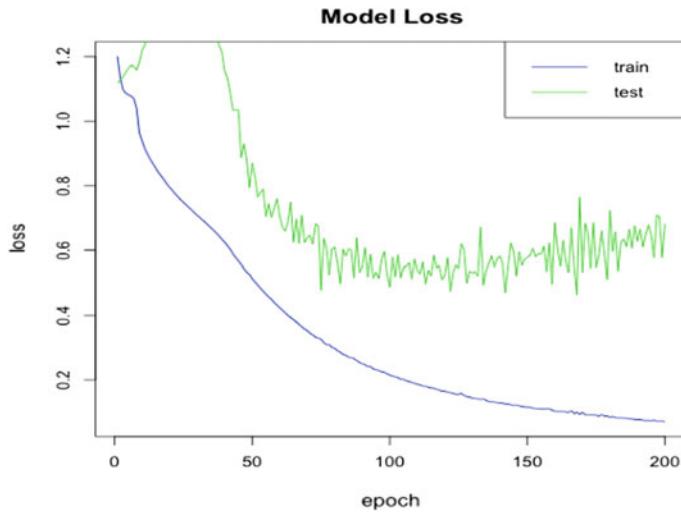


Fig. 2 Graph to represent train and test models

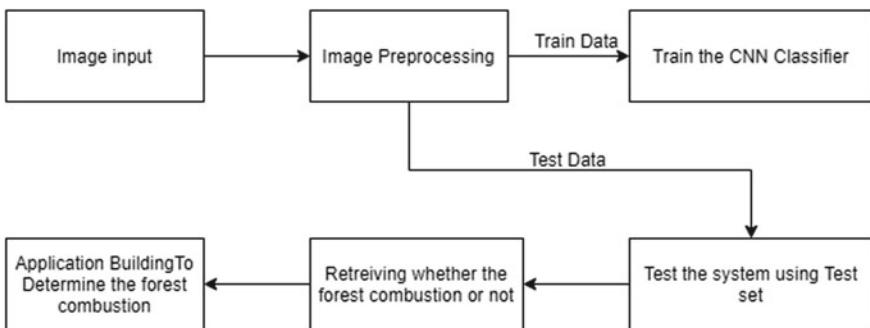


Fig. 3 Application building

3.4 Importing Building Libraries, Layers, and Their Working

We added different layers like dense, convolution, max pooling, flatten, and define our deep learning neural network using Keras packages. We imported the sequential, dense, dropout, and activation packages for defining the network architecture. ReLU is an activation function that is used to activate number of neurons at a time (1000 neurons).

An image with $256 * 256$ is filtered through convolution layers and all of them perform their specific activities. After there will be a classification done according to the images selected, whether it is categorical or binary cross entropy-based on that it selects the functions. If it contains categorical—uses SoftMax function and binary—uses Sigmoid (Fig. 4).

```

In [1]: M #importing the libraries
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.models import load_model
import numpy as np
import cv2
from skimage.transform import resize

Using TensorFlow backend.
C:\Users\mabit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
np.int16 = np.dtype(({"qint16": np.int16, "int16": np.int16}))
C:\Users\mabit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
np.uint8 = np.dtype(({"quint8": np.uint8, "uint8": np.uint8}))
C:\Users\mabit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
C:\Users\mabit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
np.int32 = np.dtype(({" qint32": np.int32, "int32": np.int32}))
C:\Users\mabit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
np.int64 = np.dtype(({" qint64": np.int64, "int64": np.int64}))
C:\Users\mabit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:521: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
np_resource = np.dtype([("resource", np.ubyte, 1)])
C:\Users\mabit\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541: FutureWarning: Passing (type,
1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint8 = np.dtype(({"qint8": np.int8, "int8": np.int8}))
C:\Users\mabit\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542: FutureWarning: Passing (type,
1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_uint8 = np.dtype(({"quint8": np.uint8, "uint8": np.uint8}))
C:\Users\mabit\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type,
1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
<ipython> In [1]: np.int16 == np.int16(1)

```

```

In [3]: M model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))

WARNING:tensorflow:From C:\Users\mabit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.place
holder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\mabit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf ran
dom_uniform is deprecated. Please use tf.random.uniform instead.


```

```

In [4]: M model.add(MaxPooling2D(pool_size=(2,2)))

WARNING:tensorflow:From C:\Users\mabit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.
max_pool is deprecated. Please use tf.nn.max_pool2d instead.


```

```

In [5]: M model.add(Flatten())


```

```

In [6]: M model.add(Dense(output_dim=128,init='uniform',activation='relu'))

C:\Users\mabit\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your 'Dense' call to the Keras 2 AP
I: 'Dense(activation="relu", units=128, kernel_initializer="uniform")'
    """Entry point for launching an IPython kernel.


```

```

In [7]: M model.add(Dense(output_dim=2,activation='softmax',init='uniform'))

C:\Users\mabit\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your 'Dense' call to the Keras 2 AP
I: 'Dense(activation="softmax", units=2, kernel_initializer="uniform")'
    """Entry point for launching an IPython kernel.


```

Fig. 4 Adding layers, libraries activation function

4 Experimental Analysis

When we give an image, it predicts accurately and tells whether the given image is the forest with fire or without fire, and we also get an alarm sound if it recognizes the image as a forest with fire. OpenCV is also used to detect the images with the help of the camera.

As a result, when we uploaded an image by choosing it, we will get the predict button then after clicking that the model will predict whether it is a forest with fire or without fire.

This is done by the UI which was developed by using HTML and Python linked with each other, and we also used OpenCV to detect the images by using the camera. We have created a user interface using a flask model. Flask is a Web application that

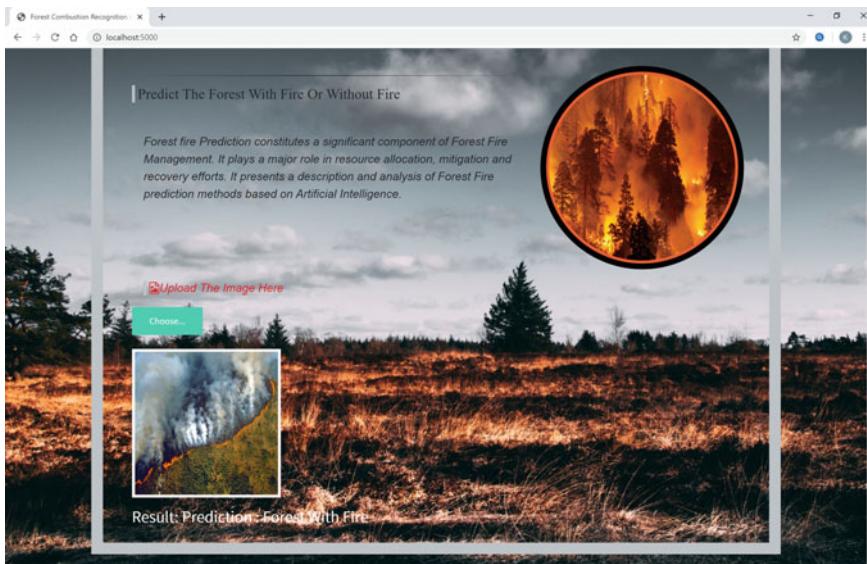


Fig. 5 Predicted output—forest with fire

we do use to inter-link the back end and front end for any of the projects. Here, in this paper, we used HTML and Python3 as our front and back ends and uses the language of Python.

4.1 Results

OpenCV. It was built to provide the usual framework to computer vision to facilitate the use of system recognition in commercial products and is considered as an ML software library (Figs. 5 and 6).

By using OpenCV, we can open the camera, and it can be able to predict whether the forest is with fire or without fire (Figs. 7 and 8).

Abbreviations and Acronyms (Table 1)

5 Conclusion and Future Scope

In this paper, a deep learning technique was used to detect the forest fires with the help of the CNN algorithm, Python3, and Flask. In this paper, we proposed a productive forest combustion technique using image processing techniques [8]. The algorithm uses the train and test model. It will train our model and further test

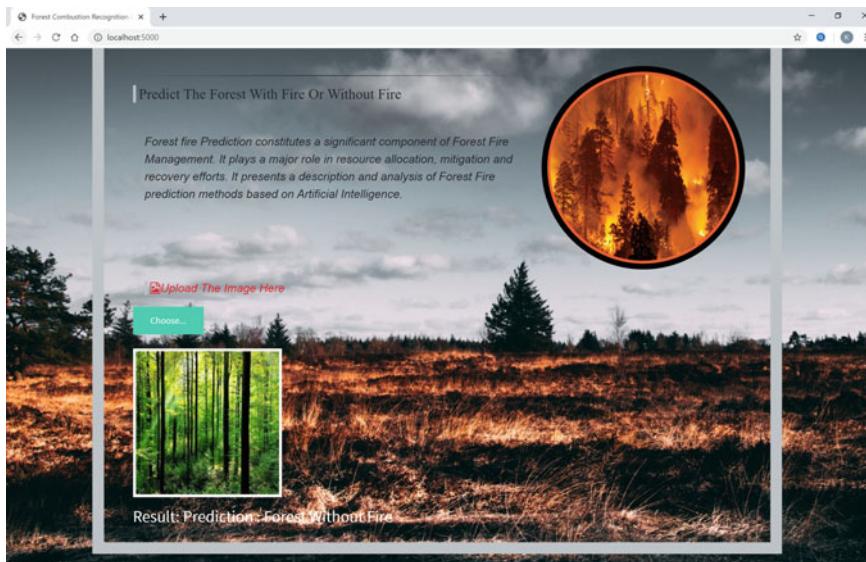


Fig. 6 Predicted output—forest without fire

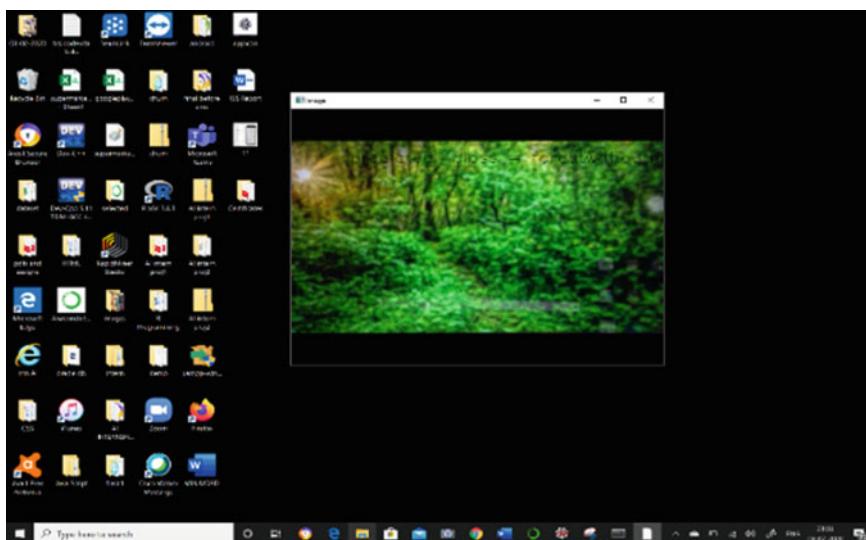


Fig. 7 Forest without fire—OpenCV

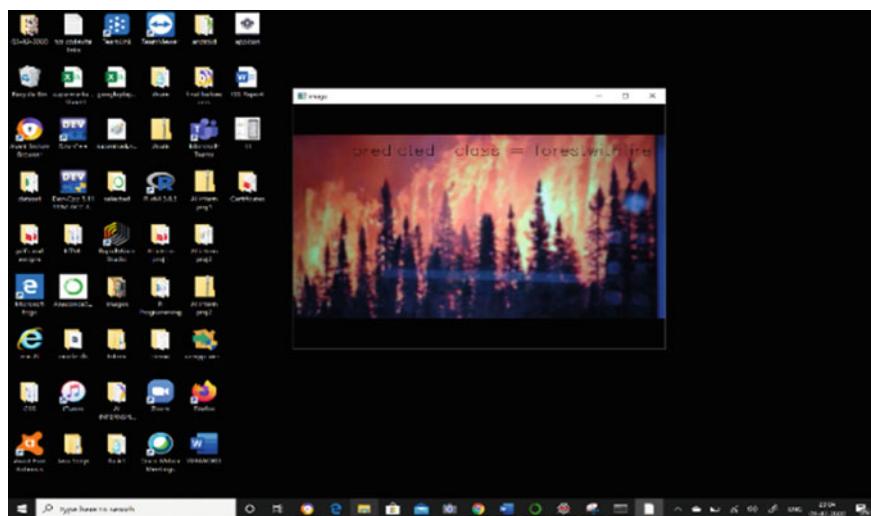


Fig. 8 Forest with fire—OpenCV

Table 1 List of words used

AI	Artificial intelligence
ML	Machine learning
HTML	Hypertext mark-up language
CNN	Convolution neural networks
ANN	Artificial neural networks
OpenCV	Open-source computer vision library
ReLU	Rectified linear units
GUI	Graphical user interface

whether the fire is present or not. The performance of the CNN algorithm is tested on a dataset containing various images of forest collected from different websites, 100 were actual fire pictures, and another 100 were without fire. These results prove that this algorithm in deep learning gives good detection and accuracy rates. So, it can be used in automatic forest fire recognition systems.

There is another self-governing method implemented like smoke detection, and it had been done based on images and few of the rule-based image processing algorithms are written by Mubarak, Honge Ren as mentioned here in the form of an article. We have trained and tested the model got the accuracy of 95%, got the alarm sound in Jupyter notebook and predicted the images with fire or without fire using OpenCV. Further, we wanted to develop our UI linked to the HTML page by getting the alarm in our system even by using sensors, cameras and make our model to run accurately within a fraction of seconds so that we can identify the combustion easily and conveniently.

References

1. Bahuguna VK, Upadhyay A (2002) Forest fires in India: policy initiatives for community participation. *Int Forestry Rev* 4:127–132
2. Sakshi I, Anil KG et al (2018) Conceptual understanding of convolutional neural network—a deep learning approach. *Proc Comput Sci* 132(8):679–688
3. Shaham N (2020) Scientists and communities in Indonesia team up for peatland restoration and fire prevention efforts. *Participatory Action Res Closing Gap Between Traditional Res Dev Community Engagement* 5
4. Ahmad A, Alkhattib A (2014) A review on forest fire detection techniques. *Int J Distrib Sens Netw Detection Techn* 10(4)
5. Satendra, Koushik AD (2014) Forest fire disaster management. National Institute of Disaster Management, Ministry of Home Affairs, New Delhi: NIDM, New Delhi
6. Matthew DZ, Rob F (2014) Visualizing and understanding convolution neural networks. New York University, Springer International, USA, Switzerland, pp 881–883
7. Sakr GE, Elhajj IH, Mitri G, Wejinya UC (2010) Artificial intelligence for forest fire prediction. In: 2010 IEEE/ASME international conference on advanced intelligent mechatronics, Montreal, ON, pp 1311–1316
8. Mubarak A, Mahmoud I, Honge R (2018) Forest fire detection using a rule-based image processing algorithm and temporal variation. *Math Probl Eng* 8

Chapter 30

The Importance of Diversity in Multi-objective Evolutionary Algorithms



Carlos A. Coello Coello and Ma. Guadalupe Castillo Tapia

1 Introduction

Since the early days of evolutionary computation, researchers in the area realized of the relevance of diversity. It is known that after running an evolutionary algorithm for a very large number of generations, it tends to converge to a single solution because of stochastic noise (computers rely on pseudo-random numbers). Because of this, diversity has become a fundamental research topic in this area.

A variety of techniques to maintain diversity are available for single-objective evolutionary algorithms. However, in evolutionary multi-objective optimization, the studies of diversity have remained relatively scarce. In multi-objective evolutionary algorithms (MOEAs), diversity is indeed of utmost importance, since it plays a key role for allowing the generation of several nondominated solutions in a single MOEA's run.

In this paper, we provide a very short overview of diversity in the context of MOEAs. The remainder of this paper is organized as follows. Section 2 provides some basic concepts related to multi-objective optimization. In Sect. 3, the importance of diversity is emphasized and we briefly discuss three main topics: (1) density estimators, (2) mating restrictions, and (3) secondary populations. Section 4 provides

The first author acknowledges support from CONACyT project no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a project from the 2018 SEP-Cinvestav Fund (application no. 4).

C. A. Coello Coello

Departamento de Computación, CINVESTAV-IPN, Mexico City, Mexico
e-mail: cacoello@cs.cinvestav.mx

Ma. G. Castillo Tapia

Departamento de Administración, UAM Azcapotzalco, Av San Pablo Xalpa 180, Col. Reynosa Tamaulipas, México City 02200, Mexico
e-mail: mgct@azc.uam.mx

a succinct summary of the most relevant advances related to diversity in the context of MOEAs, while Sect. 5 briefly describes some open research topics in this area. Finally, in Sect. 6, we provide our conclusions.

2 Basic Concepts

In multi-objective optimization, the aim is to solve the following¹:

$$\text{minimize } \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \quad (1)$$

subject to:

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, m \quad (2)$$

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, p \quad (3)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, k$ are the objective functions, and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m, j = 1, \dots, p$ are the constraint functions of the problem.

The goal when tackling these problems is to obtain the best trade-off solutions (i.e., solutions in which it is not possible to improve one objective without worsening another). Such solutions are called *nondominated* and constitute the so-called *Pareto optimal set*. The image (i.e., objective function values) of the Pareto optimal set is called *Pareto front*.

3 The Importance of Diversity

The loss of diversity (also called *genetic drift*) in evolutionary algorithms is a phenomenon that has been studied since the early origins of the field. Such studies have given rise to a variety of mechanisms that allow maintaining a diversity of solutions within the population of an evolutionary algorithm for a long time.

Holland [16] proposed an operator called *crowding* which was able to identify situations in which more and more individuals dominate a certain ecological niche. When this occurs, the competition for limited resources rapidly increases. Consequently, this produces a lower life expectancy and a lower birth rate. This was the first mechanism explicitly proposed for maintaining diversity in an evolutionary algorithm. De Jong [7] experimented with Holland's operator in his Ph.D. thesis. This was the first experimental study of a mechanism explicitly designed for maintaining diversity.

¹Without loss of generality, we only assume minimization problems.

Since then, the study of diversity has remained as an active research area in evolutionary computation [26]. Nevertheless, this topic has been scarcely investigated in evolutionary multi-objective optimization (EMOO). The main interest in EMOO has been the study of the so-called **density estimators**, which are the mechanisms responsible for avoiding convergence of the population of a MOEA to a single solution. Nevertheless, over the years, a number of alternative mechanisms have been proposed to maintain diversity of an MOEA. For example: mating restrictions and use of secondary (or external) populations.

Next, we will briefly discuss these three mechanisms before providing a short and very succinct overview on some of the most representative research on diversity that has been conducted within EMOO in the last few years.

3.1 Density Estimators

The role of a density estimator is to block the selection mechanism of a MOEA, with the aim of avoiding that its population converges to a single solution. This is a very important mechanism in EMOO, because we want an MOEA to be able to generate as many different nondominated solutions as possible in a single run. Because of their evident importance, density estimators have become a standard mechanism in modern MOEAs. The most popular density estimators are the following:

- **Niching and Fitness Sharing:** This is one of the oldest density estimators and it was originally proposed for multimodal optimization [8]. The core idea of this approach is to consider each solution in the population to be surrounded by a circle of a pre-defined radius (the niche radius or threshold). Once a circle (or niche) is defined, any other solution lying inside it, is considered to compete for resources and it is, consequently, penalized (i.e., its fitness is reduced proportionally to the number of solutions sharing the same niche). The proper definition of the niche radius is the main issue with this technique.
- **Clustering:** The idea of this density estimator is the same as when using niches, but in this case, solutions are grouped in *clusters* or groups. Uniformity is an emerging behavior that arises from the definition of a certain (normally high) number of clusters [28]. Unlike niching, in this case, there is an extensive literature available on how to generate clusters of solutions in an efficient way. The issue is, however, how to deal with outliers.
- **Adaptive Grids:** In this case, an external archive is used to store nondominated solutions which are placed using a geographically based scheme in which the aim is to distribute them in an uniform way [19]. Since this scheme relies on the definition of grids, the number of them is a user-defined parameter. Additionally, this sort of scheme cannot be applied to a large number of objectives.
- **Crowding:** In this scheme, solutions are compared in pairs based on an estimation of the density of their neighborhoods [9]. Although this scheme is very efficient

and does not require any user-defined parameters, it is not applicable to a high number of objectives.

- **Entropy:** In this case, well-known concepts from information theory are adopted to establish criteria for the uniform distribution of solutions [6].
- **Parallel Coordinates:** When using this approach, a graph is represented using a digital image in which a pixel identifies the level of linear segments that are overlapped, such that the individuals that cover a wider area of the image have a higher probability of surviving [15].

3.2 Mating Restrictions

In evolutionary algorithms, crossover is normally not restricted (i.e., any parent may recombine with any other parent). However, since the early days of evolutionary algorithms, some researchers have proposed mating restrictions mechanisms as a means to preserve diversity. Goldberg [12] mentions the use of mating restrictions in single-objective optimization problems as a mechanism to prevent or minimize the “low performance offspring” (called *lethals*). In other words, mating restrictions bias the way in which individuals mate with the aim of increasing the effectiveness and efficiency of the recombination operator.

Biologically, mating restrictions are equivalent to geographical isolation (something that is normally simulated using islands in parallel evolutionary algorithms). Geographical isolation is a key component of evolution since it creates the basic conditions to allow *speciation*. For that reason, it should not be surprising that several niching techniques incorporate mating restrictions.

Goldberg and Deb [13] suggested the use of mating restrictions with respect to phenotypic distances (i.e., in the space of the decoded solutions). The idea is to allow two individuals to recombine only if they are “very similar” (this similarity is determined using some metric). This aims to produce different *species* (mating pairs) in the population [24]. Island-based parallel evolutionary algorithms use mating restrictions, which are defined in a geographical sense, since an individual can only be recombined with another one inside its same island.

Some researchers have indicated that mating restrictions should motivate the recombination of dissimilar individuals with the aim of preventing the generation of lethals. Regardless of the mechanism adopted, several MOEAs have incorporated mating constraints with the aim of reducing dominated solutions in the population. For example, Baita [2] and Loughlin and Ranjithan [22] place solutions in a grid and constrain the area within which recombination of individuals is possible. Lis and Eiben [21] allowed recombination only between individuals having different “gender.” Jakob et al. [18] implemented a rather atypical mechanism, which was based on the values of the weights of each solution (the authors used a linear aggregating function in this case).

It is worth noting, however, that the use of mating restrictions in MOEAs is very rare nowadays [5].

3.3 Secondary Populations

Horn [17] indicates that every MOEA implementation should have a secondary (or external) population and, in general, no EMOO researcher questions their importance. However, some relevant questions related to the actual implementation of a secondary population are the following: how should the secondary population interact with the main population of a MOEA? (e.g., should the secondary population be considered during the selection process?) Should we bound its size or not? If we bound its size, how do we decide which solutions to retain?

Secondary populations are required in MOEAs because we are attempting to approximate the Pareto front of a problem with the highest possible accuracy and this requires generating as many nondominated solutions as possible. However, secondary populations can also be used to improve the distribution of solutions (i.e., they can act as a density estimator) and to preserve diversity.

The first implementations of secondary populations relied on linear lists to store nondominated solutions. However, over the years, very sophisticated data structures have been adopted to implement them. For example, Mostaghim et al. [25] proposed to use quadtrees. Bringmann et al. [3] proposed the *approximation guided evolutionary algorithm* (AGE), which adopts unbounded archives. AGE stores all the nondominated solutions generated and uses the archive to assess the quality of a population by computing its additive approximation to the true Pareto front using lexicographic minimization. Fieldsend et al. [10] indicated that bounding the size of the secondary population can produce a “shrinkage” and an “oscillation” phenomenon in the Pareto fronts produced. On the other hand, they recognize that the main problem related to unbounded secondary populations is their high computational cost. This motivated them to propose the use of two new data structures (the so-called *nondominated trees* and the *PQRS trees*) for the efficient storage and retrieval of solutions from an unbounded archive. Laumanns et al. [20] proposed a relaxed form of Pareto dominance called ϵ -dominance, which is used to filter out solutions in an external archive of a MOEA. The core idea is to define a set of boxes of size ϵ and allow only one nondominated solution within each box (e.g., the closest to the lower lefthand corner, if the objectives are being minimized). Zapotecas and Coello [23] proposed the use of the *convex hull of individual minima* (CHIM) for maintaining well-distributed solutions in the secondary population of a MOEA. Zhao and Suganthan [30] proposed an *ensemble* of ϵ values and an *ensemble* of external archives for a multi-objective particle swarm optimizer. The idea in this case is to avoid the pre-sampling that is required to estimate an appropriate value of ϵ .

4 Recent Advances on Diversity in MOEAs

Recent research on diversity in the context of MOEAs has focused on the development of new performance indicators and diversity maintenance mechanisms (mainly regarding density estimators) for problems having a large number of objectives (the so-called *many-objective optimization problems*).

Adra and Fleming [1] proposed two new density estimators (called DM1 and DM2) which were assessed on a set of many-objective optimization problems. DM1 is an adaptive strategy that regulates the diversity of the population based on the spread of solutions at each generation. DM2 consists of an adaptive mutation operator that defines the mutation range of the decision variables at each generation.

In [11], a new performance measure for assessing diversity online is proposed and is then adopted to develop a new selection criterion for an MOEA. The core idea is to measure the loss of diversity produced by a solution at two consecutive generations by using a geometric interpretation of both convergence and spread.

In [29], a new bio-inspired performance measure for assessing diversity is proposed. This approach is based on the accumulation of dissimilarity in the population and it adopts the L_p norm to measure the distance between pairs of solutions in order to estimate their dissimilarity.

In [4], a new diversity indicator based on reference vectors is proposed. This approach has a relatively low computational cost in many-objective problems. Its core idea is that each reference vector can be considered as a representative of a subregion of objective space and that the coverage of each solution can be assessed by the number of reference vectors to which it is associated. This way, diversity can be assessed by using the coverage of all the solutions.

5 Some Challenges in This Area

There are several topics related to diversity in MOEAs that are worth exploring in the next few years. For example:

1. The design of new density estimators for many-objective problems is still an open research area (see for example [29]). It would be interesting, for example, to design density estimators that can promote diversity in both decision variable and objective function space.
2. Determining what is a good distribution in a high-dimensional space is indeed an open research area. In this regard, it is worth analyzing the use of s -energy [14] and other similar diversity indicators that have been recently adopted in MOEAs.
3. The design of new schemes to assess diversity which are suitable for designing new selection mechanisms for MOEAs (e.g., promoting a faster convergence) is also a promising research area (see for example [27]).

6 Conclusions

This paper has provided a very short overview of the role and importance of diversity in multi-objective evolutionary algorithms, emphasizing the importance of doing more research in this area by providing some pointers toward some topics that are worth exploring in the near future.

References

1. Adra SF, Fleming PJ (2011) Diversity management in evolutionary many-objective optimization. *IEEE Trans Evol Comput* 15(2):183–195
2. Baita F, Mason F, Poloni C, Ukovich W (1995) Genetic algorithm with redundancies for the vehicle scheduling problem. In: Biethahn J, Nissen V (eds) Evolutionary algorithms in management applications. Springer-Verlag, Berlin, pp 341–353
3. Bringmann K, Friedrich T, Neumann F, Wagner M (2011) Approximation-guided evolutionary multi-objective Optimization. In: Proceedings of the 21st international joint conference on artificial intelligence (IJCAI 2011). AAAI Press, Barcelona, Spain (2011), pp 1198–1203
4. Cai X, Sun H, Fan Z (2018) A diversity indicator based on reference vectors for many-objective optimization. *Inf Sci* 430:467–486
5. Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, New York. ISBN 978-0-387-33254-3
6. Cui X, Li M, Fang T (2001) Study of population diversity of multiobjective evolutionary algorithm based on immune and entropy principles. In: Proceedings of the congress on evolutionary computation 2001 (CEC'2001). vol 2, . IEEE Service Center, Piscataway, New Jersey (2001), pp 1316–1321
7. De Jong AK (1975) An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan
8. Deb K, Goldberg DE (1989) An investigation of niche and species formation in genetic function optimization. In: Schaffer JD (ed) Proceedings of the third international conference on genetic algorithms. George Mason University, Morgan Kaufmann Publishers, San Mateo, California, pp 42–50
9. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
10. Fieldsend JE, Everson RM, Singh S (2003) Using unconstrained elite archives for multiobjective optimization. *IEEE Trans. Evol Comput* 7(3):305–323
11. Gee SB, Tan KC, Shim VA, Pal NR (2015) Online diversity assessment in evolutionary multi-objective optimization: a geometrical perspective. *IEEE Trans Evol Comput* 19(4):542–559
12. Goldberg DE (1989) Genetic algorithms in search. Addison-Wesley Publishing Company, Reading, Massachusetts, Optimization and Machine Learning
13. Goldberg DE, Deb K (1991) A comparison of selection schemes used in genetic algorithms. In: Rawlins GJE (ed) Foundations of genetic algorithms. Morgan Kaufmann, San Mateo, California, pp 69–93
14. Hardin DP, Saff EB (2004) Discretizing manifolds via minimum energy points. *Notices of the AMS* 51(10):1186–1194
15. Hernández Gómez R, Coello Coello CA, Alba Torres E (2016) A Multi-objective evolutionary algorithm based on parallel coordinates. In: 2016 Genetic and evolutionary computation conference (GECCO'2016). ACM Press, Denver, Colorado, USA, pp 565–572. ISBN 978-1-4503-4206-3

16. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, Michigan, USA, An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence
17. Horn J (1997) Multicriterion decision making. In: Bäck T, Fogel D, Michalewicz Z (eds) Handbook of evolutionary computation, vol 1. IOP Publishing Ltd. and Oxford University Press, pp F1.9:1–F1.9:15
18. Jakob W, Gorges-Schleuter M, Blume C (1992) Application of genetic algorithms to task planning and learning. In: Männer R, Manderick B (eds) Parallel problem solving from nature, 2nd Workshop. North-Holland Publishing Company, Amsterdam, Lecture Notes in Computer Science, pp 291–300
19. Knowles J, Corne D (2003) Properties of an adaptive archiving algorithm for storing nondominated Vectors. *IEEE Trans Evol Comput* 7(2):100–116
20. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multi-objective optimization. *Evol Comput* 10(3):263–282
21. Lis J, Eiben AE (1996) A multi-sexual genetic algorithm for multiobjective optimization. In: Fukuda T, Furuhashi T (eds) Proceedings of the 1996 international conference on evolutionary computation, IEEE, Nagoya, Japan, pp 59–64
22. Loughlin DH, Ranjithan S (1997) The Neighborhood constraint method: a genetic algorithm-based multiobjective optimization technique. In: Bäck T (ed) Proceedings of the seventh international conference on genetic algorithms. Michigan State University, Morgan Kaufmann Publishers, San Mateo, California, pp 666–673
23. Martínez SZ, Coello CAC (2010) An archive strategy based on the convex hull of individual minima for MOEAs. 2010 IEEE congress on evolutionary computation (CEC'2010). IEEE Press, Barcelona, Spain, pp 912–919
24. Mitchell M (1996) An introduction to genetic algorithms. The MIT Press, Cambridge, Massachusetts
25. Mostaghim S, Teich J, Tyagi A (2002) Comparison of data structures for storing pareto-sets in MOEAs. Congress on evolutionary computation (CEC'2002), vol 1. IEEE Service Center, Piscataway, New Jersey, pp 843–848
26. Sareni B, Krähenbühl L (1998) Fitness sharing and niching methods revisited. *IEEE Trans. Evolut Comput* 2(3):97–106
27. Toffolo A, Benini E (2003) Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evol Comput* 11(2):151–167
28. Toscano Pulido G, Coello Coello CA (2004) Using clustering techniques to improve the performance of a particle swarm optimizer. In: KD et al (ed) Genetic and evolutionary computation–GECCO 2004. Proceedings of the genetic and evolutionary computation conference. Part I. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102, Seattle, Washington, USA, pp 225–237
29. Wang H, Jin Y, Yao X (2017) Diversity assessment in many-objective optimization. *IEEE Trans Cybernet* 47(6):1510–1522
30. Zhao SZ, Suganthan PN (2010) Multi-objective evolutionary algorithm with ensemble of external Archives. *Int J Innovat Comput Inf Cont* 6(4):1713–1726

Chapter 31

Viewer’s Sentiments on Game of Thrones: An Automated Lexicon-Based Sentiment Analysis on Real-Time YouTube Comments



Shivam Sharma and Hemant Kumar Soni

1 Introduction

Human life has thrived and reached the flourishing stage of the current human scenario by the virtue of exchanging ideas, knowledge and impulse of thoughts. Prior to the introduction of language, primitive men used to communicate by sign language. Later on, an equivalent word for each emotion was defined and humans started to communicate by cloaking impulse of thoughts into its equivalent word or set of words. Humans can communicate either verbally or via text with the help of language. In the last one decade, the utilisation of social media has grown at such a massive pace that a gigantic raw data has been spawned so far. In these social media sites, a humongous volume of population is connected and people do share all sorts of emotions related to political, cultural, local and global issues. These kinds of raw data could be processed by computer with the help of natural language processing (NLP). Sentiment analysis is one of the most crucial techniques of NLP. Predictions [1, 2] and sentimental analysis have emerged as new research domain in machine learning.

YouTube is one of the substantial social media platforms where people generate the video contents and viewers send their feedback through comments. In the proposed work, sentiment analysis on the YouTube’s user comments of a “review channel” reviewing the six episodes of “Game of Thrones” (GOT) Season 8 is done. The reviewers in the YouTube video have given their live reaction on each of the six episodes. The polarity of the user comments in all the six videos is individually examined. The Web application “AutoSentilyser” has been designed to decipher the concealed sentiments by using interactive visualisation tools. This helps to examine the report of the final results with bar plots, 3D pie charts, word clouds and HTML

S. Sharma · H. K. Soni (✉)

Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Madhya Pradesh, Maharajpura, Gwalior, MP, India

Highlighter file. The sentiment analysis is performed on the real-time YouTube user comments by using the dictionary-based lexicon approach. The proposed work “AutoSentilyser” has automated the process of fetching the real-time YouTube comments in Web application and executing the sentiment analysis.

2 Related Work

The work [3] proposed the use of tidytext package in R for the sentiment analysis of Harry Potter series. Three lexicons present in the package tidytext, i.e. afinn, bing and nrc, are employed here. The paper has analysed all the three lexicons and has performed individual comparison. The authors, in this work, have stumbled across the different sentiment intensities of these lexicons and manifested their individual workings and results.

The sentimental analysis process on the Twitter data is performed in this work [4]. Performance estimation of two platforms R language and RHadoop tool is done here. R is a strong tool in data extraction and data analysis, but when the size of the data increases, the performance of R degrades. Herein, RHadoop is used to deal with big data and resolving the problems faced while using R language.

The researchers have utilised the concept of matching of words [5]. The proposed concept is used to match the word in text and vocabulary of lexica. This draws out the polarity associated with the sentiment lexica. The paper claims that the work has generated good results. This can further be extended to multilingual work [6].

This article explores the sentiment analysis on the blogs written in Urdu language [7]. Two approaches, i.e. lexicon approach and supervised machine learning techniques, are used and compared critically. The researched work claims that lexicon-based approach outperforms machine learning techniques in terms of time, accuracy and efforts.

An automated approach is designed to perform the sentiment analysis (SA) for the polarity assessment of text [8]. The work investigates the SA tools’ ability to draw the true sentiments. The researchers worked on 900 consumer reviews. Factors like review length and product type affect the SA techniques to find true sentiments.

3 Proposed Method

The proposed approach is designed in the form of a Web application named “AutoSentilyser”. “AutoSentilyser” has an interactive user-friendly interface which is easy to use. The output is also provided in the form of charts and figures. These figures are self-explanatory. In the proposed work, the authors have used the lexicon-based approach to find the sentiments and emotions associated with the user comments.

The authors have collected the user’s comments from six discrete YouTube live “reaction videos”. In “reaction videos”, the channel members give their reaction on

Table 1 Description of data set

Videos	URL	Comments
Video 1	https://www.youtube.com/watch?v=o6558Fbk_E8	1216
Video 2	https://www.youtube.com/watch?v=rskEK0IqOAM	1451
Video 3	https://www.youtube.com/watch?v=ft-GVQ1egPo	3562
Video 4	https://www.youtube.com/watch?v=Nbia70PWt4M	2414
Video 5	https://www.youtube.com/watch?v=MoZUE-pzCS8	3158
Video 6	https://www.youtube.com/watch?v=HCgOU-9gk8M	2690

some content. The name of the YouTube channel is “Blind Wave”. In these videos, the reaction panellists have given their reaction on the episodes of “Game of Thrones” Season 8. There are six episodes in the Season 8 of “Game of Thrones”. Each episode is individually reacted by the panellists in six discrete videos. Researchers have executed NLP [9] techniques for sentiment analysis on real-time user comments for each video individually. This helped the researchers in analysing the sentiments of user’s feedback associated with each episode of Season 8. In Table 1, the URL for all the six videos and the number of real-time comments on each video are detailed.

3.1 *Implementation of “AutoSentilyser”*

The implementation of the proposed automated Web application “AutoSentilyser” is done using R language. The proposed work utilises the dictionary-based lexicon approach [10], for performing the process of sentiment analysis. The “AutoSentilyser” is designed to ease the entire descriptive analytics of the sentiments on the real-time YouTube video comments. The steps involved in the implementation of “AutoSentilyser” are elucidated in this section. The following are the steps of implementation:

3.1.1 Input

The URL of the video on which the sentiment analysis has to be performed is given in the input of the “AutoSentilyser”. The YouTube API key is authenticated in the R environment. This authenticated YouTube API permits the Web application to fetch real-time comments from any YouTube video. After fetching the comments, these are stored as an object in R environment.

3.1.2 Pre-processing

The comments stored in the R environment are not suitable to perform sentiment analysis at this point. These comments are required to be pre-processed before performing sentiment analysis. The steps of pre-processing are elucidated in Fig. 1. The following are the steps involved in pre-processing:

- **Change Upper Case to Lower Case:** In this pre-processing step, all the upper-case letters are changed to the lower-case letters. For example, the word “Sentiment” is converted to “sentiment”.
- **Remove Stop Words:** Stop words like “and”, “but”, etc., do not contribute in the variation of sentiment score, so it is good to remove or filter out these as well.
- **Remove Numbers:** Numbers like 1, 2, 3 ... are of no use when the sentiment of some text is classified, so these are removed in the pre-processing step.
- **Remove Punctuation:** Punctuations are also removed from the text, as lexicon method focuses on the sentiments of individual words. Punctuation does not alter the polarity and intensity of sentiment score.
- **Stemming:** All the words are brought down to their root or stem word; for instance, “loving”, “lovely”, “loved” all are brought down to their root word and that is “love”.
- **Strip Whitespaces:** After removing the numbers, stop words and punctuations, some spaces are left in the text. These spaces are filled in this step of pre-processing.

After executing all the steps of pre-processing one by one, the comments get pre-processed. These pre-processed comments are fit to perform the sentiment analysis now.

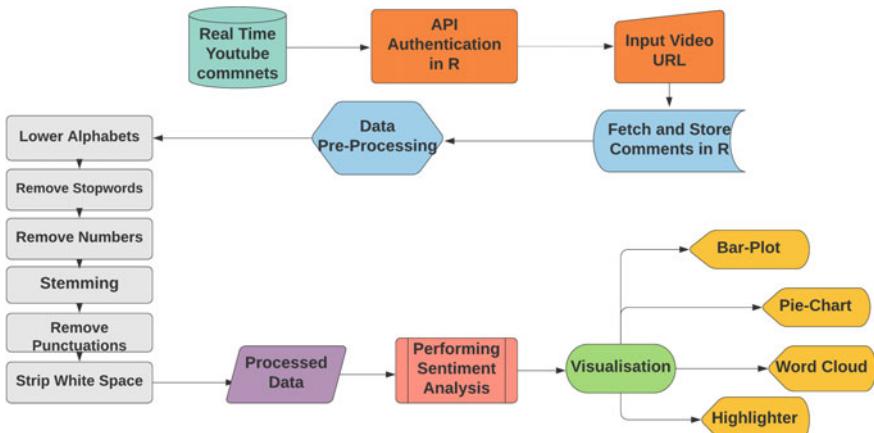


Fig. 1 Pictorial representation of proposed work

3.1.3 Performing Sentiment Analysis

In the proposed work, the sentimental analysis on the user's real-time comments [11] is performed by using a dictionary-based lexicon approach as manifested in Fig. 1. The Web application calls the NRC dictionary using `get_nrc_sentiment()` function. This NRC dictionary calculates the presence of eight human emotions, i.e. "fear", "trust", "sadness", "anger", "anticipation", "disgust", "joy" and "surprise". NRC dictionary also calculates the presence of positive and negative emotion with their corresponding valence in the text file. The classification of the individual comments as either "positive", "negative" or "neutral" is done by using `sentiment_by()` function in R. This function finds the polarity at sentence level on the basis of valence shifters [12], in the text. Valence shifters are divided into three categories. These three types of valence shifters are detailed below.

- **Amplifiers** These are the words which act as intensifiers in the sentence, for example, "extremely", "totally", "highly", "absolutely", "certainly", etc.
- **De-amplifiers** Those words which act as down toners in the sentence are known as de-amplifiers, for example, "slightly", "hardly", "a tiny bit", "barely", "almost".
- **Negators** Those words which bring the negative polarity in the sentence are known as negators, for example, "does not", "not", "cannot", etc.

3.1.4 Visualisation

This is the descriptive analytics step of the sentimental results accomplished after performing the sentiment analysis process. The proposed Web application manifests the result to the users via interactive bar plots, pie charts, word clouds, highlighted HTML file. The bar plot communicates the percentages of eight human emotions, i.e. "trust", "fear", "anticipation", "joy", "anger", "sadness", "disgust" and "surprise" in the language used in the comments. The pie chart communicates the percentages of "positive" and "negative" emotion in the text. Interactive word cloud expresses the words used most frequently throughout the comments. The highlighted HTML file shows the positive and negative comments highlighted in green and pink colours, respectively. This step of visualisation is detailed in Fig. 1.

4 Case Study on GOT Season [8] Episodes

Episode 1. "Winterfell" In the first video [13], detailed in Table 1, the reaction panellists from the channel "Blind Wave" have given their live reaction on the first episode of GOT Season 8 "Winterfell". People after watching the video gave their feedback on the live reaction of episode 1 in the comment section. The real-time 1216 user comments are fetched automatically from this video with the help of "AutoSentilyser". After performing the sentiment analysis, the results are visualised with the help of interactive bar plots, pie charts, word cloud and HTML highlighter

file. In Fig. 2, the pie chart manifests the percentage allocated to positive and negative sentiments present in 1216 user's comments. This pie chart expounds that 48.065% of 1216 comments manifest positive sentiments and remaining 51.935% manifest negative sentiment. Figure 3 manifests the word cloud for user comments. In word cloud, the words that appear with the large font size are the ones used most frequently.

Fig. 2 Positive versus negative % for episode 1

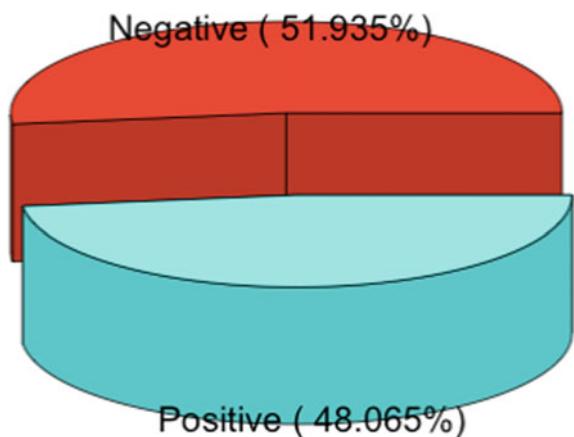


Fig. 3 Word cloud for episode 1

Words with the same colour in the word cloud express the same frequency. Here, “Jon” is the most frequent word used. It means that people in the comment’s section of this video wrote mostly about the character “Jon”.

The eight human emotion’s percentages concealed in the text, i.e. “trust”, “fear”, “anticipation”, “joy”, “anger”, “sadness”, “disgust” and “surprise”, can be visualised with the help of bar plot in Fig. 4.

The “AutoSentilyser” also generates the interactive HTML Highlighter file which can be used to visualise the positive and negative texts in the document. Figure 5 shows the highlighted HTML file for the episode 1 video user comments. Here, the positive comments are highlighted with the “green” colour and negative comments are highlighted with the “pink” colour. The texts with no highlights are treated as neutral.

Episode 2. “A Knight of the Seven Kingdom” In the second video [14] detailed in Table 1, the reaction panellists have given their live reaction on the second episode

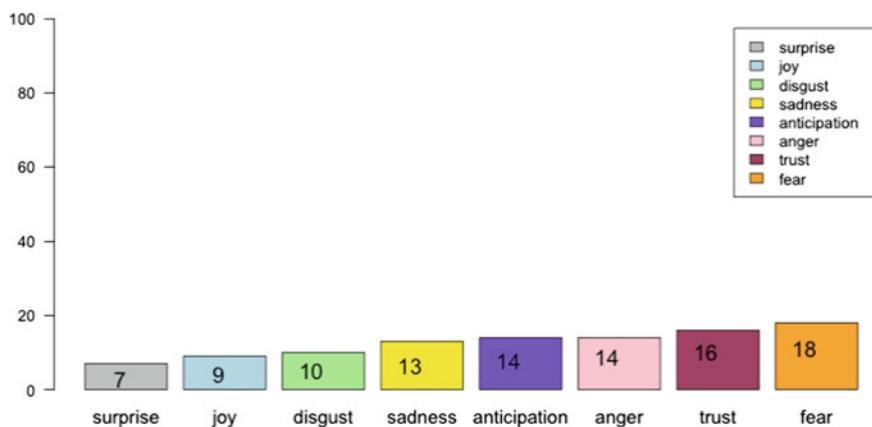


Fig. 4 Emotion percentages for episode 1

227: +.212

The only character they haven't shown is Ghost...

228: -.237

I feel you guys that Sam scene was heartbreakin lol

229: +.469

“ What is dead may never die” they all put their fists up! My favorite men!

230: -.410

Four of you watching on a tiny laptop screen makes me so sad. You miss so much detail that way 😭

Fig. 5 Highlighted HTML file for episode 1

of “GOT” Season 8. There are 1451 user comments on this video. As manifested in Fig. 6, the positive and negative percentages on episode 2 are 50.784% and 49.216%, respectively. “Night King” did not feature in episode 2. Here, “Night King” is the highly commented character as manifested in Fig. 7.

Episode 3. “The Long Night” In third video [15], detailed in Table 1, episode 3 “The Long Night” is reacted upon by the panellists. This video has 3562 user’s comments. In Fig. 8, the positive and negative percentages on episode 3 are 43.782% and 56.218%, respectively. In episode 3, “Arya” kills the “Night King” for which in Fig. 9 “Night King” and “Arya” are the most commented characters.

Episode 4. “The Last of the Starks” This is the fourth video [16] where the panellists have given their reaction upon the fourth episode of “GOT” Season 8. This video received 2414 comments on the live reaction. Positive and negative percentages

Fig. 6 Positive versus negative % for episode 2

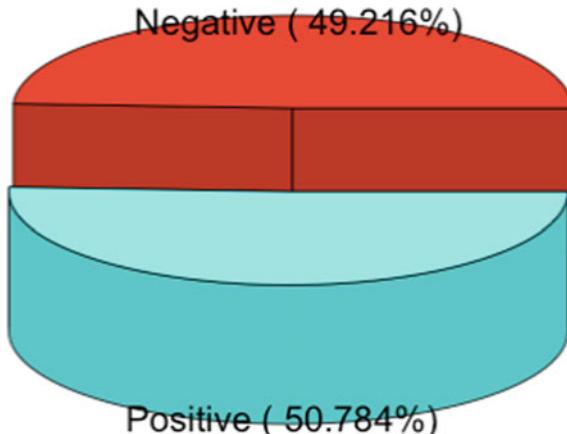


Fig. 7 Word cloud for episode 2



Fig. 8 Positive versus negative % for episode 3

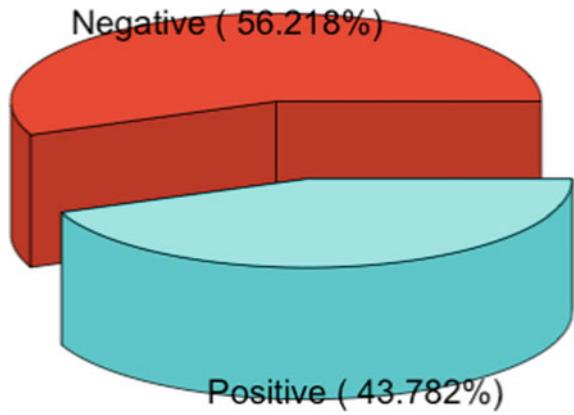


Fig. 9 Word cloud for episode 3



on episode 4 are 43.575% and 56.425%, respectively, in Fig. 10. In this episode, “Jon” reveals his true parentage to “Sansa” and as per Fig. 11 “Jon” and “Sansa” are the highly commented characters.

Episode 5. “The Bells” Fifth video [17] is reacted upon. This video received 3158 comments. In Fig. 12, the positive and negative percentages on episode 5 are 42.267% and 57.733%, respectively. In this episode, “Dany” riding on her dragon destroys the iron fleet and most of the city’s defences. Figure 13 expresses that “Dany” is the highly commented character in the comment section of episode 5.

Episode 6. “The Iron Throne” This is the final video, video 6 [18] which is the live reaction video on the finale for the Season 8 of “GOT” or the final episode’s reaction video. This video received 2690 comments. As manifested in Fig. 14, the positive

Fig. 10 Positive versus negative % for episode 4

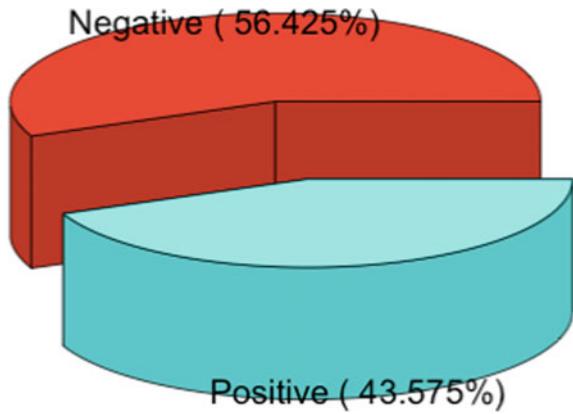


Fig. 11 Word cloud for episode 4

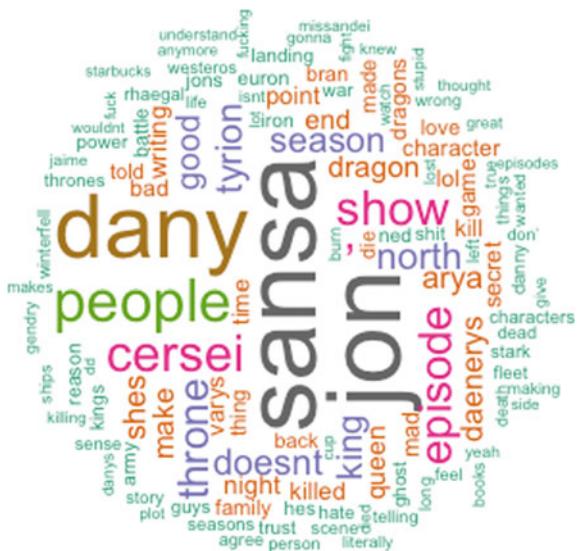


Fig. 12 Positive versus negative % for episode 5

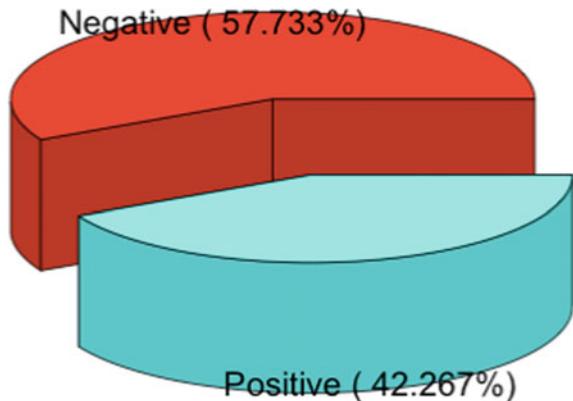


Fig. 13 Word cloud for episode 5

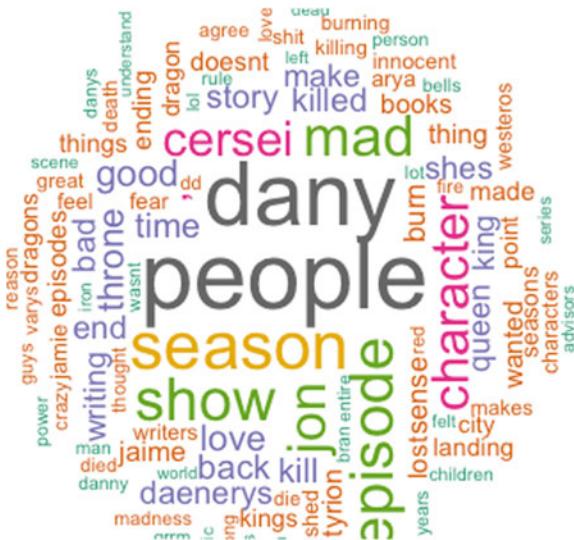
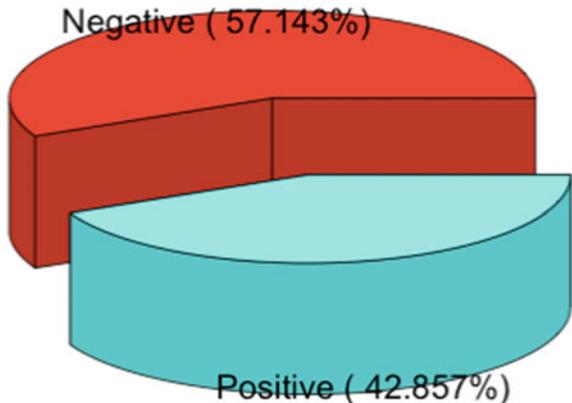


Fig. 14 Positive versus negative % for episode 6



and negative percentages on episode 14 are 42.857% and 57.143%, respectively. In the final episode of the Game of Thrones, Bran emerged out to be the surprising new king. As expressed in Fig. 15, King Bran is the most frequently discussed character in episode 6.

4.1 Critical Analysis

In this part of the paper, a comparative study of the work is done. By analysing the results of Table 2, authors have figured out that episode 2 is the most liked episode

Fig. 15 Word cloud for episode 6



Table 2 Critical comparison of the work

Episodes	Positive %	Negative %
Episode 1	48.065	51.935
Episode 2	50.784	49.216
Episode 3	43.782	56.218
Episode 4	43.575	56.425
Episode 5	42.267	57.733
Episode 6	42.857	57.143

with “50.784%” positive sentiment polarity. Here, episode 5 is the most disliked episode with highest negative polarity, i.e. “57.733%”.

The proposed lexicon-based approach works better in comparison with that of machine learning approach in terms of efforts, time and accuracy. Since the supervised machine learning method necessitates data with properly labelled outputs or response variables. When labelled data is not present to train the machine or the quantity of data is less, then the results will not be good. In addition, machine learning approach requires more computational power.

5 Conclusion

The proposed work bestows a good psychological study on the viewers of “GOT” Season 8. The sentiment analysis on each of the episodes individually helps the

authors to decipher which episodes are liked most and least among the six. The automated approach “AutoSentilyser” provides an amazing platform to perform sentiment analysis on real-time YouTube comments via interactive visualisations. The Web application uses the dictionary-based lexicon approach for sentiment classification. By analysing the sentimental results, the authors have come across that the percentage of positivity starts decreasing from episode 2. User comments on episode 2 video have received the maximum positive %, i.e. “50.784%” among the six others. User comments on episode 5 video have received the maximum negative %, i.e. “57.733%”. This communicates that users have liked the review of episode 2 most and this liking starts to decrease from episode 2. The authors after analysing the results conclude that the users liked episode 2 video and disliked episode 5 video most among the other six episodes. In each episode, a certain character is given more emphasis which is communicated by the word cloud visualisation of each episode. In future works, sentiment analysis can be performed on a large number of comments by integrating R with tools like Hadoop and Apache Spark. As the memory in R is not adequate enough to deal with gigantic data, thus, processing big data and performing sentiment analysis on big data in R require more time.

References

1. Sharma S, Soni HK (2020) An unemployment prediction rate for Indian youth through time series forecasting. In: Algorithm for Intelligent System. Springer
2. Soni HK, Sharma S (2020) Big data analytics for market prediction via consumer insight. In: Sharma S, Valiur R, Sinha GR (eds) Big data analytics in cognitive social media and literary text: theory and praxis. Springer
3. Priyavrata SN (2018) Sentiment Analysis using tidytext package in R. In: 2018 first international conference on secure cyber computing and communication (ICSCCC). IEEE, pp 577–580
4. Kumar S, Singh P, Rani S (2016) Sentimental analysis of social media using R language and Hadoop: Rhadoop. In: 2016 5th international conference on reliability, Infocom technologies and optimization (trends and future directions) (ICRITO). IEEE, pp 207–213
5. Araque O, Zhu G, Iglesias CA (2019) A semantic similarity-based perspective of affect lexicons for sentiment analysis. Knowl-Based Syst 165:346–359
6. Hogendoorn A, Heerschap B, Frasincar F, Kaymak U, de Jong F (2014) Multi-lingual support for lexicon-based sentiment analysis guided by semantics. Decis Support Syst 62:43–53. <https://doi.org/10.1016/j.dss.2014.03.004>
7. Mukhtar N, Khan MA, Chiragh N (2018) Lexicon-based approach outperforms supervised machine learning approach for Urdu sentiment analysis in multiple domains. Telemat Inform 35:2173–2183
8. Al-Natour S, Turetken O (2020) A comparative assessment of sentiment analysis and star ratings for consumer reviews. Int J Inf Manage 54:102132
9. Zeroual I, Lakhouaja A (2018) Data science in light of natural language processing: an overview. Proc Comput Sci 127:82–91
10. Nie R, Tian Z, Wang J, Chin KS (2020) Hotel selection driven by online textual reviews: applying a semantic partitioned sentiment dictionary and evidence theory. Int J Hosp Manage 88:102495. <https://doi.org/10.1016/j.ijhm.2020.102495>
11. Wyeld T, Jiranantanagorn P, Shen H, Liao K, Bednarz T (2021) Understanding the effects of real-time sentiment analysis and morale visualisation in backchannel systems: a case study. Int J Hum Comput Stud 145:102524

12. Polanyi L, Zaenen A (2006) Contextual valence shifters. In: Computing attitude and affect in text: theory and applications. Springer-Verlag, Berlin/Heidelberg, pp 1–10
13. https://www.youtube.com/watch?v=o6558Fbk_E8
14. <https://www.youtube.com/watch?v=rskEK0IqOAM>
15. <https://www.youtube.com/watch?v=ft-GVQ1egPo>
16. <https://www.youtube.com/watch?v=Nbia70PWt4M>
17. <https://www.youtube.com/watch?v=MoZUE-pzCS8>
18. <https://www.youtube.com/watch?v=HCgOU-9gk8M>

Chapter 32

Improving the Accuracy of Writer Detection of Handwritten Text Using Image Hashing



Devvrat Bhardwaj and Prateek Thakral

1 Introduction

Writer recognition has been an active research area in the past. Handwritten biometric recognition belongs to the field of behavioral biometrics, which is related to the measurement of uniquely identifiable patterns in human activities. Naturally produced handwriting is a subconscious expression of motor control, habitual actions, and thoughts. Hence, handwriting is unique to every individual. As handwriting styles can be vastly different depending on individual strokes of writing, the chances of handwritten data being misread or misinterpreted are indeed high. This research focuses on improving the accuracy of writer recognition by image and signal processing of the features (pressure, time, and co-ordinates) obtained while writing, i.e., dynamic information rather than static information, i.e., taking into consideration the sample which is actually written. The techniques used to improve accuracy were image hashing along with dynamic time warping. Images that are perceived to be similar should have similar hashes (where “similarity” is defined on the basis of hamming distance among the hashes). Benefit of image hashing is, that is, that it is extremely fast in finding duplicates and similar images as instead of searching for the whole of the image, we look for the hash value (data of fixed size) of the image [1]. Despite the difference in actual bits of their data, if the images look practically identical to a human, they hash similarly. Therefore, adjusting contrast or brightness does not affect the hash value much. Comparison of two hashes of the same length, is done on the basis of hamming distance (defined as the number of positions at which respective bits are different). Hashes with a zero hamming distance mean that the two hashes, and hence, the images are indistinguishable. Therefore, the images having least hamming distance imply similarity in the images [2].

D. Bhardwaj · P. Thakral (✉)

Department of Computer Science and Engineering, Jaypee University of Information Technology, Solan, India

There were accelerations and decelerations during the course of an observation of a given handwriting sample. These peaks and dips were unique to every writer because every individual inhibits different muscle memory and, hence, writes with unique strokes. As the data was plotted (Pressure v/s Time) in a linear sequence, dynamic time warping was used to further analyze it and find optimal match between real and test sample [3]. Application includes writer identification and verification as well as tracking a person's natural handwriting fluctuations over time. The rest of the paper is organized as follows: Sect. 2 discusses related work, methodology used for detailed implementation is provided in Sect. 3. Section 4 concludes final results along with their graphical representation. Finally, conclusion is done in Sect. 5.

2 Related Work

The work done in current field can be analyzed in this section briefly by explaining various algorithms proposed by different authors.

Bashir et al. [4] presented a Dynamic Time Warping (DTW) technique which significantly reduced data processing time and memory size of multi-dimensional time-series collected using a biometric smart pen device BiSP. The DTW algorithm was applied for time-series analysis of five different types of sensor data providing pressure, acceleration and tilt data of the pen. The results concluded that processing time and memory size could significantly be reduced without deterioration of performance in single character and word recognition. Moreover, excellent accuracy in recognition was achieved mainly due to reduced dynamic time warping RDTW technique and a novel pen device BiS.

In the paper by Tamimi et al. [5] the author provides an evaluation of the handwriting as a potential biometric identifier according to a standard framework. Firstly, they present the required conditions for a human characteristic to be considered as a biometric identifier. They also provide a generic biometric system, detailing all its various components, and the two associated modes in which it can operate: Identification and Verification. In the second part of this paper they evaluated and discussed the validity of the handwriting as biometric identifier based on the framework defined.

The paper by Buza et al. [6] presented person identification using keystroke dynamics. The author solved time-series classification problem using 1-nearest neighbor (1NN) classifier with dynamic time warping (DTW) as distance measure.

Harakannanavar et al. [7] made a comprehensive study on the existing biometric techniques along with their usage and limitations in real world scenarios. It also describes various security and technical issues related to biometric systems.

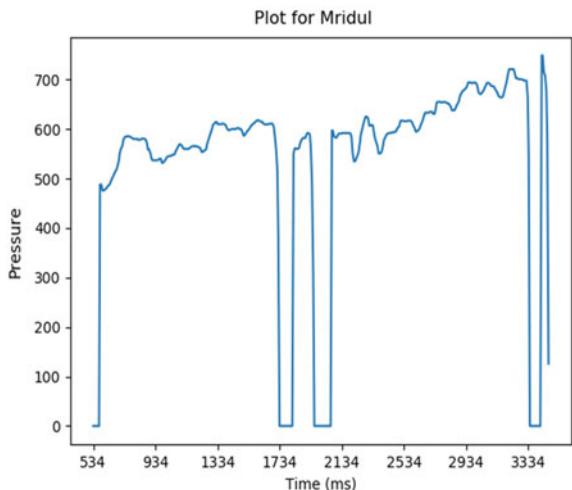
3 Methodology

Forgery is pretty much impossible as the person forging cannot imitate the same pressure–time pattern as that of the original writer. Imitating the same pressure peaks as original writer will result in careful and slow writing, hence increasing time to write the word or sentence in a flow.

The handwriting of the test subjects was captured using Wacom graphics tablet and Eye and Pen et al. [8] software which recorded pressure, time, and co-ordinates of the written sample. The pen pressure is received in non-scaled units (0–1023) rather than in force units per area. The dataset used in this experiment was collected from 20 writers (eight words, one alphabet, one sentence, and one test case out of these 10 samples). The data was exported in a.txt file, which then was preprocessed. Various kinds of noises such as unwanted pen strokes and null pressure before the first stroke were removed and the data then normalized without discarding valuable information. The collected samples were independent of the displacement and rotation of the graphics tablet as well as the movement of wrist, arm and shoulder since the co-ordinates of the samples were discarded and were not used in analysis. This normalized data was then plotted via matplotlib and the image was saved for analysis. Data processing and classification were done in Python 3.7.4 (32-bit) on a computer equipped with Intel Core i5 processor (2.4 GHz, 4 GB RAM).

Analysis

Fig. 1 Sample pressure versus time plot for the word “Serially” written by Mridul



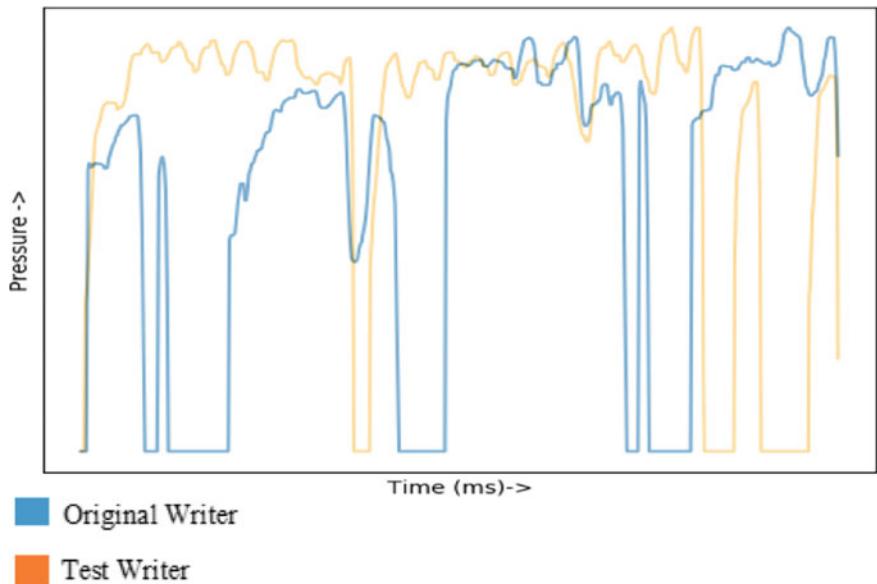


Fig. 2 Pressure–time plot of word “Impossible” by writer 8 as compared to test case of writer 17 with avg. hash difference of 20

3.1 Image Hashing

There are hash collisions if images are similar in case of image hashing. The analysis was done by plotting pressure versus time graph of the data obtained from acquisition of different words and sentences in the samples collected. The test case word was compared with every other reference graph of the same word by applying image hashing (average hashing). The hash differences were recorded against respective writers. This was done with every test case and respective words/characters/sentence.

3.2 Dynamic Time Warping

As different samples may differ in speed, dynamic time warping was used to measure the similarity between the two sequences (pressure/time plots for test sample and writer’s sample). The minimum distances between the sequences were obtained by DTW and were stored against the writer as a true measure of similarity between them (writer’s sample and test sample).

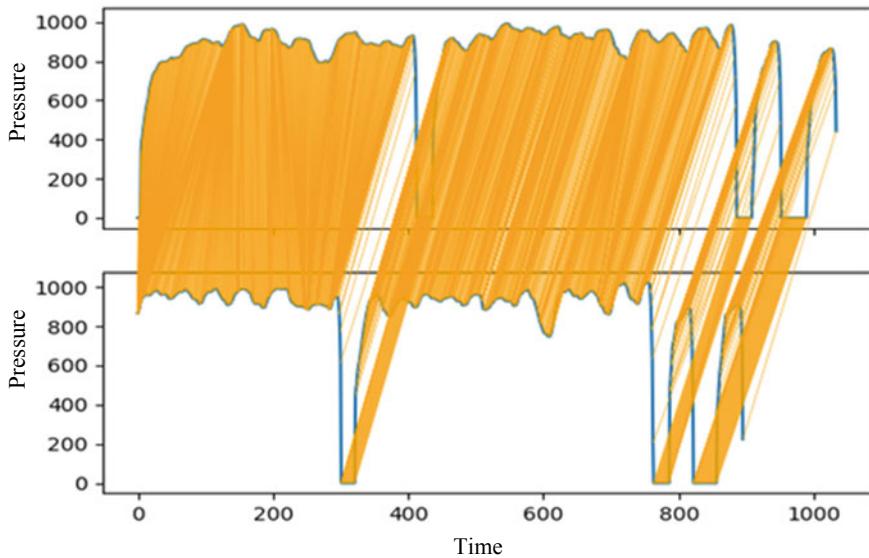


Fig. 3 DTW path between writer's sample (above) and test sample (below) for writer 17 and test case 17 with DTW distance 2593

4 Implementation and Results

For a sample case study, we have taken four authors for the word “jovial,” i.e., writer 7, 9, 10, and 14 and a test case from one of these authors.

Image hashing (average and difference) were applied on pressure versus time plot. The least average image hash difference gave a nearly accurate indication of the original writer.

DTW was applied on the pressure signals in succession to average hashing for the writers having the least avg. hash difference and the second least avg. hash difference (writers 9 and 10 in this case as shown below), against the test signal. The min DTW distance among them indicated the writer of the text.

Moreover, DTW was applied on the pressure-time signal (writer v/s test with 20 writers for 20 test cases). The minimum DTW distance between test and writer on a single test sample indicated the original writer.

After analyzing the minimum values of avg. image hash and DTW distance in this case, it can be inferred that writer 10 is the original author among writers 7, 9, 10, and 14.

After analyzing every test case with the respective samples, the following results were obtained.

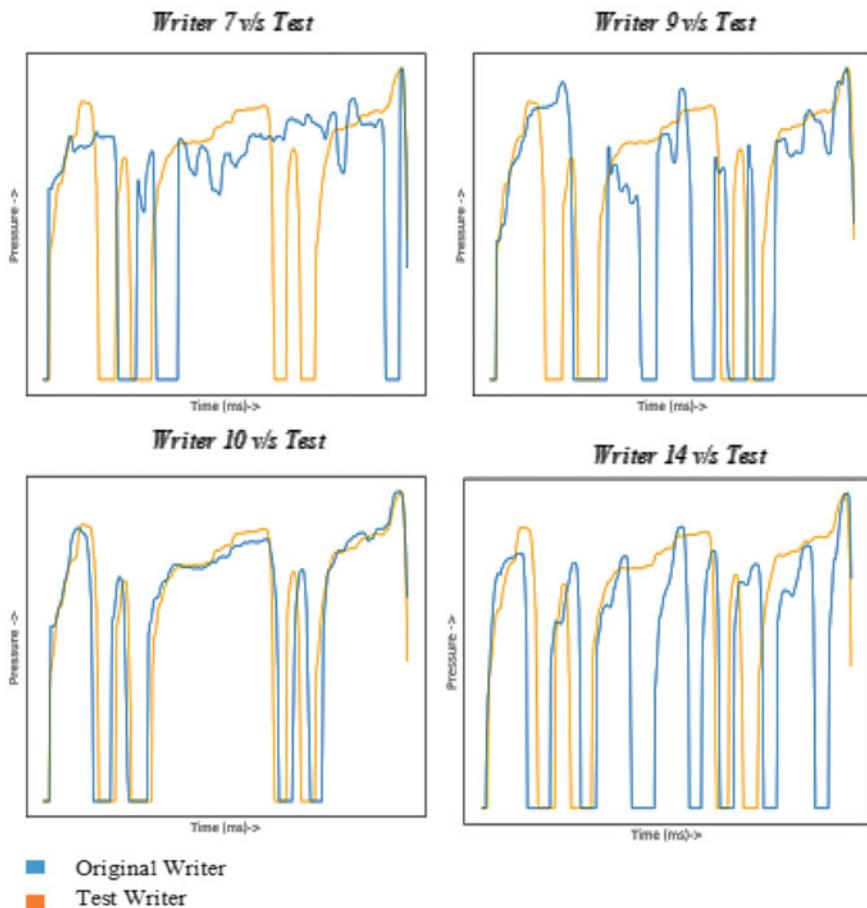


Fig. 4 Pressure–time plots of word “jovial” written by writers 7, 9, 10, and 14, which are compared with the test case, respectively. After applying average image hashing, the results are stored in Table 1

Table 1 Average image hash and DTW distance values for sample cases compared against test case of writer “10”

Case	Average image hash	DTW distance
7 versus test	22	3195.47
9 versus test	7	3520.96
10 versus test	4	1691.25
14 versus test	15	4582.37

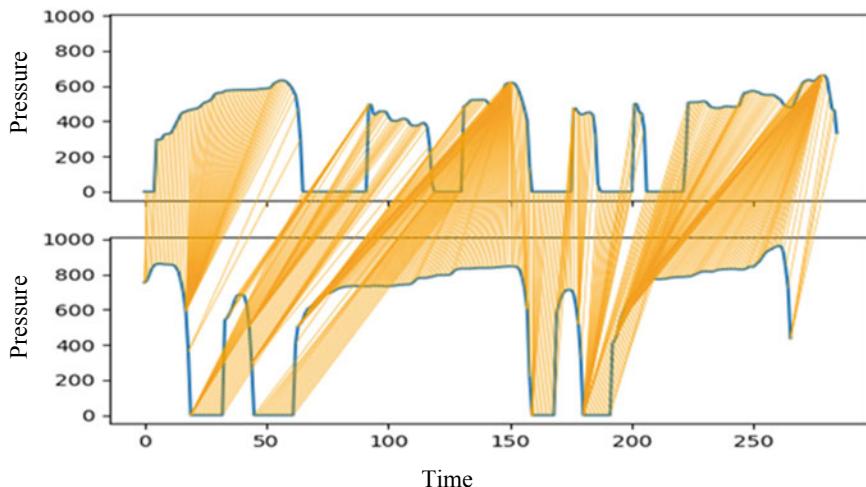


Fig. 5 Dynamic time warping: writer 9 (above) against test case (below) with DTW distance 3520.96

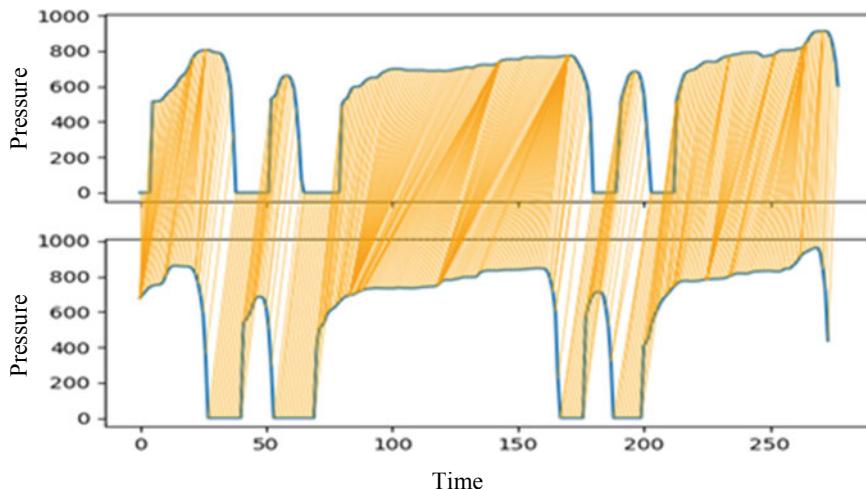


Fig. 6 Dynamic time warping: writer 10 (above) against test case (below) with DTW distance 1691.25

Table 2 Accuracy of different techniques in identification of the original writer

Technique used	Identification accuracy (%)
Dynamic time warping only	65
Difference hash difference only	65
Average hash difference only	80
Dynamic time warping after average hash difference	90

5 Conclusion

We have presented a new algorithm for improving accuracy to detect writer of a text sample. In this paper, we have investigated the feasibility of this method in 8 words, 1 sentence, and 1 alphabet. The method proposed in this paper has certain advantages which are discussed as follows. Firstly, the method functions on the pressure-time plot obtained after preprocessing. Therefore, size and position variations in the actual text samples are not of concern.

Secondly, this method needs no complex computation and is pretty fast and exhibits high accuracy. It may also be easily applicable in practical applications. Thus, it is a promising technique for biometric personal identification. The accuracy can further be enhanced by using high-grade hardware that can measure parameters such as angle of writing (i.e., pen tilt), velocity of the pen and grip pressure, which can be used in combination with the techniques analyzed in this paper.

References

1. Namboodiri A, Gupta S (2006) Text independent writer identification from online handwriting. In: Tenth international workshop on frontiers in handwriting recognition, Université de Rennes 1, La Baule France
2. Pugnaloni M, Federeconi R (2013) Forensic handwriting analysis: a research by means of digital biometrical signature. In: Criminalistics and forensic examination: science, studies, practice. Vilnius, Lithuania, 21–22 June 2013
3. Patil B, Patil P (2018) An efficient DTW algorithm for online signature verification, pp 1–5. <https://doi.org/10.1109/ICACCT.2018.8529614>
4. Bashir M, Kempf J (2008) Reduced dynamic time warping for handwriting recognition based on multidimensional time series of a novel pen device. World Acad Sci Eng Technol Int J Electr Comput Eng 2
5. Bensefia A, Tamimi H (2018) Validity of handwriting in biometric systems. In: PRAI 2018: proceedings of the international conference on pattern recognition and artificial intelligence
6. Buza K (2016) Person identification based on keystroke dynamics: demo and open challenge. CAiSE Forum
7. Harakannanavar SS, Renukamurthy PC, Raja KB (2019) Comprehensive study of biometric authentication systems, challenges and future trends. Int J Adv Netw Appl
8. Alamargot D, Chesnet D, Dansac C, Ros C (2006) Eye and Pen: a new device to study reading during writing. Behav Res Methods Instr Comput 38(2):287–299

Chapter 33

Software Defect Prediction by Strong Machine Learning Classifier



Meetesh Nevendra and Pradeep Singh

1 Introduction

In a software development life cycle, software defect prediction (SDP) plays a significant role in reducing cost and enhancing the software system's quality. The prediction of earlier defect-prone modules will reduce the allocated time and man-power resources. However, in the past few decades, researchers efficiently utilized machine learning (ML) algorithms to identify the defect-prone modules. The ensemble learning algorithm is one of the branches in ML. It works on the principle that it utilized several learning algorithms to solve the problem. It shows the enhanced generalizability than that of a single learner; thus, ensemble algorithms are very useful. The adaptive boosting algorithm (AdaBoost) is one learning technique that works on an ensemble learning algorithm's principle. The AdaBoost algorithm has been introduced by Freund and Schapire [1] in 1997.

AdaBoost is very popular because of its simplicity and effectiveness. The AdaBoost algorithm has been used for diabetic analysis, gender identification, dictionary learning, traffic control, face recognition, image classification, bug count prediction, face detection, fraud detection and text detection. AdaBoost is mostly utilized to improve the predictive power of other learning algorithms. It generates a weak or base classifier in every iteration of running an algorithm by calling the base learner, which helps in reducing the generalization error and enhancing the prediction performance. Usually, the AdaBoost algorithm is used as a decision tree as its base learning. However, the algorithms suffer in achieving the desired result when the data set is imbalanced, and the base learner cannot achieve 50% accuracy. To overcome this

M. Nevendra (✉) · P. Singh

Department of Computer Science and Engineering, National Institute of Technology, Raipur, India
e-mail: mnevendra.phd2018.cs@nitrr.ac.in

P. Singh
e-mail: psingh.cs@nitrr.ac.in

problem, we introduce a strong machine learning classifier called AdaBoost.ET, which incorporates both bagging and boosting to improve the AdaBoost's prediction performance and reduce the generalization error. In our proposed approach, AdaBoost.ET, we utilized an extra tree algorithm as a base learner of AdaBoost. The extra tree algorithm has been introduced by Pierre et al. [2] in 2006. It is a tree-based ensemble approach that randomly includes attribute and cut-point choice when splitting a tree node, reducing variance and increasing accuracy. The proposed approach has experimented publicly on open-source defect data collected from the tera-PROMISE repository [3]. We evaluated the robustness of our proposed approach with different variants of AdaBoost algorithms, and the outcome shows the significant performance of AdaBoost.ET. The remaining work is planned as follows: Section 2 provides the research of previous work; Sect. 3 presents the detailed analysis of data sets and performance measure; Sect. 4 provides the empirical research of our proposed approach; Sect. 5 analyses the outcomes of the proposed approach, and lastly, conclusions of the research are conversed in Sect. 6.

2 Related Work

SDP can be originated as a binary classification problem, where each module has been classified as defective or not. Many ML techniques have been utilized for building the SDP techniques [4–7]. Some of the methods which utilized for SDP are Naïve Bayes [8], SVM [9], LSTM [10], K-means [11], neural network [5], CNN [12], Ridge [4] and Lasso [4]. Malhotra [13] systematically reviews the past 24 years of research that uses machine learning techniques for SDP and compares the different ML techniques for SDP. Wang et al. [14] has also conducted a comparative study of several ML algorithms for SDP and found that the ensemble classifier achieves superior performance than a single classifier; they also discovered that the voting algorithms enhance performance defect prediction. AdaBoost is a modest algorithm that generates a strong classifier by a combination of weak or base classifiers. It established one of the top ten ML algorithms in data mining. In the experiment conducted by Settoufi et al. [15] using the top 10 ML algorithms, they found that the AdaBoost algorithm achieved the third rank. In several different classification domains, AdaBoost has been utilized such as detecting and reading text [16], diabetic analysis [17], vessel segmentation [18], gender identification [19], dictionary learning [20], traffic control [21], face recognition [22], image classification [23], bug count prediction [24], face detection [7], fraud detection and text detection.

Malhotra [13] performed an empirical study on different ML approaches and found that in SDP, ensemble learning algorithms are not extensively used (i.e. 18.47%). She also found that only 17% of the researchers used the AdaBoost algorithm. The evidence illustrates that there is an insufficiency of investigation in the field of SDP using AdaBoost. To overcome this research gap, we proposed our approach and given a direction for the researchers. AdaBoost is also well suited to handle the imbalanced data and generates a significant classification result [25]. Xuan et al.

[26] utilized the AdaBoost algorithms to reduce the actual data size. However, Seiffert et al. [27] employed the AdaBoost algorithm to enhance the software system's quality. They also disclose that the boosting method performs significantly better than data sampling methods. Wang and Yao [28] also used the AdaBoost algorithm for SDP; they made some changes in modern AdaBoost algorithms, which significantly enhanced the model.

3 Proposed Approach

To enhance the prediction performance of the AdaBoost algorithm without using balancing techniques, we proposed a strong machine learning algorithm called AdaBoost.ET for SDP. We employed the extra tree (ET) algorithm as a base learner of the AdaBoost. As shown in Fig. 1, we separate the original imbalanced data set into k parts (i.e. $k = 10$) where AdaBoost.ET is trained using $k-1$ parts of the software defect data set. In contrast, the learned model is evaluated in the remaining part of the data set. In Algorithm 1, the pseudo-code of our proposed approach is shown. In the algorithm, we employed extra tree as a base learner of the AdaBoost algorithm. As our data set is imbalanced, the algorithm's prediction tends towards the majority of the class, which leads to a bias problem. The extra tree algorithm overcomes the problem of the algorithm's bias nature, enhancing the generalizability and prediction performed without using the balancing techniques.

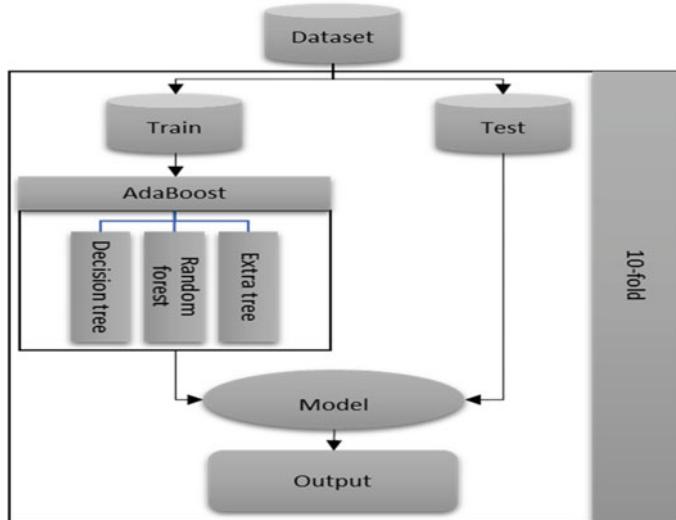


Fig. 1 Proposed model framework

The outcomes are validated using k -floods (where $k = 10$); we execute every operation ten times. At every iteration, we emphasize reducing generalize error and enhance the prediction performance. We plot the ROC curve for each iteration, which is shown in Fig. 2, and also, we recorded the average accuracy. In Algorithm 1, $p_1, q_1, \dots, p_n, q_n$ are the training label from some field P , and the labels are $q_i \in \{0, 1\}$.

In every iteration $t = 1, \dots, R$, a distribution D_t is computed. Here the base learner is utilized to find a weak proposition $h_t: P \rightarrow \{0, 1\}$, with low weighted error ε_t . The final proposition H calculates the sign of a weighted combination of a weak proposition.

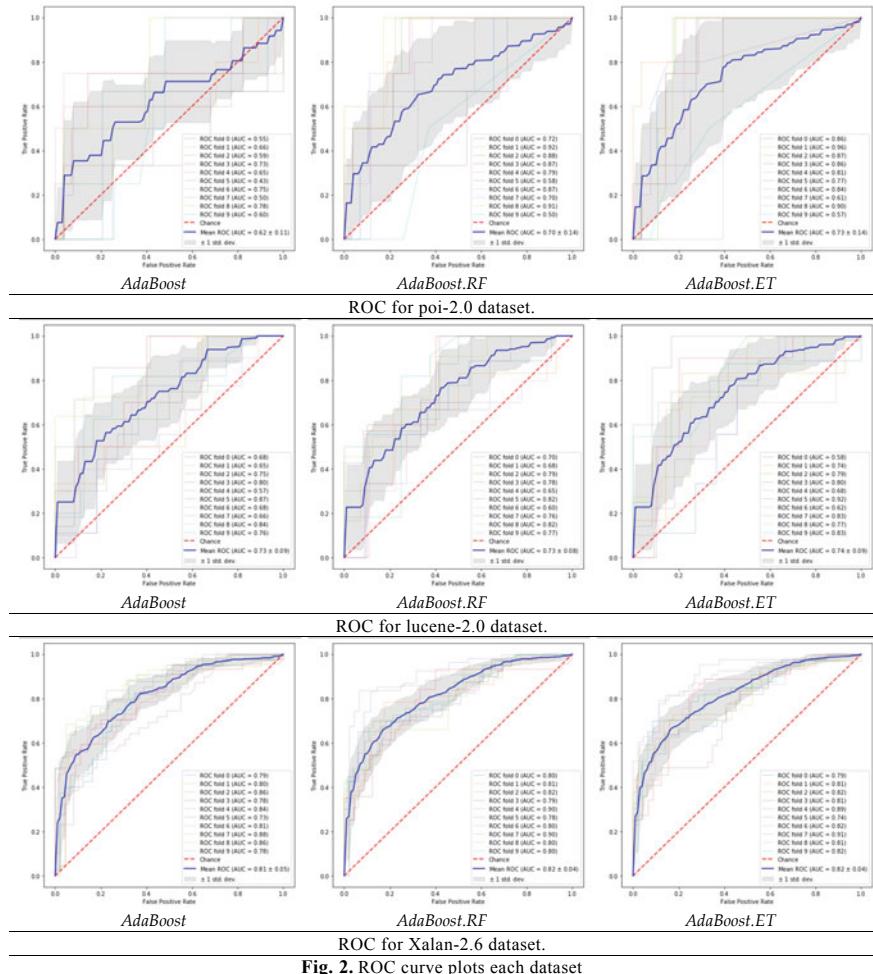


Fig. 2 ROC curve plots each data set

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(p) \right) \quad (1)$$

where H is calculated as a weighted majority vote of the weak proposition.

Algorithm 1 The boosting algorithm AdaBoost.ET

Given: $(p_1, q_1), \dots, (p_n, q_n)$ where $p_i \in P, q_i \in \{0, 1\}$.
 Initialize: Weigh vector: $W_1(n) = \frac{1}{n}$ for $n = 1, 2, \dots, n$
 For $t=1, \dots, R$:
 Train Extra-Tree as a using distribution D_t .
 Get a weak hypothesis $h_t: P \rightarrow \{0, 1\}$.
 Aim: select h_t with low weighted error:
 $\varepsilon_t = \sum_{n:h_t(p_n) \neq q_n} W_t(n)$
 Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right)$
 Update, for $j = 1, \dots, r$:
 $W_{t+1}(j) = \frac{W_t(j) \exp(-\alpha_t q_j h_t(p_j))}{Z_t} Z_t$
 Where Z_t is a normalization factor
 Output the final hypothesis:
 $H(x) = \text{sign} (\sum_{t=1}^T \alpha_t h_t(p))$

We demonstrate the capability of our approach to open-source software defect data set originally collected by Jureczko and Madeyski [30]. We also compare our proposed approach with different variants of AdaBoost algorithms. The outcomes are demonstrated using accuracy and ROC curve.

4 Data sets and Performance Measure

4.1 Data set

For the experiment of our proposed approach, we utilized the historic defect data set, which is collected from the tera-PROMISE data repository [5]. The data set is a collection of features and classes. We utilized the imbalanced data set for the investigation. Table 1 displays the description of data sets. The columns represent the

Table 1 Statistics of data sets

Data set	LOC	Number of instances	Defects instances	Percentage of defects (%)
Lucene-2.0	50,596	295	91	30.80
Poi-2.0	93,171	314	37	11.80
Xalan-2.6	411,737	985	411	41.70
Total	555,504	1594	539	33.80

line of code (LOC), total number of instances, total number of defective instances and the total percentage of the defective instance. As the table shows, the data sets' defective percentages are 30.8, 11.8 and 41.7%, which are imbalanced.

4.2 Performance Measure

To estimate our approach, we utilized accuracy and receiver operating characteristics (ROC) curve.

4.2.1 Accuracy

Accuracy is a highly used assessment measure for classification execution; it defined the fraction between the actual classified samples to the overall number of a sample as follows [29]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

where T and F indicate true and false, and P and N specify the amount of positive and negative samples, correspondingly.

4.2.2 Receiver Operating Characteristics (ROC)

The receiver operating characteristics (ROC) curve is a graph that represents the true positive rate (TPR) on the y-axis and the false positive rate (FPR) on the x-axis.

5 Result

To measure our anticipated approach's usefulness, we utilized the ROC curve and accuracy because they are the most informative measure for classification algorithms. The ROC curve is computed for k -times (i.e. $k = 10$) and estimated the k run's mean. Figure 2 represents the ROC curve of the experimental outcomes, where AdaBoost.ET obtained a mean 0.73 ± 0.14 ROC with the poi-2.0 data set, 0.74 ± 0.09 with the Lucene-2.0 data set and, 0.82 ± 0.4 with the Xalan data set, which is the highest ROC among all presented model. Also, the accuracy is computed for each run, and the average of k -fold is shown in Table 2. The obtained accuracy shows that AdaBoost.ET achieved the highest accuracy in two out of three data sets, which are 0.876 ± 0.050 in the Poi-2.0 data set and 0.776 ± 0.049 in the Xalan-2.6 data

Table 2 Accuracy of AdaBoost.ET compared with AdaBoost and AdaBoost.RF

Data set	Accuracy		
	AdaBoost	AdaBoost.RF	AdaBoost.ET
Lucene-2.0	0.637 ± 0.077	0.708 ± 0.075	0.693 ± 0.075
Poi-2.0	0.869 ± 0.033	0.860 ± 0.062	0.876 ± 0.050
Xalan-2.6	0.732 ± 0.038	0.743 ± 0.044	0.776 ± 0.049

set. From the outcomes, we can accomplish that the anticipated algorithm achieves better performance in both the low percentage of defect data and a high percentage of defect data.

AdaBoost.ET produced significant results compared to modern AdaBoost and AdaBoost.RF. The outcomes show that the proposed approach achieves the highest ROC and accuracy regarding SDP, whereas using the random forest as a weak learner of Adaboost (Adaboost.RF) performs well, nevertheless, the extra tree accomplishes more significantly with the imbalanced data set. AdaBoost.ET shows better prediction performance to obtained software defects than the AdaBoost.RF. The conclusions give clear guidance that AdaBoost.ET performs significantly for SDP with an imbalanced data set.

6 Conclusion

To address the problem of the AdaBoost algorithm while using the imbalanced data set for SDP, we proposed a strong machine learning classifier named AdaBoost.ET, where we employed extra tree as a base learner of AdaBoost to enhance and incorporate both bagging and boosting to improve AdaBoost's prediction performance. We compared our proposed approach with AdaBoost and AdaBoost.RF. Experimental results validated that the AdaBoost.ET approach surpasses the other variants of the AdaBoost approach. Our proposed model obtained a mean of 0.73 ± 0.14 ROC with the Poi-2.0 data set, 0.74 ± 0.09 with the Lucene-2.0 data set and, 0.82 ± 0.4 with the Xalan data set, which is the highest ROC compared benchmark models. The accuracy of the proposed strong machine learning classifier is achieved due to the bagging-based approach at the base, which reduces variance at each step while boosting decreases the bias.

References

- Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139
- Geurts P et al (2006) Extremely randomized trees. *Mach Learn* 63(1):3–42
- tera-PROMISE: Welcome to one of the largest repositories of SE research data. no date

4. Yang X, Wen W (2018) Ridge and lasso regression models for cross-version defect prediction. *IEEE Trans Reliab* 67(3):885–896
5. Arar ÖF, Ayan K (2015) Software defect prediction using cost-sensitive neural network. *Appl Soft Comput* 33:263–277
6. Okutan A, Yıldız OT (2014) Software defect prediction using Bayesian networks. *Empir Softw Eng* 19(1):154–181
7. Huda S et al (2018) An ensemble oversampling model for class imbalance problem in software defect prediction. *IEEE Access* 6:24184–24195
8. Arar ÖF, Ayan K (2017) A feature dependent Naive Bayes approach and its application to the software defect prediction problem. *Appl Soft Comput* 59:197–209
9. Wei H et al (2019) Establishing a software defect prediction model via effective dimension reduction. *Inf Sci (Ny)* 477:399–409
10. Liang H, Yu Y, Jiang L, Xie Z (2019) Seml: a semantic LSTM model for software defect prediction. *IEEE Access* 7:83812–83824
11. Öztürk MM et al (2015) A novel defect prediction method for web pages using k-means++. *Expert Syst Appl* 42(19):6496–6506
12. Viet Phan A et al (2017) Convolutional neural networks over control flow graphs for software defect prediction. In: 2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI). IEEE, pp 45–52
13. Malhotra R (2015) A systematic review of machine learning techniques for software fault prediction. *Appl Soft Comput J* 27:504–518
14. Wang T et al (2011) Software defect prediction based on classifiers ensemble. *J Inf Comput Sci* 8(16):4241–4254
15. Settouti N et al (2016) Statistical comparisons of the top 10 algorithms in data mining for classification task. *Int J Interact Multimed Artif Intell* 4(1):46
16. Chen X, Yuille AL (2004) Detecting and reading text in natural scenes. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit* 2
17. Roychowdhury S et al (2013) DREAM: diabetic retinopathy analysis using machine learning. *Biomed Heal Inform IEEE J* 99:1
18. Lupaşcu CA et al (2010) FABC: retinal vessel segmentation using AdaBoost. *IEEE Trans Inf Technol Biomed* 14(5):1267–1274
19. Hao Y et al (2017) Gender identification based on the fusion of Adaboost and SVM 1:2201–2206
20. Barstügan M, Ceylan R (2018) The effect of dictionary learning on weight update of AdaBoost and ECG classification. *J King Saud Univ Comput Inf Sci*
21. Leshem G, Ritov Y (2007) Traffic flow prediction using adaboost algorithm with random forests as a weak learner. *Proc Int Conf* 1(1):193–198
22. Yao M, Zhu C (2017) SVM and Adaboost-based classifiers with fast PCA for face reocognition. In: 2016 IEEE international conference on consumer electronics ICCE-China 2016, pp 0–4
23. Nayak DR et al (2016) Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests. *Neurocomputing* 177(December):188–197
24. Nevendra M, Singh P (2019) Software bug count prediction via AdaBoost.R-ET. In: 2019 IEEE 9th international conference on advanced computing (IACC). IEEE, pp 7–12
25. Han J et al (2012) Data mining: concepts and techniques
26. Xuan J et al (2015) Towards effective bug triage with software data reduction techniques. *IEEE Trans Knowl Data Eng* 27(1):264–280
27. Seiffert C et al (2009) Improving software-quality predictions with data sampling and boosting. *IEEE Trans Syst Man Cybern Part-ASystems Humans* 39(6):1283–1294
28. Wang S, Yao X (2013) Using class imbalance learning for software defect prediction. *IEEE Trans Reliab* 62(2):434–443

29. Sokolova M et al (2006) Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. Springer, pp 1015–1021
30. Jureczko M, Madeyski L (2010) Towards identifying software project clusters with regard to defect prediction. In: Proceedings of 6th international conference on prediction modelling software engineering—PROMISE'10, p 1

Chapter 34

FECG Extraction Using 1D Convolution Neural Network



Yojana Sharma, Shashwati Ray, and Om Prakash Yadav

1 Introduction

The heart of the fetus and its growth is considered in the early stages of pregnancy. Fetal electrocardiogram (FECG) is a graphical representation of the electrical activity of the heart. FECG is utilized as a routine process to obtain vital information about the fetal like heart rate, waveform, fetal development, fetal maturity and congenital heart disease [1, 2].

FECG can be broadly measured in two different manners: direct(invasive) method and indirect(non-invasive) method. In the invasive method, the electrode should move through the mother's abdomen and reach the womb to meet the head of the fetus. Although the method provides pure FECG signal, this may create problems to mother as well as fetus. Popular direct methods include scalp electrode (spiral, clip or suction) method [2]. Indirect way (non-invasive) of FECG measurement is the extraction of FECG recorded on the mother's abdomen. Abdomen signal includes MECG (Mother ECG), FECG and noise signal. The MECG signal is 5–20 times bigger in amplitude and always superimposes FECG. Basically, noise signal constitutes muscular noise, electrodes noise, baselines noise and recording system noise [2].

These noises degrade the quality of FECG and thus prevent correct analysis and diagnosis. Also, the extraction of FECG from the contaminated signal is required to get the reliable and accurate information about the status of the fetal and to detect abnormalities present that will further help for confirming fetal well-being, to determine twin pregnancies and also to check fetus is alive or dead.

Machine learning (ML) models are accessible in the bio-medical field to identify patterns and trends. The most significant advantage of ML algorithms is their ability

Y. Sharma (✉) · S. Ray
Bhilai Institute of Technology, Durg, CG, India

O. P. Yadav
PES Institute of Technology and Management, Shivamogga, Karnataka, India

to improve efficiency and accuracy over time. Deep learning (DL) is a subset of ML which has multiple hidden layers and level, and each layer interprets data differently. Thus, a DL model can identify patterns and correlations between data automatically without being explicitly programmed. Because of hidden layers and levels, these models can also deal with voluminous data. These two advantages of DL make them a better choice for bio-medical signals over traditional ML models [3].

In this paper, we propose a 1D convolutional neural network (CNN) DL algorithm to extract FECG signals and enhance the quality of FECG signals of the MIT-BIH database. CNN's are fully connected networks, i.e., each neuron in one layer is connected to all neurons in the next layer, which makes these networks prone to overfitting data [3].

Rest of the article is arranged as: In Sect. 2, we provide recent existing cognitive methods to reduce noise from FECG signals. In Sect. 3, the proposed methods and materials are presented. Results and discussion are produced in Sect. 4. Finally, the conclusions regarding the proposed method are stated in Sect. 5.

2 Existing Noise Reduction Techniques Using Machine Learning

In [4], adaptive filters are presented for FECG extraction. These filters utilize maternal reference signals. In [5], a group of state-space equations is used for modeling the temporal dynamics of FECG signals, to design a Bayesian filter for ECG denoising from a single channel mixture of MECG and FECG. However, as mentioned in [5], the filter fails to discriminate when there is overlapping between FECG and MECG.

A functional link artificial neural network (FLANN) model proposed by Dey et al. [6] has been utilized to reduce the Gaussian and baseline wander noise.

The learning capability of ANN is further improved by increasing the depth of the network, i.e., the addition of several hidden layers. This has given the concept of DL that can handle intricate patterns and objects in massive data sets. CNN models have multiple layers and many parameters which are utilized to extract optimized features present in patterns/objects. However, these CNNs cannot be directly applied to the 1D scarce signal. Thus, 2D CNN models are modified to 1D CNN models and are applied to 1D biomedical signal for classification, diagnosis, health monitoring, anomaly detection, etc. [1].

The 1D CNNs were used for ECG beat classification and arrhythmia detection using log spectrogram by transforming each ECG beat to a 2D image [1]. Even 1D CNN was directly applied on ECG data. Other applications of 1D CNN include arrhythmia detection in ECG, structural health monitoring, anomaly detection in ECG signals [1].

Lei et al. [7] presented a DL feature representation for ECG identification. In this approach, a deep fusion features are extracted through CNN. Then, the neural network and SVM are used to categorize the ECG signals.

3 Methods and Materials

3.1 Data Collection and Organization

The fetal ECG synthetic database (FECGSYNDB) has been utilized for the proposed model. The database is an online freely available repository. It consists of sizeable simulated adult and non-invasive FECG (NI-FECG) signals. The individual data contains fetal–maternal noise mixtures which can be segregated as a unique source and thus provide separate waveform files for each signal source [8]. Additionally, the noises added to FECG signals are 0, 3, 6, 9 and 12 dB to make the database robust for research. In this article, 87 different FECG, MECG and noises (at different) levels have been used. Individual signals considered are of 10 seconds, sampled at 250 Hz (sample size 2500) with 16-bit resolution.

The target for the proposed model is the respective subject FECG signal at 0 dB. Data for the model is organized as: Initially, FECG signal of a subject at 0 dB is considered. For the same subject, FECG signal at 3, 6, 9 and 12 dB are then collected. Next, MECG and noise signals for the same subjects are downloaded. The composite signal, as given by (1), acts as input data for the proposed CNN model.

$$\text{Mixed} = \text{FECG} + \text{MECG} + \text{Noise1} + \text{Noise2} \quad (1)$$

3.2 Development of Proposed 1D CNN Model

A CNN consists of a sequence of layers, and each layer transforms an input image to an output image with some differentiable function that may or may not have parameters. Details about the architecture of CNN models can be obtained from [1]. All the signals and algorithms have been processed and developed in MATLAB environment on a tenth generation computer having core i3 processor with Windows 10 operating system, 8 GB RAM and hard disk of 1 TB. Deep learning toolbox and the specified CNN parameters have been suitably chosen.

The input data (mixed-signal) is obtained from (1). Its size is 87×2500 . However, for 1D CNN network, mixed signal is reshaped as $1 \times 2500 \times 1 \times 87$. Accordingly, the target (FECG at 0 dB) is set as 87×2500 . The input image layer size is thus 1×2500 . Four fully connected layers having 2500 neurons are used as a hidden layer. Finally, the output layer is selected as the regression layer as the output signal has to be extracted and enhanced. The proposed 1D CNN network architecture is shown in Fig. 1.

The training rate for the training is to be suitably adjusted between 0 and 1 as very low the learning rate results in long training time, and also high rate may result in sub-optimal results. The factor for dropping the learning rate, learn rate drop factor is a scalar from 0 to 1. Gradient threshold helps in minimizing the loss functions, and its value should be between 0 and ∞ . When the gradient exceeds the value of

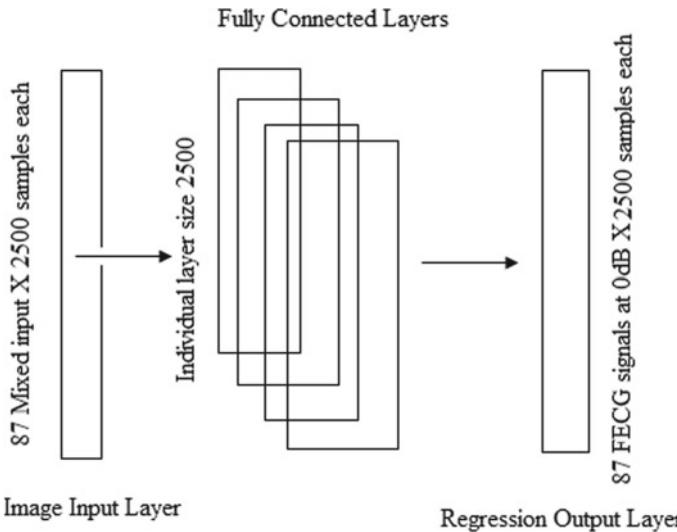


Fig. 1 Proposed 1D CNN model for extraction and enhancement of FECG signal from mixtures of MECG signals and noises

gradient threshold, then the gradient is clipped according to the gradient threshold method. An epoch is the full pass of the training algorithm over the entire training set. Mini-batch size is the number of samples used for each training to evaluate the gradient of the loss function and update the weights. An epoch corresponds to the number of cycles allotted to complete the learning process.

The training algorithm used is “Adam” (adaptive moment estimator), with an initial learning rate of $1e^{-5}$, learning rate drop factor as 0.09, maximum epochs set as 650, gradient threshold as 10 and batch size as 50.

4 Results and Discussion

The proposed 1D CNN model is trained with the specified parameters. The training curve is shown in Fig. 2. The proposed model is successfully trained within the specified number of epochs.

The mini-batch root mean square error (RMSE) obtained after the training is 85.09, and the mini-batch loss is 3620.4. Figure 2 shows the iteration-wise learning progress. These errors can be further reduced by increasing the number of epochs.

Figure 3 shows the result of the trained model with FECG at 0 dB, mixed FECG and the predicted FECG signal.

The trained model is tested with an additional 10 FECG signals from the same database. The input to the model is arranged and provided during the training period. The signal to noise ratio (SNR_{in}) of the test signal, which is the mixed signal, is

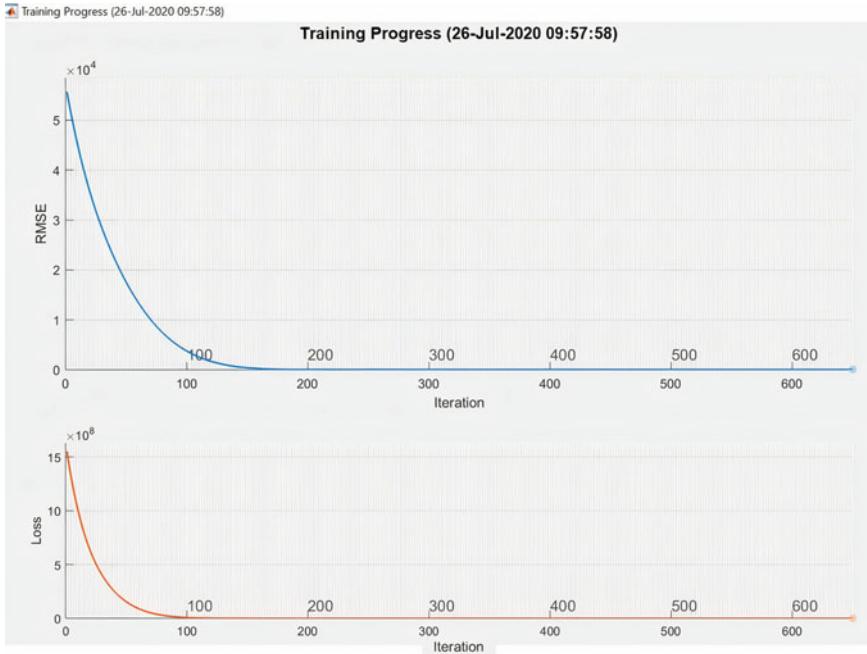


Fig. 2 Training of proposed 1D CNN model for FECG signal enhancement

measured. The test signal is then applied to the trained model, and SNR of predicted signal (SNRo) is noted. The RMSE between the test and the predicted signal is also evaluated. Table 1 shows the SNRin, SNRo, improvement in SNR (SNRim) and RMSE of the tested samples.

From Table 1, it is observed that the signal content of FECG signals is less as compared to noise content of the signal, and hence, denoising is required before processing the FECG signals. Since the noise content of test samples is variable, thus the trained model is predicting different values for corresponding FECG signals which are indicated as SNRo. From the table, it is inferred that the model is capable of improving SNR (SNRim) in the range of 0–23 dB, depending upon the level of noise present in the FECG signals. The RMSE values reported for the test samples are also within the acceptable limits. Signals having high noise content would result in a higher value of RMSE, and the less noisy signal would result in lower RMSE, which is demonstrated in the entries of Table 1. Jagannath et al. [9] developed a Bayesian deep learning consisting of Bayesian filter and a deep learning model to extract FECG from MECG signal and reported SNRim of 25.76 dB which is almost same as that reported in the proposed work. Different independent component analysis-based FECG enhancement techniques reported in [10] could able to improve SNR of synthetic FECG signal to 25 dB for signals having initial noise levels up to -30 dB. Radana et al. in [11] compared least mean square (LMS) and recursive least square

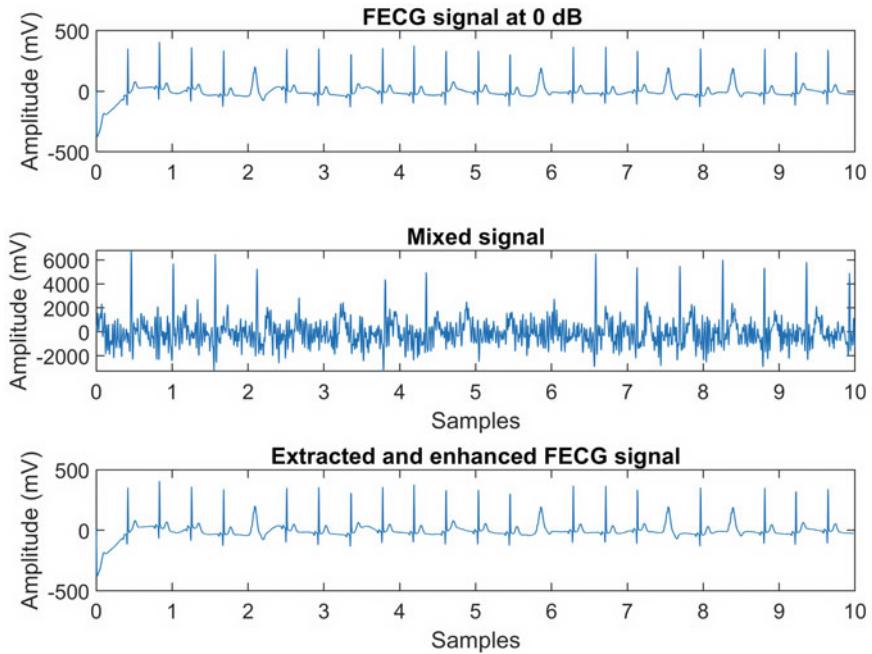


Fig. 3 FECG signal at 0 dB, mixed signal and the output of the trained model for the same FECG signal

(RLS) filter to extract FECG from abdominal MECG signal and obtained SNRim as 7.7 dB for LMS and 4.6 dB for RLS algorithm, respectively. Deep recurrent neural networks proposed in [12] obtained 7.71 dB of SNR from an input signal having -8.82 dB of SNR. Deep neural networks based on denoising autoencoders were applied to ECG signals to suppress noise, reported an improvement in SNR ranging from 21.56 dB to 22.96 dB with RMSE 0.037 [13]. In the proposed, although the SNRim is comparable with the existing methods, RMSE is very low, i.e., of order 10^{-3} , which is significantly less than those reported in the literature.

5 Conclusion

FECG signal reflects the electrophysiological activity of the fetal heart. These signals are required to detect fetus growth and abnormalities related to fetus heart. Generally, the non-invasive method is preferred to collect FECG, but these signals are often contaminated with various types of noises. Noises prevent correct diagnosis and are required to be removed before the analysis. Deep learning models, especially CNN, produce accurate and faster results even for signals having voluminous data. In this

Table 1 Performance assessment of the proposed 1D CNN over 10 test samples of MIT-BIH database

Test Sample	SNRin (dB)	SNRo (dB)	SNRim (dB)	RMSE ($\times 10^3$)
1	-25.1456	-7.9605	17.1851	0.5601
2	-29.432	-5.8413	23.5907	0.8642
3	-21.5736	-8.0347	13.5389	0.6077
4	-29.7633	-29.7515	0.0118	0.0015
5	-12.4957	-6.7077	5.788	0.5893
6	-20.7772	-6.6498	14.1274	0.7141
7	-17.3129	-6.6487	10.6642	0.6008
8	-15.884	-15.8795	0.0045	0.0026
9	-20.0756	-20.0746	0.001	0.0015
10	-20.2909	-4.3934	15.8975	0.8151

paper, the concept of 2D CNN has been modified to 1D CNN, and the same has been successfully tested over ten noisy FECG data of the MIT-BIH database. The proposed 1D CNN model is tested, and the obtained results are also found to be satisfactory. In future, this model would be used for training of an extensive database.

References

1. Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ (2019) 1D convolutional neural networks and applications: a survey. arXiv preprint [arXiv:1905.03554](https://arxiv.org/abs/1905.03554)
2. Webster JG (2014) The physiological measurement handbook. CRC Press, Boca Raton
3. Alpaydin E (2020) Introduction to machine learning. MIT Press, Cambridge
4. Sameni R, Shamsollahi MB, Jutten C (2008) Model-based bayesian filtering of cardiac contaminants from biomedical recordings. *Physiol Measur* 29(5):595
5. Sameni R (2008) Extraction of fetal cardiac signals from an array of maternal abdominal recordings. Ph.D. dissertation, Institut National Polytechnique de Grenoble-INPG; Sharif University of ...
6. Dey N, Prasad Dash T, Dash S (2011) ECG signal denoising by functional link artificial neural network (flann). *Int J Biomed Eng Technol* 7(4):377–389
7. Lei X, Zhang Y, Lu Z (2016) Deep learning feature representation for electrocardiogram identification. In: 2016 IEEE international conference on digital signal processing (DSP). IEEE, pp 11–14
8. Andreotti F, Behar J, Zaunseder S, Oster J, Clifford GD (2016) An open-source framework for stress-testing non-invasive foetal ECG extraction algorithms. *Physiol Measur* 37(5):627
9. Jagannath D, Raveena Judie Dolly D, Peter JD (2019) A novel bayesian deep learning methodology for enhanced foetal cardiac signal mining. *J Exp Theor Artif Intell* 31(2):215–224
10. Jaros R, Martinek R, Danys L, Latal J, Siska P (2019) Fetal ecg signal processing by different ica-based algorithms. In: International symposium on advanced electrical and communication technologies (ISAECT). IEEE, pp 1–4

11. Kahankova R, Martinek R, Bilik P (2016) Non-invasive fetal ecg extraction from maternal abdominal ecg using lms and rls adaptive algorithms. In: International Afro-European conference for industrial advancement. Springer, pp 258–271
12. Antczak K (2018) Deep recurrent neural networks for ecg signal denoising. arXiv preprint [arXiv:1807.11551](https://arxiv.org/abs/1807.11551)
13. Xiong P, Wang H, Liu M, Zhou S, Hou Z, Liu X (2016) Ecg signal enhancement based on improved denoising auto-encoder. Eng Appl Artif Intell 52:194–202

Chapter 35

HMM Model for Brain Tumor Detection and Classification



Parth Sharma and Rakesh Sharma

1 Introduction

Neural networks are a bunch of algorithms that resemble the working of human neurons such that the system can recognize different patterns, just like the human brain [1]. These neural networks can convert the raw data to a recognizable pattern using machine learning perception like clustering [2]. Various neural network architectures are utilized nowadays with a different selection of hyper-parameters for many applications. These choices of hyper-parameters lead to different neural network architectures for varied applications [3]. Nowadays, neural network chips and boards are present for different applications [4]; however, a vast majority uses neural network simulators on standard serial machines. Artificial neural networks have three layers, namely: the input layer, concealed layer, and output layer. The raw data is fed to the input layer, and this input layer presents the data to the concealed layer, where this data is processed with the help of a linear weighted and thresholding operation. The weights are learned with each learning example in the training phase [5]. Finally, the processed data is fed to the output layer and can be accessed by the user.

The word tumor is frequently used to denote the inflammation of body tissues. In this process, the uncontrolled growth of tissues starts inside the body [6]. The brain tumors can be classified as primary and secondary tumors based on their place of origin. A primary brain tumor takes birth in the brain region, and a secondary brain tumor originates at other body organs and by spreading reaches the brain region [7]. Brain tumors can also be classified based on the nature of the tumor. There are three types of brain tumors based on the nature of the tumor: benign, pre-malignant, and malignant. The benign tumor is a steady growth of tissues where the tissues around

P. Sharma (✉) · R. Sharma

Department of Electronics and Communication Engineering, National Institute of Technology,
Hamirpur, Hamirpur, India

e-mail: rakesh.sharma@nith.ac.in

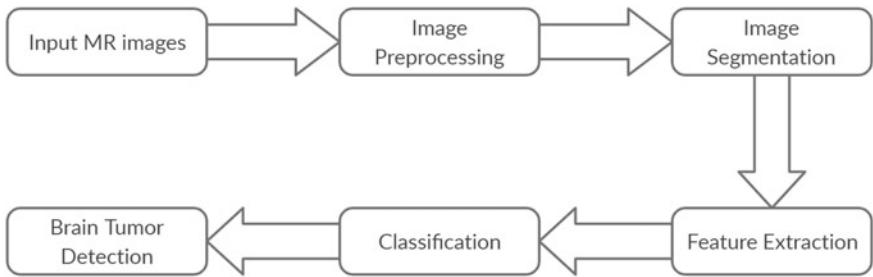


Fig. 1 Basic steps required in brain tumor detection

the tumor are not affected much. The pre-malignant brain tumor refers to the pre-cancerous phase or the initial stage of cancer. It may be life-threatening if the tumor is not treated at this stage. A malignant tumor belongs to the final stage of the tumor. In this stage, treatment of a brain tumor is highly painful and mostly leads to the death of the patient.

The brain tumor detection requires different basic steps as shown in the flow diagram (Fig. 1).

In pattern recognition, classifier plays a vital role. HMM is one of the statistical learning model which has the ability to absorb both variability as well as similarity between patterns. This HMM classifier is based on empirical risk minimization or ERM principal where the decision rule is chosen based on finite known examples or the training Set.

2 Recent Work

From the past decade, brain tumor detection using machine learning algorithms is getting popular. A huge advancement is seen in the field of brain tumor detection. Some of the procedures are analyzed below.

A brain tumor model using a thresholding algorithm was proposed by Hunnur et al. [8]. Various brain tumor detection techniques were also compared in this research as well. The brain tumor could be detected from the MR images collected from the patient database using an efficiently proposed technique. Efficient tumor detection was shown as per the results achieved when performing simulations. Sobel edge detection operation was used to extract the boundary of the tumor. The research described the size and stage of the tumor.

Mathew and Anto [9] proposed and implemented a novel mechanism to detect and classify brain tumors. Noise removal, feature extraction, segmentation, and classification were the important steps performed in this research. The anisotropic diffusion filters were used in the proposed work to preprocess MR brain images. The discrete wavelet transform (DWT)-based features are extracted to perform feature extraction.

For the segmentation stage, the extracted features were given as input. To perform tumor segmentation and classification, the support vector machine (SVM) classifier was used. Around 86% of accuracy was achieved as per the given outputs.

Kurnar et al. [10] used the K-means clustering algorithm for detecting brain tumors. The recommended algorithm was based on segmentation and morphological operation. Initially, the preprocessing of the MRI scan was done. Then, K-means clustering was implemented in this image. To extract tumors from the preprocessed MR scanned image, morphological operations were carried out. At last, the computation of the extracted tumor part was done. Morphological operation eliminated the pixels not related to the tumor. In this way, the extraction of the tumor was done. Later, the tumor region was measured. A tool named SCILAB was used in this work. This tool is an open-source software. It was possible to use a complete application for detecting a tumor in other body organs.

An attempt for developing an automated integrated image segmentation model was made by Jagan [11]. The main objective of his work was to detect the tumor in three-dimensional MR images. He combined the fuzzy C clustering algorithm with the most recognized enhanced EM (Expectation maximization) technique. This system combines the segmentation result of the most recognized technique in an optimum manner. It also exhibits the enhancement in the segmentation of brain MR image. To evaluate the performance results of the recommended system, simulated brain fluid-attenuated inversion recovery MR images and real brain datasets were used. In contrast to state-of-the-art techniques, the recommended approach performed better in terms of accuracy, sensitivity, and specificity as per the achieved tested results.

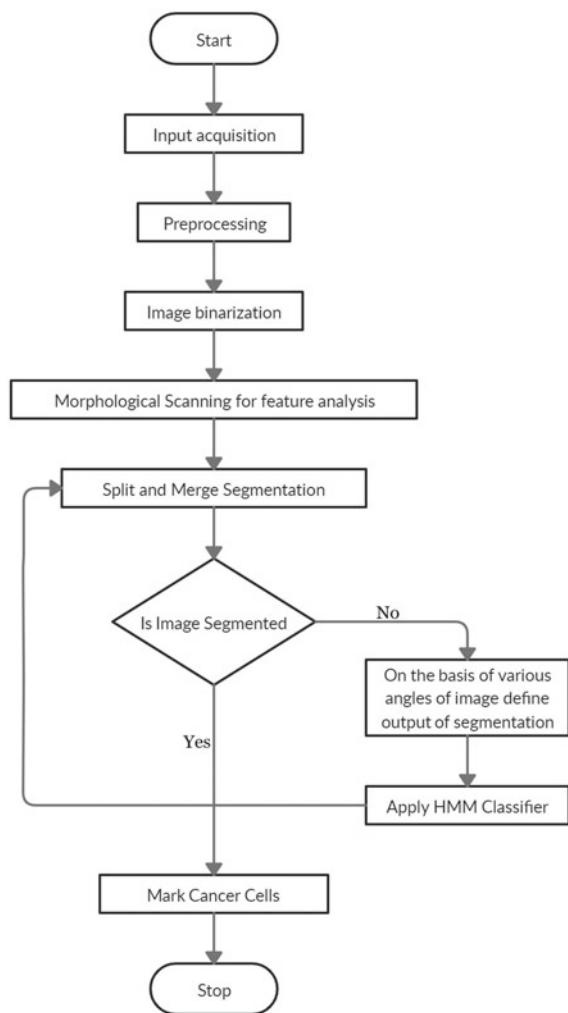
Watershed Dynamic Angle Projection—Convolution neural network (WDAPP-CNN)-based state-of-the-art tumor detection technique was proposed by Jemimma et al. [12]. The watershed algorithm performs the segmentation of the tumor region in an accurate manner. The texture features of the brain were extracted using a dynamic angle projection pattern. The convolution neural network (CNN) classified the benign and malignant parts of the MRI brain scan. A dataset called BRATS was used in this work for efficiently detecting and testing the irregularity of the brain image. The tested results depicted that the proposed approach achieved better dice score efficiency and sensitivity of 93.5% and 94.2% respectively than other existing approaches.

Sazzad et al. [13] proposed a novel automatic technique for brain tumor detection. The proposed approach included improvement at the early stage for minimizing changes in gray-scale color. To eliminate redundant noise, filter operations were carried out. Noise removal to large extent proved very useful in segmentation. OTSU's segmentation based on the threshold was carried out in place of color segmentation for the testing of grayscale images. At last, feature knowledge given by pathologists was utilized for the detection of cancerous areas. In contrast to other existing algorithms, the recommended algorithm provided much efficient results in terms of accuracy. The proposed algorithm also maintained the satisfactory accuracy level provided by pathologists.

3 Research Methodology

In this paper, a new method is presented for brain tumor detection and classification. The brain MR images are taken as the input to the algorithm. A combination of region-based segmentation and classification is applied to the MR images for classifying the cancerous and non-cancerous brain cells. The region-based segmentation is performed using a split and merge technique, and the classification is done using an HMM classifier. The steps involved in brain tumor detection and classification algorithm are shown as a flowchart in Fig. 2.

Fig. 2 Flowchart demonstrating proposed methodology for brain tumor detection and classification



MR images chosen for the process are acquired from the standard BRATS 2013 dataset.

The MR image is firstly converted to 256×256 pixels resolution. RGB to gray-scale conversion is then done on the image to eliminate any color component present. After removing the color component, the median filter is applied to the gray-scale MR image. The median filter removes the noise present in the image along with preserving the edges. The median filtered MR image is then passed through skull stripping algorithm. Skull stripping is one of the special preprocessing techniques in which the brain region is separated from other redundant body parts like the skull [14]. The application of skull stripping on MR image leads to better segmentation of the critical brain region from the redundant non-brain area in the MR image.

The gray-scale image needs to be converted into a binary image in order to segment the brain tumor. Otsu's thresholding operation is used for this purpose. It involves the selection of an optimum thresholding value after considering and analyzing all possible threshold values [15].

In the segmentation process, split and merge algorithm is used that splits MR image into homogeneous quadrants and the process continues till all the quadrants formed are homogeneous [16]. Finally, similar portions are merged.

For feature extraction, scale invariant feature transform (SIFT) method is used [17]. Each image has unique corners, and these corners remain unchanged on the application of rotation operation on the MR image. But the corners change if a scaling operation is applied to the MR image. Hence, the window that initially used to detect corners will not be able to detect these corners. The SIFT method is applied to overcome this problem of scale variance.

Finally, HMM or hidden Markov model is used for classification. It is a probabilistic model in which the system is assumed to be a Markov process with memoryless states, and the classification is performed based on the calculated hidden state probabilities.

Figure 3a–d shows the MR image at various processing steps of the proposed algorithm. Figure 3a shows the input image that is used for the performance analysis. Figure 3b depicts the boundary box to locate the tumor. Figure 3c shows the classification results of the cancerous cells from non-cancerous cells with HMM classifier. Figure 3d presents the segmented tumor from the brain region of the MR image.

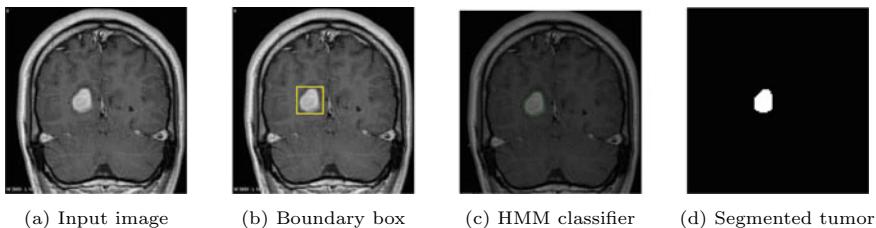


Fig. 3 Brain tumor detection using HMM classifier

4 Results and Discussion

The set of five images is used to perform the comparison between weight-based classifiers support vector model (SVM) and the probabilistic hidden Markov model (HMM) for brain tumor detection and classification. These images are shown in Fig. 4.

4.1 Performance Analysis Parameters

1. **Mean Square Error (MSE):** Mean square error or MSE is defined as the average of square of the difference between the original image and the graded image. The mathematical expression of MSE is given by

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|f(i, j) - g(i, j)\|^2, \quad (1)$$

where f : original image matrix

g : graded image matrix

m : number of row pixels

n : number of column pixels.

2. **Peak Signal to Noise Ratio (PSNR):** It is the ratio of the maximum signal power to the power of the noise present in the image. The mathematical formula to calculate PSNR is given by

$$\text{PSNR} = 20 \log_{10} \frac{\text{MAX}_f}{\sqrt{\text{MSE}}}, \quad (2)$$

where MAX_f : maximum pixel power in the image.

3. **Accuracy:** Accuracy is the ratio of number of correctly classified samples to the total number of samples used for classification. The mathematical expression of Accuracy is given by

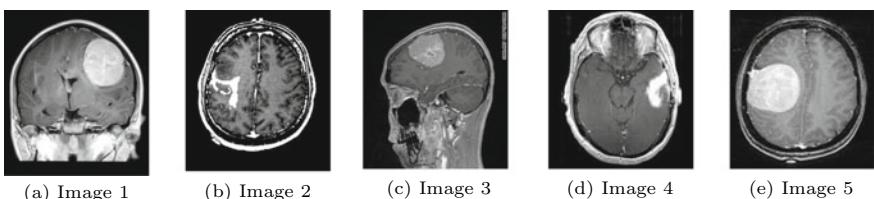


Fig. 4 Test images dataset

Table 1 Performance parameters for different classifier on test images

Image Number	MSE	PSNR (dB)	Accuracy (%)
<i>(a) SVM classifier</i>			
Image 1	4	36	80
Image 2	2.4	40	83.46
Image 3	1.8	43	86.78
Image 4	2	42	84.67
Image 5	1.8	43	86.78
<i>(b) HMM classifier</i>			
Image 1	3.1	38	90.2
Image 2	2.2	41	94.56
Image 3	1.4	45	96.8
Image 4	1.8	43	95
Image 5	1.5	44	94

$$A_i = \frac{T}{N} * 100, \quad (3)$$

where T : number of correctly classified samples

N : total number of samples for classification.

These performance parameters are estimated for both SVM-based and the proposed HMM-based classification on the set of five test images. Table 1a, b presents the performance parameter values on test images for SVM-based and HMM-based classifier, respectively.

5 Conclusion

In this paper, a HMM-based classification algorithm for brain tumor detection is presented. Also, its efficacy in terms of detection accuracy, PSNR, and MSE is demonstrated. The performance analysis suggests that both the methods, weight-based classification method (SVM) and probabilistic-based model (HMM) perform quite similar in terms of PSNR and MSE for classification. However, HMM-based classification method performs quite well compared to weight-based classification in terms of detection accuracy. There is an increase in detection accuracy from 83.982 to 94.224% by the use of the HMM-based classification for brain tumor detection. Also, the value of PSNR increases from 41.4 to 42.45 dB and MSE decreases from 2.54 to 2.12. Hence, it is concluded that HMM-based classifiers can be a good alternative to detect and classify brain tumors in MR images as compared to weight-based classification methods.

References

1. Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, Mazoyer B, Joliot M (2002) Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage* 15(1):273–289
2. Van Dijk KR, Hedden T, Venkataraman A, Evans KC, Lazar SW, Buckner RL (2010) Intrinsic functional connectivity as a tool for human connectomics: theory, properties, and optimization. *J Neurophysiol* 103(1):297–321
3. Oksengaard A, Haakonsen M, Dullerud R, Engedal K, Laake K (2003) Accuracy of CT scan measurements of the medial temporal lobe in routine dementia diagnostics. *Int J Geriatr Psychiatry* 18(4):308–312
4. Bhima K, Jagan A (2016) Analysis of MRI based brain tumor identification using segmentation technique. In: 2016 international conference on communication and signal processing (ICCSP). IEEE, pp 2109–2113
5. Rueda MR, Posner MI, Rothbart MK (2005) The development of executive attention: contributions to the emergence of self-regulation. *Dev Neuropsychol* 28(2):573–594
6. Sorg C, Riedl V, Mühlau M, Calhoun VD, Eichele T, Läer L, Drzezga A, Förstl H, Kurz A, Zimmer C et al (2007) Selective changes of resting-state networks in individuals at risk for alzheimer's disease. *Proc Nat Acad Sci* 104(47):18760–18765
7. Stam CJ (2014) Modern network science of neurological disorders. *Nat Rev Neurosci* 15(10):683–695
8. Hunnur MSS, Raut A, Kulkarni S (2017) Implementation of image processing for detection of brain tumors. In: 2017 international conference on computing methodologies and communication (ICCMC). (IEEE), pp 717–722
9. Mathew AR, Anto PB (2017) Tumor detection and classification of mri brain image using wavelet transform and svm. In: 2017 international conference on signal processing and communication (ICSPC). IEEE, pp 75–78
10. Kumar M, Sinha A, Bansode NV (2018) Detection of brain tumor in MRI images by applying segmentation and area calculation method using scilab. In: 2018 4th international conference on computing communication control and automation (ICCUBEA). IEEE, pp 1–5
11. Jagan A (2018) A new approach for segmentation and detection of brain tumor in 3d brain mr imaging. In: 2018 2nd international conference on electronics, communication and aerospace technology (ICECA). IEEE, pp 1230–1235
12. Jemimma T, Vetharaj YJ (2018) Watershed algorithm based dapp features for brain tumor segmentation and classification. In: 2018 international conference on smart systems and inventive technology (ICSSIT). IEEE, pp 155–158
13. Sazzad TS, Ahmed KT, Hoque MU, Rahman M (2019) Development of automated brain tumor identification using mri images. In: 2019 international conference on electrical, computer and communication engineering (ECCE). IEEE, pp 1–4
14. Kalavathi P, Prasath VS (2016) Methods on skull stripping of MRI head scan images'a review. *J. Dig. Imaging* 29(3):365–379
15. Tirpude N, Welekar R (2013) Effect of global thresholding on tumor-bearing brain MRI images. *Int J Eng Comput Sci* 2(3)
16. Rana R, Bhadauria H, Singh A (2013) Study of various methods for brain tumour segmentation from MRI images. *Int J Emerg Technol Adv Eng* 3(2):338–342
17. Cheung W, Hamarneh G (2009) *n*-SIFT: *n*-dimensional scale invariant feature transform. *IEEE Trans Image Process* 18(9):2012–2021

Chapter 36

ANN Control Algorithms with Different Training Methods as Applied to PMSM Drive



Krishna Kokre and S. V. Jadhav

1 Introduction

PMSMs are becoming popular due to the advantages over induction motors, like improved efficiency, higher torque to weight ratio, etc. Hence, PMSM drives are increasingly used in industries for applications such as robotics, CNCs and industrial process automation[1]. The drive must operate dynamically at adjustable speeds with precision and robustness. The conventional PID controllers cannot meet the required specifications, especially in the presence of motor parameter variations, nonlinearity, changes in set points, etc. All such situations should be handled by controller, so as to ensure high quality of final product. Hence, there is demand for modern and intelligent control algorithms like ANN which are capable of handling situations like these. With the invention of vector control of induction machine by K. Hasse and F. Blashke, the research of AC motor drives accelerated significantly [2]. Since then several control techniques are investigated for electrical drives. Recently, artificial intelligence-based techniques such as ANN, fuzzy and neurofuzzy are implemented for effective speed control [3]. ANN is one of the popular control techniques researched and used.

For PMSM, it has a characteristic parallel and distributed processing and can map nonlinear relations between input and output. The ANN-based speed controller design for DC motor is presented in [4]. The multi-layer ANN structure described, mainly, performs two functions. The first one is to recognize the nonlinear system dynamics, and the second is to control the motor voltage so that the speed and the position are made to follow predefined tracks. In [5], ANN is used to capture the nonlinear model of motor and load. Further, ANN structure is also used to control

K. Kokre (✉) · S. V. Jadhav

Department of Electrical Engineering, College of Engineering, Pune, India

e-mail: kokrek18.elec@coep.ac.in

S. V. Jadhav

e-mail: svj.elec@coep.ac.in

speed trajectory accurately. ANN is also used for the identification of nonlinear PMSM model using three different topologies, namely current, voltage, and speed topologies in [6]. The first topology is useful when parameters are known while the remaining two work with unknown parameters. The research work also presents the design of ANN-based control for exact speed tracking. ANN-based model reference adaptive system (MRAS) speed observer is presented in [7]. A robust PMSM servodrive for nonlinear controller characteristics is reported in [8]. [9] proposes speed and stator flux estimation and control of direct torque-controlled IM. ANN control is proved to be better than PI control in forward mode, speed reversal mode and flux weakening mode [10], and its performance is compared with two more AI-based nonlinear controllers: fuzzy-based control and fuzzy SMC-based control in [11, 12]. An adaptive estimator is presented in [13] for the estimation of rotor speed in a sensor less vector-controlled IM drive. It is proved that the MRAS with an ANN-based model improves the stability of the drive at low speeds in regenerating mode. It is proved in [14] that ANN can be effectively used for compensating the torque (or speed) ripple apart from achieving the satisfactory response of the drive [15]. It presents a solution for speed control of high-performance PMSM drive with ANN-tuned PI speed controller, and it is validated for dynamic load conditions. The PI parameters are tuned online using dynamic back propagation with Levenburg–Marquardt training algorithm [16]. One more algorithm—resilient back propagation training algorithm (RBP) (Riedmiller, et.al, 1992)—is designed for speed control of PMSM. ANN online and offline training algorithms are tested for space vector modulation-based DTC of IM [17]. Scaled conjugate training method offers not only robustness but also fast transient and smooth steady-state response [18]. In this paper, an ANN speed controller is with above-mentioned three training algorithms that are presented.

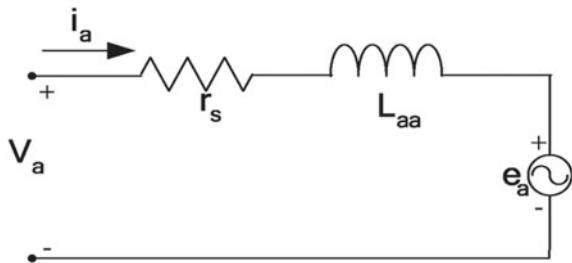
2 Modelling of PMSM

A PMSM has a wound stator, a permanent magnet rotor assembly, and an internal or external devices like encoder to sense rotor position. The encoder provides position feedback for adjusting frequency and amplitude of stator voltage reference properly to maintain rotation of the rotor. The permanent magnets are embedded in the rotor (interior permanent magnet-type rotor). Due to the absence of field winding, PMSM has advantages over its counterpart induction motors. These include low rotor inertia, efficient heat dissipation, and reduction of motor size and hence increase in torque to weight ratio.

Figure 1 shows the equivalent circuit of the single-phase star-connected stator of PMSM. The three-phase voltages of the PMSM in stationary reference frame of stator can be described as follows:

$$V_a = r_s i_a + \frac{d\lambda_a}{dt} \quad (1)$$

Fig. 1 Equivalent circuit of stator of PMSM



$$V_b = r_s i_b + \frac{d\lambda_b}{dt} \quad (2)$$

$$V_c = r_s i_c + \frac{d\lambda_c}{dt} \quad (3)$$

where V_a , V_b and V_c are the phase voltages, the stator winding resistance per phase is r_s , stator three-phase currents are i_a , i_b and i_c respectively, and also the flux linkages of the three-phase windings are λ_a , λ_b and λ_c , respectively. The three-phase stationary reference frame variables of PMSM are expressed in two-phase d - q synchronously rotating reference frame by Park's transformation.

The stator flux linkage equations of PMSM in d - q reference frame rotating at synchronous speed are given by

$$V_{sd} = R_d * i_{sd} + L_{sd} * \frac{di_{sq}}{dt} + \psi_{sq} * \omega \quad (4)$$

$$V_{sq} = R_q * i_{sq} + L_{sq} * \frac{di_{sd}}{dt} - \psi_{sd} * \omega \quad (5)$$

where ω is rotor speed, L_{sq} and L_{sd} are the quadrature and direct axis inductances, and R_q and R_d are the quadrature and direct axis resistances, respectively. The permanent magnet excitation can be modelled as a constant current source i_{fr} . The rotor flux is along the d -axis, so d -axis rotor current is i_{fr} . The q -axis current in the rotor is zero because it is assumed that there is no flux along that axis. This is called rotor field orientation. Then the stator flux linkages can be written as

$$\psi_{sd} = L_{sd} * i_{sd} + \psi_{fr} \quad (6)$$

$$\psi_{sq} = L_{sq} * i_{sq} \quad (7)$$

The electromagnetic torque is given by

$$T_e = \frac{3}{2} * \frac{p}{2} \{ \psi_{fr} * i_{sq} + (L_{sd} - L_{sq}) * i_{sd} * i_{sq} \} \quad (8)$$

$$\psi_{fr} = L_m * i_{fr} \quad (9)$$

The mechanical torque is given by

$$T_e = J \frac{d\omega}{dt} * \frac{1}{P} + B * \omega \frac{1}{P} + B * T_l \quad (10)$$

where J —moment of inertia, B —viscous friction coefficient, T_l is load torque, and the d and q-axes currents are constant in rotor reference frame. If δ is angle between stator and rotor flux, Eq. 8 is given by

$$T_e = \frac{3}{2} * \frac{p}{2} \left[\psi_{fr} * \sin(\delta) + \frac{1}{2} (L_d - L_q) * i_s^2 \sin(2\delta) \right] \quad (11)$$

For field oriented in PMSM substituting $\delta = 90$ in Eq. (11), therefore, the electromagnetic torque equation changes to

$$T_e = \frac{3}{2} * \frac{p}{2} \psi_{fr} * i_s \quad (12)$$

The control strategy when $\delta = 90$ is also called as constant torque angle control. This type of control is used for speeds less than base speed of the motor.

3 Field-Oriented Control of PMSM

Field-oriented control (FOC), is a variable frequency drive (VFD) control method where the permanent magnet produced flux is assumed to be present along the d-axis of the rotor, as we discussed in previous section. So the magnetizing current drawn by stator (i_{ds}) should be zero, as the magnetization is done completely by rotor permanent magnets. Thus, stator draws only torque component i_{qs} of the current from the supply. Figure 2 shows block diagram of field-oriented control of PMSM using artificial neural network controller in the speed loop.

The above system consists of four main parts: a motor (PMSM), a PWM inverter, two current control loops with PI and a speed control loop with ANN.

As shown in the diagram, the control system of the drive calculates the torque reference value by using ANN-based speed control loop. Further, two PI controllers generate the voltage references. Thus, the value of d-axis current i_{ds} is regulated to zero, and the reference value of i_{qs} is calculated to meet the torque requirements. The reference voltage values are realized by space vector modulation (SVM). The

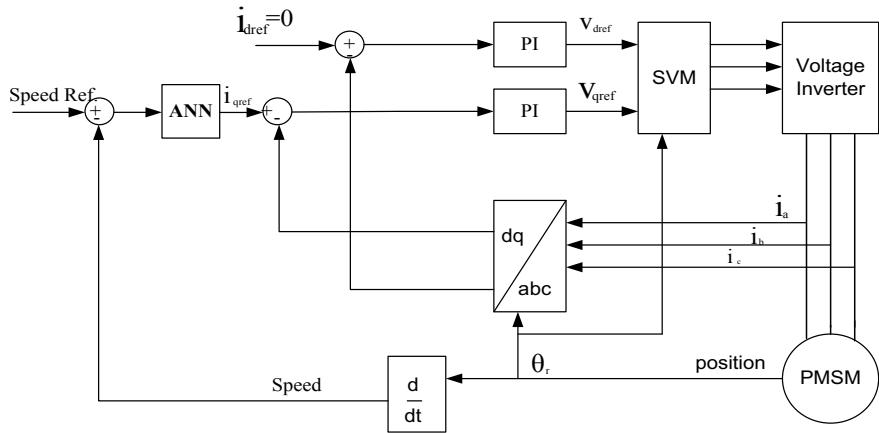


Fig. 2 Block diagram of FOC of PMSM

speed controller consists of ANN module as shown in Fig. 3. Error in the speed and its time derivative are inputs; whereas, reference current is the output of the network.

As shown in Fig. 4, the neural network contains three layers of neurons: input layer, hidden layer and output layer. The number of neurons in the input and output

Fig. 3 Structure of speed controller

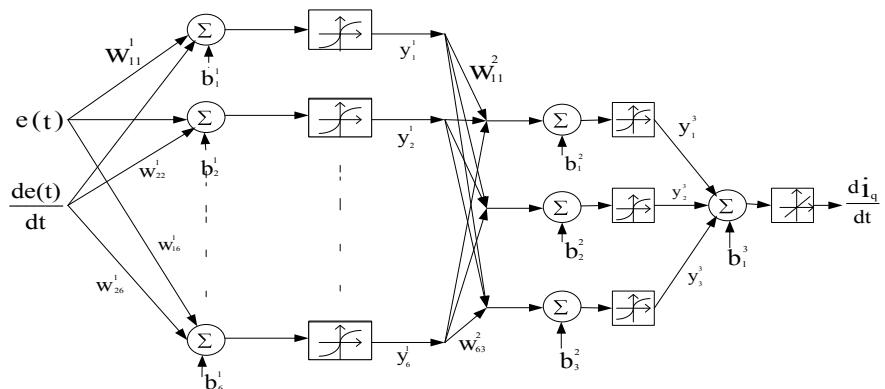
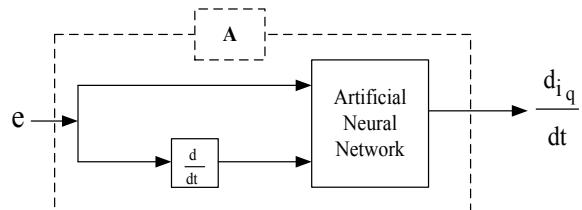


Fig. 4 Multi-layer ANN structure used for RBP algorithm

layers depends on input and output variables. The number of hidden layers and the number of neurons in each layer depend on system dynamics and the desired degree of accuracy.

The neural network structure has two important phases: training and testing the network. A diverse data set of input and output mappings is selected for training. During training, the weights of neural network are adjusted so as to map the input of the system to its output. The neural network training is repeated until the error converges to a value less than an acceptable error value. After the network is trained, it is tested to correctly predict the system output for a given input. The weight improvement to minimize the error applies the standard gradient descent optimization method, where the weight improvement is done one at a time starting backward from the output layer. Consider that a neural network is being trained with the input pattern, and the squared output error for all the output layer neurons of the network is given by

$$E_p = \sum_{i=0}^n (E_i^p - Y_i^p)^2 \quad (13)$$

where E_i^p = desired output of the p th neuron in the output layer, Y_i^p = corresponding actual output, n = dimension of the output vector, Y_p = actual network output vector, and E_p = corresponding desired output vector. The total sum of square error (SSE) for the set of m patterns is then given by

$$\text{SSE} = E = \sum_{m=1}^m E_p = \sum_{m=1}^m \sum_{i=0}^n (E_i^p - Y_i^p)^2 \quad (14)$$

The weights of the neurons are altered to minimize the value of the objective function SSE by gradient descent method, as mentioned before. The weight update equation is then given as

$$W_{ij}(k+1) = W_{ij}(k) - \sigma \frac{dE_p}{dW_{ij}(k)} \quad (15)$$

where σ = learning rate, $W_{ij}(k+1)$ = new weight between i th and j th neurons, and $W_{ij}(k)$ = corresponding previous weight. The weights are updated iteratively for all the training patterns. In some cases, mean square error is taken as the objective function. In order to converge SSE to a global minimum, a momentum term is added to the right of, where a small value for further improvement of back propagation algorithm is possible by making the learning rate step size adaptive, i.e.

$$\sigma(k+1) = \mu \sigma(k) \mu < 1.0 \quad (16)$$

So that oscillation becomes minimum as it settles to the global minimum point.

(A) Resilient Back Propagation (RBP) Algorithm

Back propagation algorithm has an issue with setting the learning rate as one has to compromise between the accuracy or the computation time. The RBP algorithm is an up-gradation of back propagation algorithm, and this algorithm uses the sign (positive or negative) of the gradient to show the direction of the adjustment weight. It introduces a momentum term such that it scales the influence of previous step on the present step. The momentum term accelerates the convergence in shallow regions of the error function E. It generally converges much faster than other algorithms. This algorithm uses the sign of the gradient to show the direction of the adjustment of weight. The presence of momentum term makes the RBP algorithm parameter dependent and less robust.

(B) Levenberg-Marquardt (L-M) Algorithm

A combination of Gradient Descent and Gauss-Newton methods are used in L-M algorithm. In number of applications, the L-M algorithm can assure problem-solving through its adaptive behaviour. The L-M update rule is written as

$$x(k+1) = x(k) - [J^T J + \mu J]^{-1} J^T E_p \quad (17)$$

where $x(k+1)$ is a new weight, $x(k)$ is a current weight, and J is a Jacobian matrix expressed in terms of error function E .

(C) Scaled Conjugate Gradient (SCG) Algorithm

SCG is one way to solve the system of linear equations. In SCG training algorithm, the search is performed along with conjugate directions, which produce generally faster convergence than gradient descent directions. The iteration of conjugate gradient is defined in the equation below

$$x_k = x_{k-1} + \alpha_k d_{k-1} \quad (18)$$

where k is the iteration index, α_k is the conjugate parameter at k th iteration, and d_k is the training direction search vector. The conjugate parameter is a function of quadratic approximation of the error function which makes it more robust and independent of user defined parameters.

4 ANN Speed Controller Implementation

The input to the ANN is speed error and its derivative in a discrete way. The output signal of the ANN is derivative of current reference, i.e. torque command, and it is necessary to calculate integral of this signal. Two hidden layers are selected: the first hidden layer has six neurons, and the second hidden layer has three neurons. The

ANN structure is shown in Fig. 4. The next task is to generate a training data. For training data, the well-tuned PI-based simulation of PMSM is performed. The speed response s obtained is divided into 1000 time steps for training data. At each step, (n) of speed error value and its derivative are calculated, which results in a pair of input signals of ANN, those are $e(n)$, $de(n)/dt$. With this training data, the above network is trained using back propagation algorithm.

$$e(n) = w_{\text{ref}}(n) - w_{\text{actual}}(n) \quad (19)$$

$$\frac{de(n)}{dt} = -\frac{dw_{\text{actual}}(n)}{dt} \quad (20)$$

$$\frac{di_{q\text{ref}(n)}}{dt} = \frac{d^2w_{\text{actual}}(n)}{dt^2} \quad (21)$$

$$\begin{aligned} \begin{bmatrix} Y_1^1 \\ Y_2^1 \\ Y_3^1 \\ Y_4^1 \\ Y_5^1 \\ Y_6^1 \end{bmatrix} &= F \left\{ \begin{bmatrix} W_{11}^1 & W_{21}^1 \\ W_{12}^1 & W_{22}^1 \\ W_{13}^1 & W_{23}^1 \\ W_{14}^1 & W_{24}^1 \\ W_{15}^1 & W_{25}^1 \\ W_{16}^1 & W_{26}^1 \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \end{bmatrix} + \begin{bmatrix} b_1^1 \\ b_2^1 \\ b_3^1 \\ b_4^1 \\ b_5^1 \\ b_6^1 \end{bmatrix} \right\} \\ \begin{bmatrix} Y_1^3 \\ Y_2^3 \\ Y_3^3 \end{bmatrix} &= F \left\{ \begin{bmatrix} W_{11}^2 & W_{21}^2 & W_{31}^2 & W_{41}^2 & W_{51}^2 & W_{61}^2 \\ W_{12}^2 & W_{22}^2 & W_{32}^2 & W_{42}^2 & W_{52}^2 & W_{62}^2 \\ W_{13}^2 & W_{23}^2 & W_{33}^2 & W_{43}^2 & W_{53}^2 & W_{63}^2 \end{bmatrix} \begin{bmatrix} Y_1^1 \\ Y_2^1 \\ Y_3^1 \\ Y_4^1 \\ Y_5^1 \\ Y_6^1 \end{bmatrix} + \begin{bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{bmatrix} \right\} \\ \frac{di_q}{dt} &= F \left\{ \begin{bmatrix} W_{11}^3 & W_{21}^3 & W_{31}^3 \end{bmatrix} \begin{bmatrix} Y_1^3 \\ Y_2^3 \\ Y_3^3 \end{bmatrix} + b_1^3 \right\} \end{aligned}$$

Equations (17)–(19) show the mathematical modelling of Fig. 4, where W_{kij} are the weights, $e(t)$ and $de(t)/dt$ are the input to the neural net, and the corresponding output is di_q/dt .

5 Results

The field-oriented control of PMSM using ANN controller is simulated in MATLAB/Simulink. The motor parameters are shown in Table 1.

For each of the three training methods, the parameters selected are as follows:

Table 1 Motor parameters

rs	Stator resistance	1.3 Ω
Ls	Stator inductance	0.009 H
λ	Rotor flux	0.41 wb
J	Motor inertia	0.0027 kg m ²
B	Viscous damping	0.0001 kg/s
P	Number of rotor poles	6

Minimum gradient value = 1e – 7

Maximum no. of epochs = 1000

Time = inf

Goal = 0

Learning rate α = 0.001

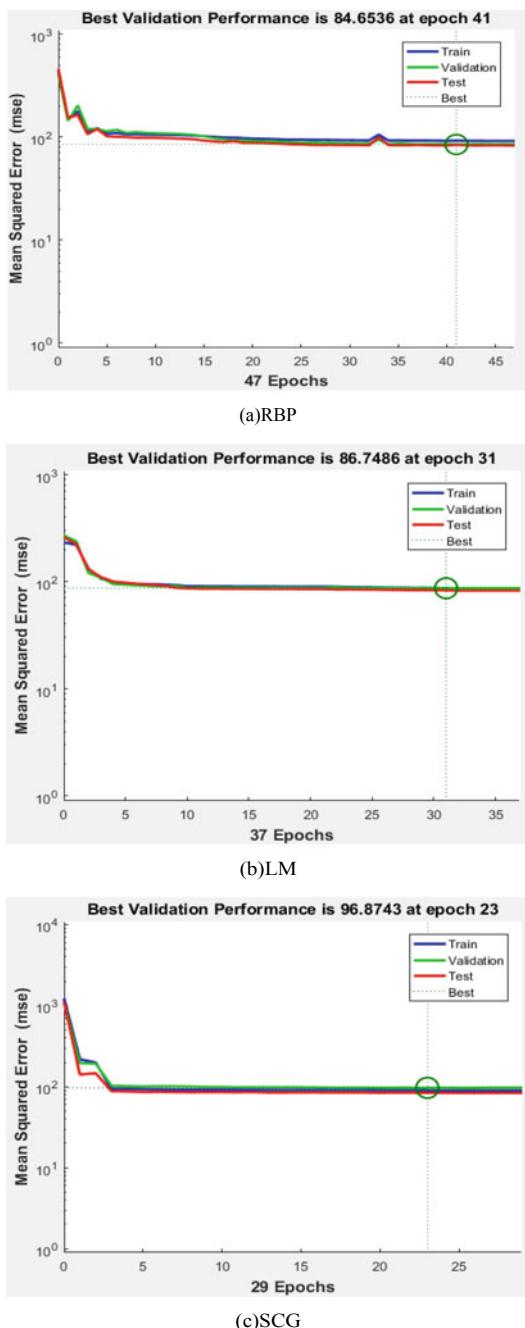
Maximum fails = 6.

Figure 5 shows the training performance plot for RBP, LM and SCG algorithms, respectively. Figure 6 shows steady-state speed responses of PMSM drive system, to step change in speed reference value. The step change of 1500 rpm is applied at 0.5 s. All the three responses settle within 0.05 s with slight variations. Figure 7 shows the systems response to load torque. The rated torque of 8.5 N-m is applied at 0.5 s. The starting torque demanded by all three ANN control training algorithms is approximately equal to each other.

The torque ripples for all control algorithms are same at 6–7%. Figure 8 shows the steady-state current of phase a for RBP, LM and SCG algorithms. The steady-state current drawn by the motor is 4.5 A, which is the rated current of the motor. The transient current drawn is approximately 9 A in case of all algorithms. The integral time absolute error (ITAE) of speed is also taken as the performance comparison parameter. It is seen that ITAE of RBP is the least. SCG gives slightly higher ITAE followed by LM algorithm. After testing the drive for speed reversal, it is seen that the speed reversal operation takes only 0.05 s. (Fig. 10). The drive works satisfactorily for 150% change in stator resistance. For all the training algorithms, it requires 0.1 s to obtain steady-state speed in forward motoring condition, and this is depicted in Fig. 11. The stator current drawn in this condition is shown in Fig. 12; whereas, ITAE is shown in Fig. 13.

A disturbance of $0.01\sin 50t$ is added in input panel to test the robustness of the drive. ITAE for forward motoring condition with disturbance is shown in Fig. 14. The performance comparison is given in Table 2. SCG has the fastest training speed with 29 epochs followed by, LM with 37 epochs and RBP with 47 epochs. Although RBP is the slowest of three, it gives almost zero speed error. SCG has speed error of 6.357 rpm; whereas, LM has the speed error of 8.275 rpm. The steady-state torque error is same for all training algorithms.

Fig. 5 Training performance plot of algorithms



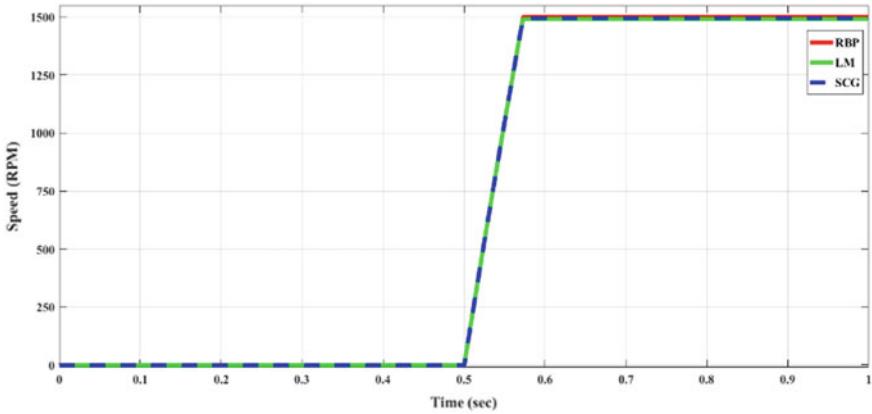


Fig. 6 ANN controller speed outputs for reference of 1500 rpm applied at 0.5 s with full load of 8.5 N m

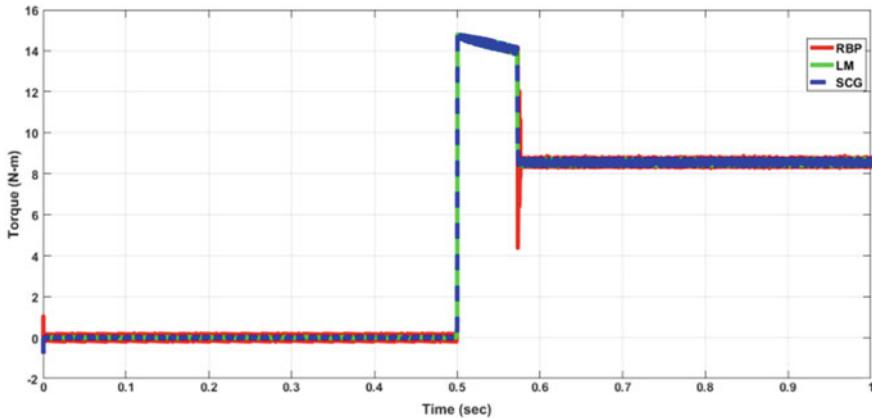


Fig. 7 Torque response for condition in Fig. 6

6 Conclusion

In this paper, three training algorithms for ANN-based FOC of PMSM drive system are compared, and the comparison is presented. Although ANN control technique offers robustness against disturbances and parameter variations, the steady-state speed error depends on the training algorithm. SCG has the fastest training capability with a fair amount steady-state speed error. RBP is the slowest but has almost negligible speed error. LM algorithm has the most error with respect to speed.

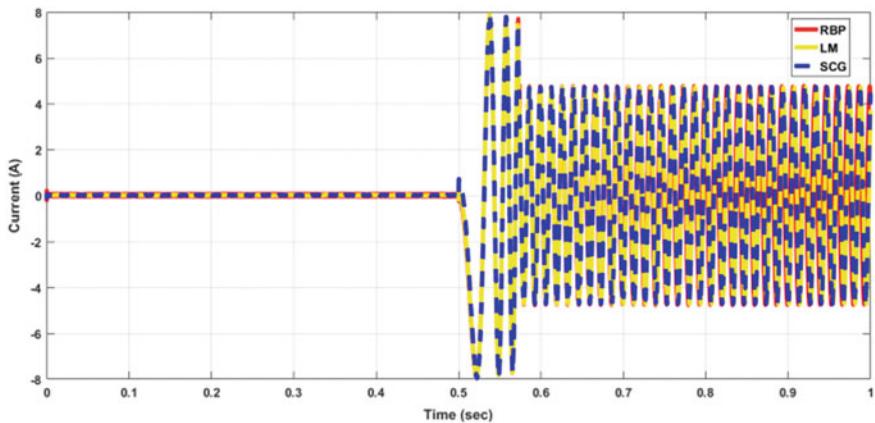


Fig. 8 Stator phase current for condition in Fig. 6

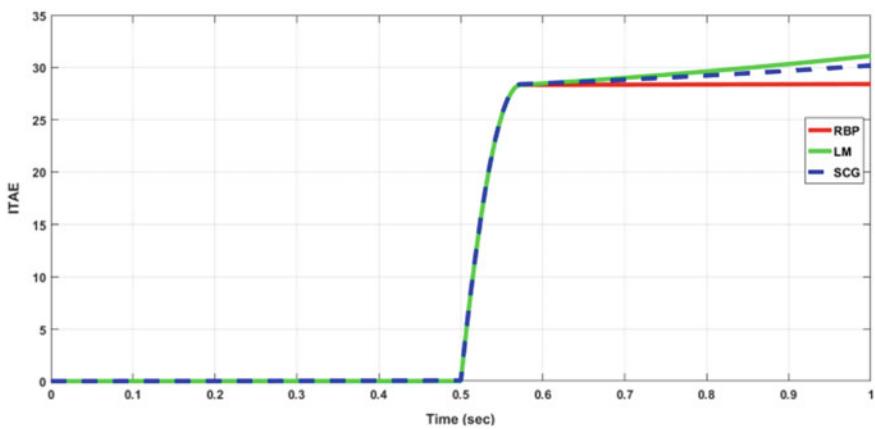


Fig. 9 ITAE of systems for condition in Fig. 6

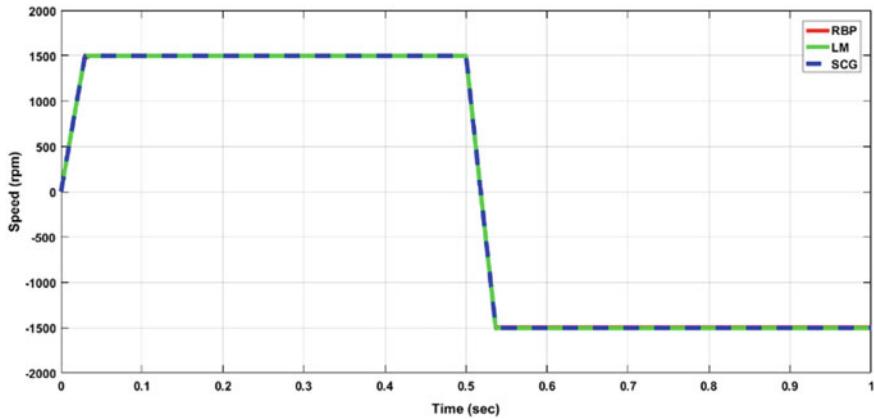


Fig. 10 Speed response of ANN controller for speed reversal

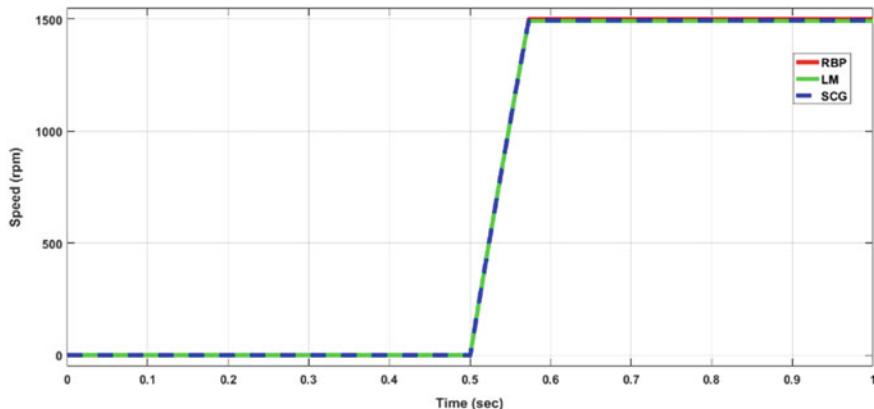


Fig. 11 Speed response when stator resistance is increased by 150%

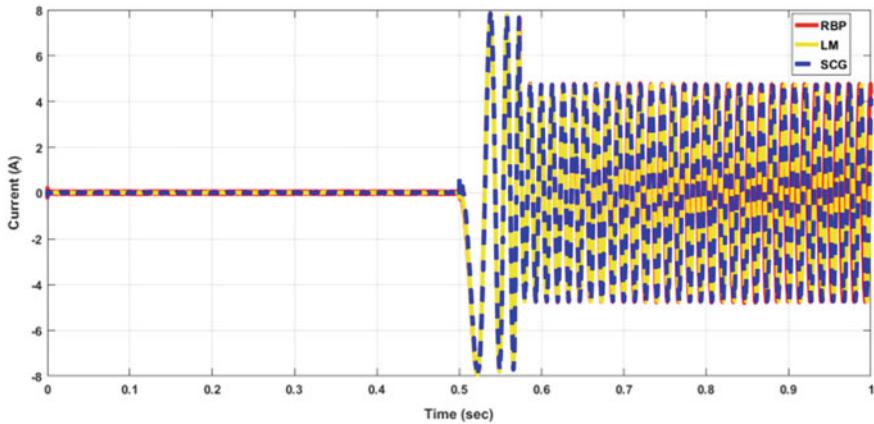


Fig. 12 Phase currents when stator resistance is increased by 150%

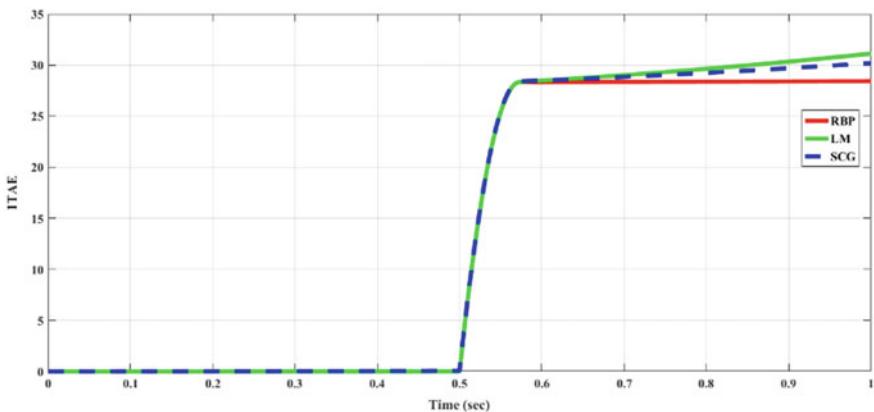


Fig. 13 ITAE of systems when stator resistance is increased by 150%

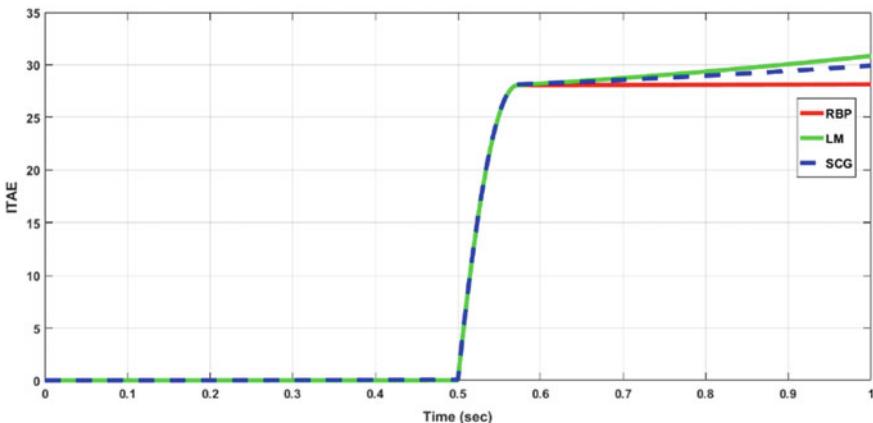


Fig. 14 ITAE of systems when disturbance is added in signal

Table 2 Comparison between training Algorithms

Algorithm	RBP	LM	SCG
Settling time	0.07	0.075	0.072
Full load steady-state speed error	0.21 RPM	8.27 RPM	6.357 RPM
Full load torque error	0.6 N m	0.6 N m	0.6 N m
Epochs	47	37	29

References

1. Slemmon GR (1992) Electric machines and drives. Addison-Wesley Publishing, Co., Inc.,
2. Blaschke F (1972) The principle of field orientation as applied to the NEW transvector closed-loop system for rotating-field machines, Siemens Rev 34(3), pp. 217–220
3. Vas P (1999) Artificial-Intelligence-based electrical machines and drives: application of fuzzy, neural, fuzzy-neural, and genetic-algorithm based techniques. Oxford Science
4. El-Sharkawi MA, AEI-Shy A, El-Sayed ML (1994) High performance drive of DC brushless motors using neural network. IEEE Trans Energy Conv 9(2)
5. Rahman MA, Hoque MA (1998) On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors. IEEE Trans Energy Convers 13(4):311–318
6. Kumar R, Gupta RA, Bansal AK (2007) Novel topologies for identification and control of PMSM using artificial neural network. In: 2007 IET-UK international conference on information and communication technology in electrical sciences (ICTES 2007), Tamil Nadu, 2007, pp 75–80
7. Gaur P, Singh B, Mittal AP (2008) Artificial neural network based controller and speed estimation of permanent magnet synchronous motor. In: 2008 joint international conference on power system technology and IEEE power India conference, New Delhi, 2008, pp 1–6. <https://doi.org/10.1109/ICPST.2008.4745351>
8. Pajchrowski T, Zawirski K (2007) Application of artificial neural network to robust speed control of servodrive. IEEE Trans Ind Electron 54(1)
9. Grzesiak M, Kazmierkowski MP (2007) Improving flux and speed estimators for sensor-less AC drives. IEEE Ind Electron Mag 1(3):8–19

10. Jadhav SV, Chaudhari BN (2013) A novel artificial neural network based space vector modulated DTC and its comparison with other Artificial Intelligence (AI) control techniques. *J Springer Lect Notes Control Syst (Springer LNCS)*
11. Jadhav SV, Srikanth J, Chaudhari BN (2010) Intelligent controllers applied to SVM-DTC based induction motor drives: a comparative study. *Intelligent controllers applied to SVM-DTC based induction motor drives: a comparative study, 2010 Joint international conference on power electronics, Drives and Energy Systems and 2010 Power India*. New Delhi 2010:1–8
12. Jadhav SV, Kirankumar J, Chaudhari BN (2012) ANN based intelligent control of induction motor drive with space vector modulated DTC. In: 2012 IEEE international conference on power electronics, drives and energy systems (PEDES), Bengaluru, pp 1–6
13. Maiti S, Verma V, Chakraborty C, Hori Y (2013) An adaptive speed sensor-less induction motor drive with artificial neural network for stability enhancement. *IEEE Trans Ind Informat* 8(4):757–766
14. Flieller D et al (2014) A self-learning solution for torque ripple reduction for non-sinusoidal permanent magnet motor drives based on artificial neural networks. *IEEE Trans Ind Electron* 61(2):655–666
15. Pajchrowski T, Zawirski K, Nowopolski K (2015) Neural speed controller trained online by means of modified RPROP algorithm. *IEEE Trans Ind Inf* 11(2)
16. Hagan MT (1994) Menhaj MB (1994) Training feed forward networks with the Marquardt algorithm. *IEEE Trans Neural Netw* 5(6):989–993
17. Babani L, Jadhav S, Chaudhari B (2016, September) Scaled conjugate gradient based adaptive ANN control for SVM-DTC induction motor drive. In: 12th IFIP international conference on artificial intelligence applications and innovations (AIAI), September 2016, Thessaloniki, Greece, pp 384–395
18. Moller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw* 6:525–533

Chapter 37

An Analysis and Comparison of Community Detection Algorithms in Online Social Networks



Sanjeev Dhawan, Kulvinder Singh, and Amit Batra

1 Introduction

Community architecture and design are exhibited by several networks which are existing in today's world like social nets, neural nets, and so forth. The term community is a very important aspect of these networks since it can help us to recognize the hidden configuration of the network. Community is regarded as cluster of vertices that are associated with one another. Over the last many years, community detection has appeared as a milestone in the arena of social network investigation. The community discovery is performed in such a way that entities or vertices pertaining to a specific community are very alike, analogous, related, and similar while they are disparate from vertices or entities that belong to the other communities. The networks which present a community arrangement may sometimes show a hierarchical community arrangement also [1]. This paper shows the analysis of social networks [2]. Social graphs obey the characteristics of large online complex networks [3, 4]. McPherson et al. [5] observed that "similarity breeds connection." Enhancing fuzzy C-mean centered cluster discovery has been done in social nets employing dynamic parallelism by Al-Ayyoub et al. [6]. A novel trust-centered cluster discovery technique in social nets has been suggested by Chen et al. [7]. A new scalable leader-cluster discovery technique for cluster discovery in social nets has been employed by Ahajam et al. [8]. User interest community detection on social media has been done employing collaborative filtering by Jiang et al. [9]. The community detection

S. Dhawan · K. Singh · A. Batra (✉)

Department of Computer Science and Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

S. Dhawan

e-mail: sdhawan2015@kuk.ac.in

K. Singh

e-mail: ksingh2015@kuk.ac.in

approach can be used in ad hoc networks to generate routing tables. Furthermore, the approach can also be employed in visualizing complex graphs.

2 Related Work and Literature Survey

Plenty of work has been carried out in the past pertinent to community detection. Emotional cluster discovery in social nets was carried by Kanavos et al. [10]. An incremental technique to discover clusters in dynamic evolving social nets was suggested by Zhao et al. [11]. Zheng et al. [12] employed privacy-preserved cluster detection in online social nets. A three-stage algorithm discovers without any knowledge about number of these communities [13]. An ant-based label propagation approach for community discovery in social networks has been suggested [14]. Such vast information can serve a useful purpose in different fields like psychology, sociology, biology, and several other fields of science. A detailed analysis and comparison has been given in the forthcoming sections.

3 Community Discovery Algorithms

3.1 Community Discovery Utilizing Label Propagation

Label propagation in a net can be depicted as the proliferation of a label to several vertices present in the graph. Label propagation technique is a category of probing technique and the major probing guidelines of LPA is to move or propagate information pertaining to label invariably within vertices. The central and vital concept of LPA is that every vertex modifies its label to the one possessed by the larger number of its adjoining or neighboring vertices (as depicted in Fig. 1).

Assume a network $G = (V, E)$ as input and maxI as the maximum no. of iterations.

Output: Community set $p = \{p_1, p_2, \dots, p_j\}$, j denotes how many communities are there.

- Step 1 Allocate each single vertex in the graph to a specific label, e.g., for a given vertex x , $px(0) = x$.
- Step 2 $r = 1$ will be considered as iteration number.
- Step 3 The vertices are arranged in some arbitrary order and set up an arranged progression $X = \{x_1, x_2, \dots, x_n\}$.
- Step 4 For every vertex $x \in X$, the vertex label is updated iteratively as per Eq. (5)

$$px(t) = f(px_1(r-1), \dots, px_j(r-1)), xi \in N(x) \quad (1)$$

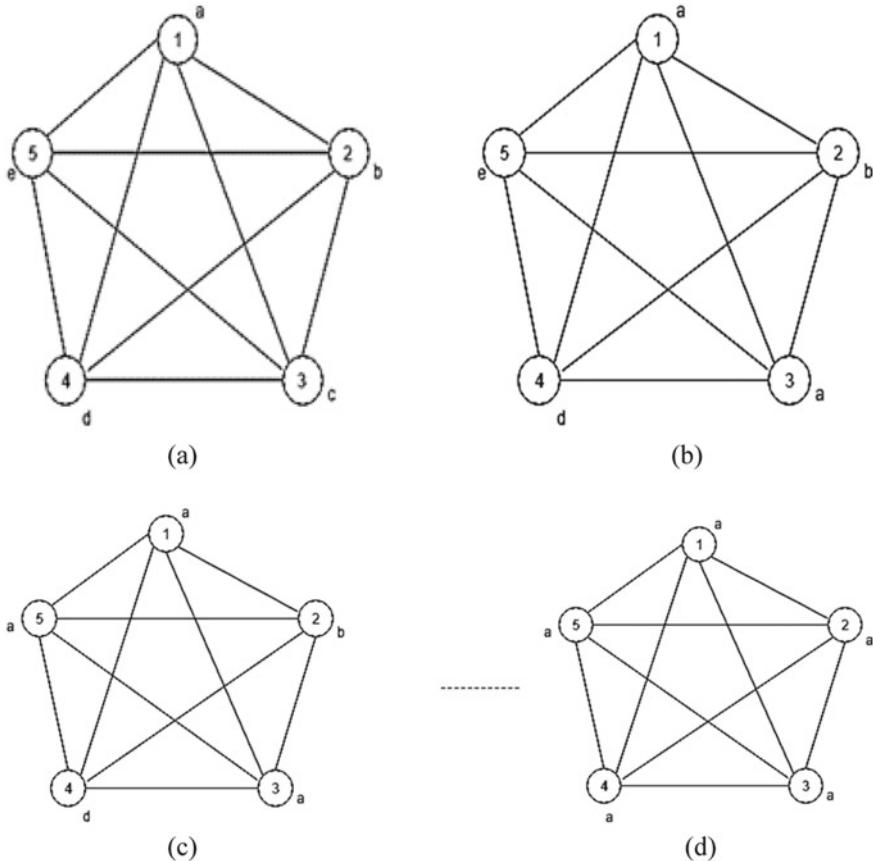


Fig. 1 Label propagation

$$px(t) = f(px_1(r-1), \dots, px_m(r-1), px(m+1)(r), \dots, px_j(r)), xi \in N(x) \quad (2)$$

As a result, a label propagation algorithm can take into consideration early vertex significance algorithm, and the blueprint can be depicted as given below:

$$\text{NS}(k) = \text{Ing}(k) + \alpha \sum_{i \in N(k)} \frac{\text{Ing}(i)}{d(i)} \quad (3)$$

where $\text{NS}(k)$ means the significance of node k , $\text{Ing}(k)$ means the earlier significance of node k , α is the parameter which varies within range 0–1, which tells us the intensity of the effect of neighboring or adjoining vertices on vertex k , $N(k)$ means the neighborhood or adjoined set of vertex k , and $d(i)$ may be depicted as the degree

of the vertex i . The blue print can be depicted as given below:

$$\text{LS}(k, l) = \sum_{i \in N^l(k)} \frac{\text{NS}(i)}{d(i)} \quad (4)$$

where $\text{LS}(k, l)$ depicts the significance of label l on the node k , and $N^l(k)$ means the set of label l around the node k . As a matter of fact, formula can be updated for the new label and can be written as given below:

$$c_k = \arg \max_{l \in \text{maximum}} \text{LS}(k, l) \quad (5)$$

where c_k depicts the most significant and dominant label, and Imaximum means the set of the greatest number of labels. LPA was suggested by Raghavan et al. [15], where firstly every vertex goes to obtain a label from the largest count of labels acquired by its adjacent nodes. Speaker–listener LPA [16] (SLPA) is an addition to LPA that could examine diverse categories of clusters like overlapping clusters, disjoint clusters, and hierarchical clusters in both bipartite and unipartite graphs. Centered on the SLPA method, Hu [17] suggested a weighted LPA (WLPA). LPA was moreover enriched by Gregory [18] in his technique “community overlap propagation algorithm (COPRA).” LabelRank algorithm [19] employs the Markov clustering algorithm and LPA. The “LabelRank” method was altered to LabelRankT approach by Xie et al. [20]. Wu et al. [21] suggested a “balanced multi-label propagation algorithm (BMPLA)” for the discovery of clusters which are overlapped (Table 1).

Table 1 Community discovery using label propagation

Proposed algorithm	Strategy	Utilization	Criterion
Wu et al. (BMLPA) [21]	Label proliferation, the concept of overlapping communities		How many vertices are there, vertices belong to which specific labels, and the average degree
Raghavan et al. (LPA) [15], motif aware label propagation [22]	Iterative label proliferation	Speaker–listener LPA [16] Weighted LPA [17] community overlap PA [18] LabelRank [19] BMPLA [21]	Nodes, labels, threshold mark, similarity vertices
Xie et al. (LabelRank) [19]	Proliferation, expansion, cut-off, conditional update	LabelRankT [20]	Coefficient of belonging, threshold vertices

3.2 *Community Discovery Centered on Semantic*

LDA [23] is employed in various semantic cluster-centered cluster discovery methods. In another task, the writers [24] have employed “author–recipient topics (ARTs)” prototype and partitioned the task amid dual stages, specifically, “LDA” clusters discovery and sampling. Xia et al. [25] created a semantic graph employing data from the subject of the comment mined from the primary HTML source archives. Ding [26] has taken into consideration the effect of topological and topical components in community discovery. A community discovery technique, “semantic tag propagation (SemTagP),” has been suggested by Ereto et al. [27] that takes into consideration output of the semantic data held during formation of the “resource description framework (RDF)” networks of community graphs. It actually is an advancement of the “LPA” [15] method that implements the “semantic” proliferation of labels. In an investigation by Zhao et al. [28], a issue-centered method comprising of an blend of communal entities cluster prediction was employed. A community abstraction technique is provided by Abdelbary et al. [29] that assimilates the subject printed within the communal graph along its “semantic” structures. “Latent semantic analysis (LSA) [30] and latent dirichlet allocation (LDA)” [23] are the dual methods widely used in the procedure to discover topical clusters. Nyugen et al. [31] have employed LDA to discover hyperactive clusters in the subject of the blog, and then, “sentiment” investigation is performed to discover the meta-clusters in these elements. A “link-content” prototype is suggested by Natarajan et al. [32] to detect topic-centered clusters in communal graphs.

4 Cluster Discovery Techniques

A latest review by Amelio et al. [33] provides a broad survey of main overlapped clustering discovery techniques, and where, they have likewise involved a group of “dynamic” graphs-centered overlapped cluster discovery. Additional work containing comprehensive survey of techniques for detecting overlapping clusters is performed by Xie et al. [34].

4.1 *Overlapping Community Detection Using Clique-Based Techniques*

We can define an i-clique as a sub-network where every pair of nodes is associated containing j vertices. We can describe a j-clique cluster as consisting of all i-cliques that can be reached from one another via a chain of adjoining j-cliques. Palla et al. [35] suggested a “clique percolation method (CPM)” to ascertain a category of clusters also called as overlapping communities. CPM firstly obtains entire cliques of the

graph and utilizes the strategy proposed by Everett et al. [36] to discover communities through module examination of clique–clique overlap matrix. CPM takes $O(\exp(n))$ time to execute. The CPM that was proposed by Palla et al. [35] was not found to be able to determine the hierarchical organization alongside overlapping factor. This drawback was overthrown by algorithm suggested by Lancichinetti et al. [37]. This algorithm carries out local exploration so as to discover the cluster for all the vertices. In this task, the vertices may be visited again and again up to infinite number of times. The foremost aim of this algorithm was to achieve local maximum on the basis of a fitness function. A tool called CFinder [38] was created by using clique percolation method for discovering the overlapping communities. There are some applications of community detection, viz. generating recommendations for E-commerce [39–41].

5 Conclusion and Future Research Directions

In this manuscript, a good attempt has been made towards comparing and contrasting several community detection algorithms which group similar items and thus leads to good quality community detection. These algorithms and approaches can be utilized to discover clusters in real-world social networks of Facebook, Twitter, Instagram, and LinkedIn and is capable of supplying large amount of information which can be applied for several tasks. These community detection approaches can be employed in diverse branches of science like biology, physics, etc. For example, communities in biological networks can assist in making us understand different methods which govern cellular processes. These community detection algorithms can be used in conjunction with deep learning techniques to build robust recommender systems.

References

1. Ozturk K (2014) Community detection in social networks. Graduate School of Natural and Applied Sciences, Middle East Technical University
2. Tang L, Liu H (2010) Community detection and mining in social media 2(1)
3. Fasmer EE (2015) Community detection in social networks. Department of Informatics, University of Bergen
4. Barabási ARA-L (1999) Emergence of scaling in random networks. *Science* 286(80):509–512
5. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. *Annu Rev Sociol* 27(1):415–444
6. Al-Ayyoub M, Al-andoli M, Jararweh Y, Smadi M, Gupta B (2019) Improving fuzzy C-mean-based community detection in social networks using dynamic parallelism. *Comput Electr Eng* 74:533–546
7. Chen X, Xia C, Wang J (2018) A novel trust-based community detection algorithm used in social networks. *Chaos Solitons Fractals* 108:57–65
8. Ahajjam S, El Haddad M, Badir H (2018) A new scalable leader-community detection approach for community detection in social networks. *Soc Netw* 54:41–49
9. Jiang L, Shi L, Liu L, Yao J, Yousuf MA (2019) User interest community detection on social media using collaborative filtering. *Wirel Netw* 4

10. Kanavos A, Perikos I, Hatzilygeroudis I, Tsakalidis A (2018) Emotional community detection in social networks. *Comput Electr Eng* 65:449–460
11. Zhao Z, Li C, Zhang X, Chiclana F, Viedma E H (2019) An incremental method to detect communities in dynamic evolving social networks. *Knowl-Based Syst* 163:404–415
12. Zheng X, Cai Z, Luo G, Tian L, Bai X (2019) Privacy-preserved community discovery in online social networks. *Futur Gener Comput Syst* 93:1002–1009
13. You X, Ma Y, Liu Z (2020) A three-stage algorithm on community detection in social networks. *Knowl-Based Syst* 187:104822
14. Razieh H, Alireza R (2020) AntLP: ant-based label propagation algorithm for community detection in social networks. *CAAI Trans Intell Technol* 5(1):34–41
15. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E Stat Nonlinear Soft Matter Phys* 76(3):1–11
16. Xie J, Szymanski BK (2012) Towards linear time overlapping community detection in social networks. *Lecture Notes Computer Science (including Subser. Lecture Notes Artificial Intelligent Lecture Notes Bioinformatics)*, vol 7301 LNAI, No. PART 2, pp 25–36
17. Hu W (2013) Finding statistically significant communities in networks with weighted label propagation. *Soc Netw* 02(03):138–146
18. Gregory S (2010) Finding overlapping communities in networks by label propagation. *New J Phys* 12
19. Xie J, Szymanski BK (2013) LabelRank: a stabilized label propagation algorithm for community detection in networks. In: *Proceedings of 2013 IEEE 2nd international network science workshop*. NSW 2013, pp 138–143
20. Xie J, Chen M, Szymanski B K (2013) LabelRankT: incremental community detection in dynamic networks via label propagation. In: *Proceedings of workshop dynamic networks management mining*, DyNetMM 2013, pp 25–32
21. Wu ZH, Lin YF, Gregory S, Wan HY, Tian SF (2012) Balanced multi-label propagation for overlapping community detection in social networks. *J Comput Sci Technol* 27(3):468–479
22. Li PZ, Huang L, Wang CD, Lai JH, Huang D (2020) Community detection by motif-aware label propagation. *ACM Trans Knowl Discov Data* 14(2):1–19
23. Blei JMDM, Ng AY (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022
24. Xin Y, Yang J, Xie ZQ, Zhang JP (2015) An overlapping semantic community detection algorithm base on the ARTs multiple sampling models. *Expert Syst Appl* 42(7):3420–3432
25. Xia Z, Bu Z (2012) Community detection based on a semantic network. *Knowl-Based Syst* 26:30–39
26. Ding Y (2011) Community detection: topological versus topical. *J Inform* 5(4):498–514
27. Erétéo G, Gandon F, Buffa M (2011) SemTagP: semantic community detection in folksonomies. In: *Proceedings of 2011 IEEE/WIC/ACM international conference on web intelligent WI 2011*, vol 1, pp 324–331
28. Zhao Z, Feng S, Wang Q, Huang JZ, Williams GJ, Fan J (2012) Topic oriented community detection through social objects and link analysis in social networks. *Knowl-Based Syst* 26:164–173
29. Abdelbary H, El-Korany A (2013) Semantic topics modeling approach for community detection. *Int J Comput Appl* 81(6):50–58
30. Deerwester HRSC, Dumais ST, Landauer TK, Furnas GW (1990) Indexing by latent semantic analysis. *J Am Soc Inf Sci* 41:391–407
31. Nguyen T, Phung D, Adams B, Tran T, and Venkatesh S (2010) Hyper-community detection in the blogosphere. In: *WSM'10—proceedings of 2nd ACM SIGMM workshop social media, co-located with ACM multimedia*, 2010, pp 21–26
32. Natarajan N, Sen P, Chaoji V (2013) Community detection in content-sharing social networks. In: *Proceedings of 2013 IEEE/ACM international conference on advanced social networks analysis mining*, ASONAM 2013, pp 82–89
33. Amelio PCA (2014) Overlapping community discovery methods: a survey. In: *Social networks analysis case study*, Lecture notes social networks. Weinheim Springer-Verlag, pp 105–125

34. Xie J, Kelley S, Szymanski BK (2013) Overlapping community detection in networks: the state-of-the-art and comparative study. *ACM Comput Surv* 45(4)
35. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435(7043):814–818
36. Everett M, Borgatti S (1998) Analyzing clique overlap. *Connections* 21(1):49–61
37. Lancichinetti A, Fortunato S, Kertész J (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New J Phys* 11
38. Adamcsek B, Palla G, Farkas IJ, Derényi I, Vicsek T (2006) CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics* 22(8):1021–1023
39. Hooda R, Singh K, Dhawan S (2014) Social commerce hybrid product recommender. *Int J Comput Appl* 100(12):43–49
40. Hooda R, Singh K, Dhawan S (2014) Social commerce hybrid product recommender. *Int J Comput Appl* 97(4):23–28
41. Dhawan S, Jyoti SK (2015) Comparison of various similarity measure techniques for generating recommendations for E-commerce sites and social websites. *Am Int J Res Sci Technol Eng Math* 11(2):219–221

Chapter 38

A Deep Learning Approach for Network Intrusion Detection Using Non-symmetric Auto-encoder



Divya Nehra, Veenu Mangat, and Krishan Kumar

1 Introduction

Computer networks are overwhelmingly plagued by security threats like denial of service, spoofing of user identity and privacy breaching. The need for a robust and effective IDS arises in order to deal with heavy incoming traffic which may have illegitimate or malicious packets. Mostly, IDSs are distributed into network-based IDSs (NIDSs), host-based IDSs (HIDSs) and hybrid IDSs. The NIDS is placed inside a network to analyse the real-time traffic in order to find out the intrusions by scrutinizing all packets moving through the network. On the other hand, the HIDS which is kept on a specific system of the network is used to examine all inward and outward traffic. Sometimes, NIDS becomes overloaded by network packets and might slip some nefarious packets, whereas HIDSs have less workload and require more resources. One of the type of IDS is the hybrid IDS that monitors specific clients as well as network activities. An intrusion prevention system (IPS) is similar to IDS but IPS can be configured to block potential threats, whereas IDS only raises alarms about the suspicious activity. To protect the networks from internal as well as external intruders, the deployment of NIDS is necessary. External intruders are those who do not belong to the network domain like in DoS and attempt break-in attacks, whereas internal intruders are those who stay in the network and sniff the confidential information like masquerading and misuse of privileges. In [1], the analysis of several machine learning techniques has been conducted and endorses the necessity of constructing capable intrusion detection systems. In [2], study of artificial neural network (ANN), decision tree, support vector machine (SVM), Bayesian networks and a self-organizing map have been done. The results achieved using machine learning-based techniques are promising, but their performance is affected due to various factors, for example misclassification of network data. In [3], a pair of

D. Nehra (✉) · V. Mangat · K. Kumar

University Institute of Engineering and Technology, Panjab University, Chandigarh, India

simulated annealing and support vector machines have been used for the detection of nefarious behaviour. The work has concluded that no machine learning algorithms are best suited or perfect for all intrusions. Machine learning techniques have to be used in an alliance with other techniques. One more factor that needs to be highlighted is regarding the big dimensionality of network flow. To summarize, the main limitations of conventional machine learning techniques are as follows: a large number of training data are required for effective learning, whereas most of the benchmark data sets available suffer from the curse of data skewness or imbalancing. Moreover, to work with big data, efficient feature learning methods are required and to achieve efficient accuracy levels with high-dimensional and unlabelled data, prominent time reducing techniques are needed. The aforementioned factors need to be dealt with, and deep learning methods can be smeared for achieving better results.

Deep learning is an innovative subcategory of machine-based learning. Presently, the applications of shallow and machine-based learning within NIDS have shown subsequent improvements in detection accuracy [4]. The deep learning techniques have gained a lot of attention from researchers as they are capable of efficient learning to find anomalies.

In this paper, the capabilities of auto-encoders have been explored. The model has multiple encoders and a single decoder with two asymmetric auto-encoders which are stacked to give a compressed output. The proposed work is a collaboration of deep and shallow learning. The proposed non-symmetric deep auto-encoder is a combination of stacked auto-encoders with the power of random forest.

Following are some innovative contributions of the paper:

- (1) We engaged a deep auto-encoder for feature extraction from a skewed data set and produced a model to identify a normal event and a nefarious one.
- (2) Proposed model makes use of stacked auto-encoders in pipeline to random forest classifier. The combined power of deep as well as shallow learning is explored to achieve significantly good results.
- (3) The performance evaluation of the suggested model is assessed using KDD Cup'99 data set. Binary class and five-class classification of attacks has been done. Performance metrics include false alarm rate, detection rate, and accuracy.

The paper is structured as follows. Section 2 offers correlated background material. Section 3 presents the interrelated work. Section 4 gives the methodology used in suggested approach. Section 5 presents the assessment of the proposed work. In Sect. 7, conclusion and future work have been discussed.

2 Background

This section provides essential background information about the proposed work.

2.1 Deep Learning

Nowadays, deep learning practices are gaining the interest of researchers. Various domains such as computer vision, image processing, and intrusion detection have successful applications of these techniques. Various reasons for wide adoption of deep learning require fewer efforts for feature engineering, flexible adaptation to novel problems, and ability to work on big data.

2.2 Auto-encoder

An auto-encoder is a category of artificial neural networks that reduces the input into a latent-space representation, and output is reconstructed using this representation. The targeted output of the network is generally set equal to its input. Moreover, learning is done using efficient data coding in an unsupervised manner. The auto-encoders are used mainly for dimensionality reduction for a set of data [5]. The two chief hands-on applications of auto-encoders are data denoising and feature reduction for data visualization. The features selected by auto-encoder are further used as inputs for the classification problem. Figure 1 shows the schematic set-up of a single auto-encoder. An auto-encoder encompasses both decoder and encoder. The encoder works for compressing the input data by extracting the features and making it low dimensionality. The decoder is employed to reconstruct input from the low-dimensional latent data. Moreover, an auto-encoder learns a low-dimensional representation of input using a hidden layer and then decodes it into an output layer [6]. Learning is done in an unsupervised manner mainly for dimensionality reduction. One of the most powerful characteristics of auto-encoder is its learning algorithm that makes it feasible to use back propagation using gradient descent methods for multilayer networks [7]. The weight updation using gradient descent is given by Eq. (1) below:

Fig. 1 Schematic representation of a single auto-encoder [8]

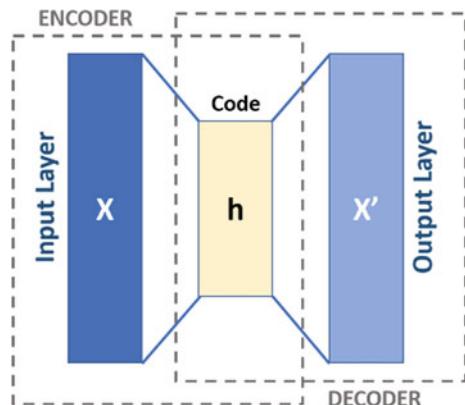
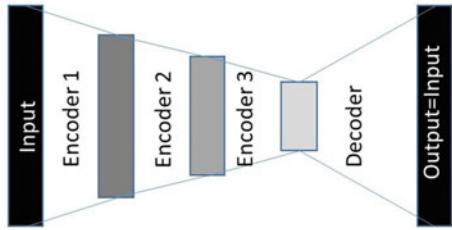


Fig. 2 Schematic arrangement of stacked deep auto-encoder [10]



$$w_{t+1} = w_t - \eta \frac{\partial c}{\partial w} \quad (1)$$

where η represents the learning rate, and $\frac{\partial c}{\partial w}$ is the derivative of cost function.

2.3 Stacked Auto-encoder (SAE)

These auto-encoders consist of two symmetrical auto-encoders. The stacking in these auto-encoders provides the layer wise learning which helps the model to acquire the dependency amid the features. They are composed of connected multiple layers of hidden units and characteristically take four or five shallow layers of encoding with the second set comprising of four or five layers of decoding. When multiple hidden layers are used to provide depth, such scenario is termed as stacked auto-encoder. This paper introduces NDAE-IDS which is a non-symmetric auto-encoder. Non-symmetric schema comprises of variable encoders and decoders and built by nesting one into the other. It was found in [9] that while multiple encoders aided in extracting the input signal, multiple decoders did not help the cause and further lead to the reconstruction error—each layer of the decoder adding a bit more to it. A graphic example of a stacked auto-encoder is shown in Fig. 2. Therefore, multiple decoders were discarded, and we built the model using asymmetric auto-encoders with multiple encoders but only one decoder.

3 Existing Work

Several works were proposed works like [11] implementing self-taught learning-based IDS by combining stacked AE and SVM for intrusion detection in binary and multi-class classification on NSL-KDD, and the researchers concluded that deep learning-based approach is able to attain good detection precision as well as accuracy. In [12], self-taught learning in association with softmax regression is used to develop a NIDS. A generative model of deep learning is chosen to deal with class imbalancing in data set. Their exertion determined that deep learning-based approaches give more

promising results than conventional methods. In [6], sparse auto-encoder for unsupervised feature learning is used with a softmax regression over NSL-KDD, and it has been reported that this method has achieved accuracy of 88.39% and recall of 95.95%. The results are claimed as improvement over machine learning methods. In [13], it is an application of unsupervised stacked deep auto-encoder (SDAE) for extraction of features from raw data. The multi-class classification has been performed. It has been summarized that deep learning is desirable for future defences in network-based intrusion detection. In [9], a novel classifier has been proposed using stacked AE in the pipeline to the random forest classification algorithm. Multi-class classification is done with the help of 13-class as well as five-class classification of attacks. The curse of dimensionality is very well handled with the help of deep learning method. Many scholars have discussed the concerns of NIDSs in attaining low false positive rate (FPR), high detection rate (DR), and accuracy in classification. Deep learning has been the best-suited algorithms to deal with above-mentioned issues related to intrusion detection. The effective deep learning-based algorithm is non-symmetric deep auto-encoder (NDAE) with improved DR and FPR [9]. Several modifications of NDAE have been proposed in [6, 11, 14]. Shone et al. [9] report the challenges of former versions of NDAE like dense volume of data, diversity in customized protocols, imbalance in data sets, and adaptability to the dynamic technologies. It works over the trained random forest (RF) model via the encoded data processed by the stacked NDAEs to classify network traffic into attack and non-attack data. Taking into consideration various advantages of auto-encoders, the development of stacked NDAE-IDS has been done for achieving better detection of intrusion.

4 Proposed Methodology

This section describes in what manner the network intrusion detection problem of detecting attacks is addressed via deep auto-encoder. The suggested NDAE-IDS takes the input vector and plots it to the latent representations. For a certain data set $X = \{x_1, x_2, \dots, x_a\}$ with a samples, where x_i is a d-dimensional feature vector, the encoder is a function that maps an input vector X to a hidden representation z as shown in Eq. (2):

$$z = Sf(W.X + bf) \quad (2)$$

where Sf is a nonlinear activation function. W (weighting) and b (bias) are the hyper-parameters. Rectified linear unit (ReLU) is used as an activation function because of its ability to work on multilayer structure. It overcomes the fading gradient problem and allows the models to learn quicker and with enhanced performance. The equation for ReLU is as follows:

$$f(x) = \max(0, x) \quad (3)$$

where x is the input to the neuron.

The decoder is also a function similar to encoder and maps the hidden representation z back to a reconstruction x' . It is represented by Eq. (4):

$$x' = Sg(V.z + bg) \quad (4)$$

where Sg is activation function for the decoder, and it is same as used for encoder. W and V have been used as weight matrix, and bf and bg are used as bias vectors for encoder and decoder, respectively. The purpose of training the auto-encoder is to lessen the variance between input and output which is computed as loss function. Cross-entropy loss function has been used which calculates the number of ‘bits’ preserved in the reconstruction as compared to the original input. It is the typically used loss function for auto-encoders. Cross-entropy loss is calculated by Eq. (5):

$$L(x, x') = \Sigma dk[x_k \log x'_{k'} + (1 - x_k) \log(1 - x'_{k'})] \quad (5)$$

where x and x' are same as defined earlier.

This deep auto-encoder works in two segments: training and testing. During the training segment, the training data set is used by the system to create a model centred on the suggested deep auto-encoder model. Then the trained model is tested with the test data set to assess the performance of the model. Figure 3 shows the configuration of proposed model. HL stands for hidden layer.

As per the structure, the input layer of the proposed NDAE-IDS model takes all the 41 features of 10% of KDD-Cup’99 data set as input and the first hidden layer of first AE selects the 12 features of 41 features. Three hidden layers have been used where the function of each layer is to provide input for hidden layer $h + 1$ which comes from the output of hidden layer h . Tenfold cross-validation of data set has helped in attaining most optimized values for hidden layers and number of neurons, thus overcoming the risk of overfitting as well.

After layer wise training of auto-encoder, the output of the last hidden layer of encoder is served as input to classifier layer which classifies the attacks into five

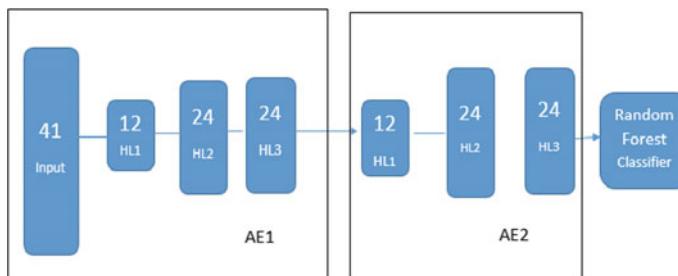


Fig. 3 NDAE-IDS classification model

classes of KDD Cup'99 data set. This work is proposed in collaboration with deep learning with a shallow learning classifier. For shallow learning, random forest (RF) classifier has been used. Random forests are the ensemble-based classifiers used for classification and regression. RF provides the solution for overfitting problem of decision trees. Various researchers such as [15, 16] have concluded that RF algorithm works best with intrusion detection. There are many more examples like [17, 18] of utilizing RF for intrusion detection domain. The proposed model uses two NDAE organized in a stack wise manner in the pipeline to random forest classifier. Each NDAE comprises of three hidden layers with the same number of neurons in each layer. To reduce the risk of overfitting, tenfold CV has been done on KDD Cup'99 data set using scikit-learn.

5 Performance Evaluation

- A. *Data set:* The proposed experiment is evaluated using KDD-CUP'99 data set which is one of the most widely used datasets in the domain of intrusion detection [19]. The literature available on this data set is really vast which convinced us to work on it. The 10% of an original data set has been used which consists of 494,021 instances. During the testing phase, the trained model is tested against test data set which consists of 18,794 instances. This data set comprises a total of 41 features out of which three are categorical features and rest are numerical features. All the instances are labelled as a particular attack class. This proposed work has shown results in two set-ups. One is for binary class classification, and another is for multi-class classification to classify the attack as normal or a specific class of attack (Table 1). Following are the classes of attack:
 - Denial of service (DoS): This type of attack makes a computing resource too much overloaded in such a way that the resource is unable to provide the services to legitimate users. Various examples of such attacks are as follows: Apache, Smurf, Neptune, Ping of death and UDP storm, etc. [20].
 - Probe: In this type of attack, the attacker scans the system or networking device to find the vulnerabilities or the loopholes to exploit the system. Examples are ipsweep, mscan, nmap, saint and portsweep [19].
 - U2R: In this type of attack, the attacker tries to gain access to the root account from a normal user account and exploits the system, e.g. perl and Xterm.
 - R2L: In such attacks, the attacker tries to gain the access of system as a local user by sending packets to the system on which it has no accounts, e.g. FTP-write, guess-password, multihop, spy and phf [21].

The algorithm has been implemented using TensorFlow. All the evaluations were performed using GPU-enabled TensorFlow running on a 64-bit Windows 10 Dell Precision tower with an Intel Core i5-8300H Processor, 16 GB RAM. The performance of NDAE-IDS is assessed through the following metrics:

Table 1 Summary of training and test data set

Dataset	Normal	Probe	U2R	R2L	DoS	Total
Training	97,278	4107	52	1126	3,91,458	494,021
Test	9711	1106	37	2199	5741	18,794

$$\text{Detection Rate (DR)} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (6)$$

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{(\text{FP} + \text{TN})} \quad (7)$$

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{(\text{TN} + \text{TP} + \text{FN} + \text{FP})} \quad (8)$$

where TP is true positive, which shows the attacks which are rightly identified. TN is true negative, which shows the normal events identified as normal. FP is false positive, which shows the normal events identified as attacks. FN is false negative, which shows the attacks identified as normal traffic. Detection rate signifies the ability to detect actual attacks. It is also known as sensitivity, recall, hit rate and true positive rate. False alarm rate is the probability of false alarm, whereas accuracy gives the nearness of the result to a specific value.

6 Experimental Results

- A. Used hyperparameters: The first phase involves training of the auto-encoder. Following are the hyperparameters used for stacked AE: number of neurons $n = 24$, number of hidden layers = 3, learning rate = 0.001. Large range is chosen so that optimum value is not missed. The values chosen are batch_size = 350 and epochs = 50.
Batch_size defines the number of samples to work through before updating the internal model parameters. Mini batch gradient descent is used for learning. Epochs are the number of times the training data set will be passed through the neural network.
- B. Two-class and five-class classification
Table 2 depicts the values for different performance evaluation metrics using proposed NDAE-IDS model for two scenarios, viz. two-class and five-class

Table 2 False alarm rate, detection rate and accuracy by NDAE-IDS for two suggested set-ups

Set-up	FAR (%)	DR (%)	Acc (%)
Two-class classification	0.38	96.54	96.54
Five-class classification	0.44	95.45	95.45

classification. The results show that NDAE-IDS has obtained 96.54% and 95.45%, respectively, for the two-class and five-class scenarios. The FAR attained for both the scenarios remains consistently low, whereas detection rate is promising. The effectiveness of the proposed model is also indicated in Figs. 4 and 5 with the help of ROC curve that is used to study the output of a classifier. The area under curve (AUC) can be observed from these. Table 3 gives the detailed class wise results obtained from five-class classification of KDD Cup'99 data set. It can be analysed from results that for three classes—Normal, DoS and Probe—our proposed model gives the FAR, DR and ACC within acceptable limits. But for the minority classes (classes with lower number of instances), our proposed model gives poor results. For class U2Rand R2L, results are significantly poor. The reason stands same as the training instances are very less, and model is unable to learn their features effectively.

C. Comparison with other approaches

Fig. 4 ROC curve for two-class

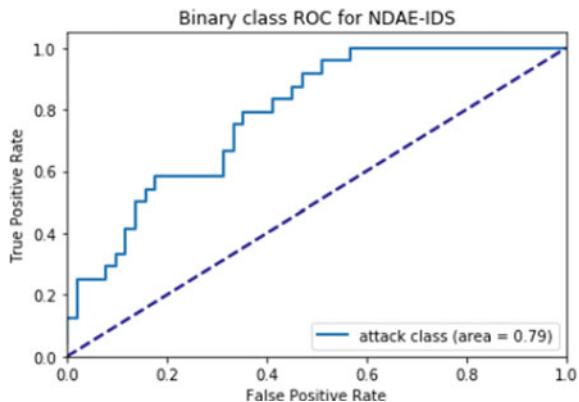


Fig. 5 ROC curve for five-class

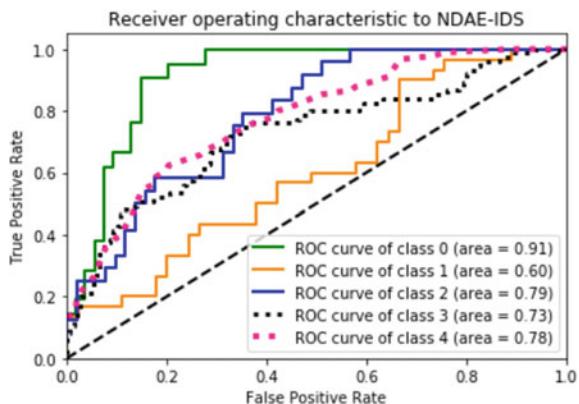


Table 3 Five-class performance matrix for NDAE-IDS

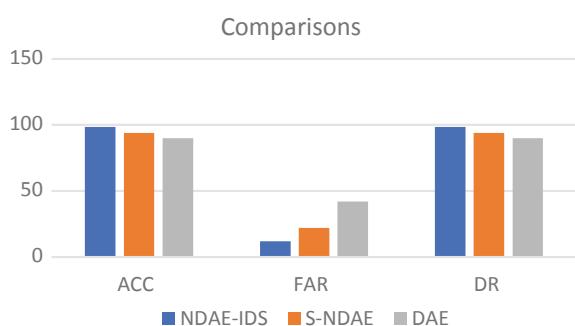
Class	FAR (%)	DR (%)	ACC (%)
Normal	8.65	98.39	98.39
DoS	8.05	97.68	97.68
U2R	100.00	0.00	0.00
R2L	0.71	8.20	8.20
Probe	10.84	96.63	96.63

Table 4 Comparison of performance metrics

Technique	Acc (%)	FAR (%)	DR (%)
NDAE-IDS	98.39	10.09	98.39
S-NDAE [9]	97.85	12.15	97.85
DAE [10]	93.49	30.42	93.49
SVM	95.21	25.30	95.21
Rfc	96.70	18.09	96.70

The comparative performance analysis of proposed approach has been carried out against two recent approaches. In [12], authors claim the achieved accuracy of 97.85% and FAR as 2.15%. They have performed five-class and 13-class classification on KDD Cup and NSL-KDD, whereas our model has yielded better results by achieving accuracy of 98.39%, FAR of 2.09% and DR of 98.39%. Farahnakian et al. [13] claim that their proposed model DAE has achieved accuracy 93.49% for KDD Cup'99 data set, whereas it can be clearly seen that our model has achieved much superior results. A very clear conclusion can be drawn from these comparisons that our model can give promising results as compared to other deep learning-based methods. Table 4 gives the comparative results, and Fig. 6 gives the graphical representation.

Fig. 6 Graphical representation of performance metrics



7 Conclusion and Future Work

This paper proposed a deep learning approach for network intrusion detection to improve attack detection with the help of auto-encoder (NDAE-IDS). Due to an increased number of attacks and network traffic, the crafting of proficient IDS has gained much attention. The auto-encoder has been the most efficient method for feature extraction from big and high-dimensional data. This is the primary reason for selecting it for this proposed model. This model consists of two auto-encoders in which the output of first auto-encoder acts as input for the next stacked auto-encoder. To train the model, layer wise training mechanism has been used, and shallow learning method, i.e. random forest has been used to classify the events into the attack and normal events. 10% of KDD Cup'99 has been used. The method shows good performance on accuracy, detection rate and false alarm rate metrics for the majority classes. Regardless, future work could continue to explore merits of deep learning methods to assess performance over minority classes in the network intrusion domain.

References

1. Zamani M, Movahedi M (2015) Machine learning techniques for intrusion detection. ARXIV 2:1–11
2. Sharma RK, Kalita HK, Borah P (2016) Analysis of machine learning techniques based intrusion detection systems Å supervised learning Å. In: Proceedings of 3rd international conference on advanced computing, networking and informatics, smart innovation, systems and technologies, vol 44, pp 485–493
3. Chowdhury MN, Ferens K, Ferens M (2010) Network intrusion detection using machine learning. In: International conference on security and management, pp 30–35
4. Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K (2012) An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst Appl 39(1):424–430
5. Bosch N, Paquette L (2017) Unsupervised deep autoencoders for feature extraction with educational data. In: 10th international conference on educational data mining, pp 11–18
6. Jayaswal A (2018) Detecting network intrusion through a deep learning approach. Int J Comput Appl 180(14):15–19
7. Guo L, Chen WH (2002) Disturbance attenuation for a class of nonlinear systems via disturbance-observer-based approach. IFAC Proc 15(1):19–24
8. K. Kim, *SPRINGER BRIEFS ON Network Intrusion Detection using Deep Learning A Feature Learning*, 1st ed. Springer Singapore, 2018.
9. Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell 2(1):41–50
10. Farahnakian F, Heikkonen J (2018) A deep auto-encoder based approach for intrusion detection system. Int Conf Adv Commun Technol ICACT 2018:178–183
11. Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K (2018) Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access 6(c):52843–52856
12. Javaid A, Niyaz Q, Sun W, Alam M (2016) A deep learning approach for network intrusion detection system. In: Proceedings of 9th EAI international conference on bio-inspired information communication technology (formerly BIONETICS)

13. Yu Y, Long J, Cai Z (2017) Session-based network intrusion detection using a deep learning architecture. In: Modeling decisions for artificial intelligence, pp 144–155
14. Seo S, Park S, Kim J (2016) Improvement of network intrusion detection accuracy by using restricted Boltzmann machine. In: IEEE 8th international conference on computational intelligence and communication networks improvement, pp 413–417
15. Ahmad I, Basher M, Iqbal MJ, Rahim A (2018) Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access* 6(c):33789–33795
16. Wang Q, Nguyen T, Huang JZ et al (2018) An efficient random forests algorithm for high dimensional data classification. *Adv Data Anal Classif* 12:953–972
17. Kursa MB, Rudnicki WR (2010) Feature selection with the Boruta package. *J Stat Softw* 36(11):1–13
18. Chang Y, Li W, Yang Z (2017) Network intrusion detection based on random forest and support vector machine. *IEEE Int Conf Comput Sci Eng IEEE Int Conf Embed Ubiquitous Comput*:635–638
19. Tavallaei M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set. In: IEEE symposium computer intelligent security defense application CISDA 2009, Cisda, pp 1–6
20. Salama MA, Eid HF, Ramadan RA, Darwish A, Hassanien AE (2011) Hybrid intelligent intrusion detection scheme. In: Soft computing in industrial applications, pp 293–303
21. Paliwal S (2012) Denial-of-service, probing and remote to user (R2L) attack detection using genetic algorithm 60(19):57–62

Chapter 39

Efficient Deep Learning Framework with Group Convolution for Segmentation of Histopathology Image



Amit Kumar Chanchal, Aman Kumar, Kumar Alabhya, Shyam Lal,
and Jyoti Kini

1 Introduction

In recent times, deep learning-based automatic detection and segmentation of histopathology images become one of the continually growing research areas due to the complexity present in such images. There are many limitations in conventional segmentation algorithm such as the discontinuity-based approach that is based on intensity variations of the pixel and works properly only if contrast is good. Boundary determined by the discontinuity-based approach is generally not continuous, and further post-processing is required to link these boundaries. There are various operators that work on the images, but not specifically work single operator on all images. The thresholding-based segmentation approach is simple and easy to implement, but this method works properly only when the image histogram has clear valleys. An image histogram having flat valleys does not work. For Otsu's segmentation method, image histogram should be bimodal. Otsu method takes the maximum interclass variance between the background and the target image as the threshold selection rule. Region-based segmentation algorithm is based on grouping, merging, sub-dividing, splitting of the pixels, and it is very expensive in terms of time and memory. Deep learning-based segmentation and detection approach segment the image automatically. We have an architecture, we provide input image to this architecture, and the features of input images learn from the model. We train the model with many numbers of inputs and keep the weight of the model to predict the unknown input.

A. K. Chanchal · A. Kumar · K. Alabhya · S. Lal (✉)

Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, Karnataka 575025, India

J. Kini

Department of Pathology, Kasturba Medical College, Mangalore, Manipal Academy of Higher Education, Manipal, India

Deep learning is driven by a proper activation function, an efficient loss function, and an effective optimization algorithm. Here we are using the rectified linear unit (ReLU) and sigmoid as an activation function, binary cross-entropy as a loss function, Adam optimization function to train the network.

The organization of the paper is as follows: In Sect. 2, we described the related research work, Sects. 3 and 4 describe the detailed architecture with the help of mathematical equations and block diagram. Visual representation of results is shown in Sect. 5, and finally, in Sect. 6, we conclude our work.

2 Related Work

Machine learning and deep learning already proved their effectiveness in the field of image processing and computer vision. There is a lot of related research for the detection and classification of microscopy images or histopathology images which have a very good impact. One of the major findings by Ronneberger et al. [1] called U-Net architecture which achieved a dominant breakthrough in the field of biomedical image segmentation. This is a U-shaped architecture, and it has a contraction path, the bottom layer, and an expansion path. Song et al. [2] employed a multi-scale deep learning method to extract invariant features where an unsupervised clustering method has been adopted to separate overlapping nuclei. For the pixel-based region segmentation of breast cancer histopathology images, Su et al. [3] proposed seven-layer fast scanning deep CNN by avoiding redundancy which has better scalability. Xing et al. [4] used CNN to segment the nuclei of three different data sets, namely breast cancer, brain tumor, and neuroendocrine tumor. Their deep CNN model generates a probability map, initializes the shape of nuclei by active contour model, and their section-based segmentation algorithm is able to identify clustered nuclei. (Kumar et al. [5]) CNN model generated a probability map to distinguish between inside nuclei and outside nuclei. Nuclei seeds are detected by setting a threshold value. The third class of nuclei is also generated which has heterogeneous chromatin distribution. This simplifies nuclei segmentation of even touching nuclei with a simple post-processing method. It can work for any complicated nuclear pleomorphism structure and can separate highly overlapped nuclei. A fully convolutional encoder-decoder architecture by Badrinarayanan et al. [6] used many potential opportunities which are actually missing in the original U-Net like batch normalization, improved up-sampling method, and many more. Xie et al. [7] used CNN to generate output feature maps, and this generated feature map is used for segmentation and detection tasks. For the purpose of semantic segmentation, Chen et al. [8] used spatial pyramid pooling in encoder-decoder architecture that retrieved multi-scale features. Pooling with multiple rates enlarges the overall field of view and captures relevant spatial feature. The use of depth-wise separable convolution makes their architecture efficient. For the diagnosis of breast cancer, Ting et al. [9] classification architecture also applied depth-wise separable convolution which is a computationally economical convolution method. Backpropagation is the weight updating method,

and ReLU is the activation method applied in the network. To reduce the computational complexity, Wu et al. [10] proposed a better up-sampling module called joint pyramid up-sampling (JPU) that accept three feature maps as input and produce better resolution feature. Their JPU module is able to extract more relevant features, which leads to better performance. To improve the performance of the existing encoder-decoder deep CNN architecture, Schlemper et al. [11] incorporate an additional gate that improves the method of up-sampling called attention gate. This attention gate adds more relevant features from down-sampling paths of similar resolution. Naylor et al. [12] address one of the major challenges in automatic segmentation of histopathology images, to segment the overlapped nuclei by developing the segmentation task as a regression problem of the distance map. They develop inter-nuclei differences based on a loss function that is able to identify the clumped nuclei. (Jung et al. [13]) The architecture used mask R-CNN for the segmentation of multi-organ histopathology image and a separate breast cancer histopathology images. Their U-net-based model used Gaussian mixture color normalization, and they also used multiple inferences-based post-processing to improve the performance of the model. To increase the magnification level by learning the complex morphological structure, a deep learning-based architecture called Deep-Hipo by Kosaraju et al. [14]. This network extracts multiple patches of the same size and captures multiple morphological structures of a varying receptive field that leads to better performance. For the segmentation of breast cancer, Priego-Torres et al. [15] used a separable dilated convolution method in encoder-decoder architecture. Their patch-wise implementation which is inspired by Deeplabv3 architecture achieved very good results in terms of intersection over the union.

3 Proposed Architecture

In the segmentation task, encoder-decoder architecture is the best fit as in [1, 6, 11]. For better segmentation accuracy, we are generally going deeper and deeper into the network to extract more relevant features that result in higher computational complexity. In this architecture shown in Fig. 1, we are doing group convolution which is the combination of standard convolution and depth-wise separable convolution that effectively reduces the total number of parameters while maintaining model accuracy.

The mathematical expression of 2D standard convolution in Eq. (1)

$$g[m, n] = f(m, n) * h(m, n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(k, l)h(m - k, n - l) \quad (1)$$

To reduce the trainable parameter, we change the convolution strategy before pooling called as depth-wise separable convolution, used in [16, 17] as shown in Fig. 2.

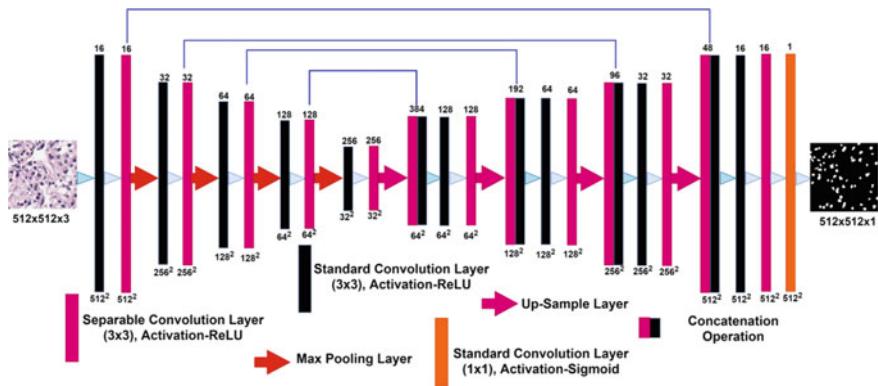


Fig. 1 Encoder-decoder architecture with standard convolution and depth-wise separable convolution layer

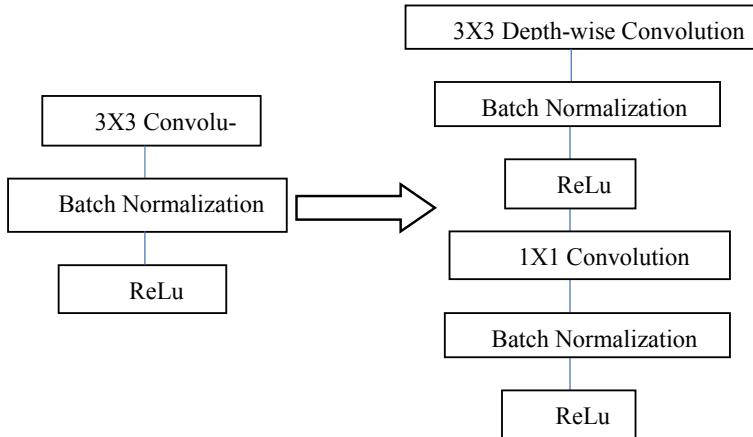


Fig. 2 Standard convolution changed to depth-wise separable convolution

Depth-wise separable convolution involves two steps as follows:

- Depth-wise convolution where we convolved the input image and kernel without changing depth. The input image of size $D_f \times D_f \times M$ convolved with M filters of $D_k \times D_k \times 1$ yields $D_p \times D_p \times M$ shown in Fig. 3. Then, the total number of multiplications involved in this step is $(D_p \times D_p \times M) \times (D_k \times D_k \times 1)$. Number of parameters in depth-wise convolution = $(D_k \times D_k \times M)$.
- Point-wise convolution used 1×1 kernel, and it has same depth as the output of previous stage. $(D_p \times D_p \times M)$ Convolved with N number of $(1 \times 1 \times M)$ filters yield $(D_p \times D_p \times N)$. Then, the total number of multiplications involved in this step is $(1 \times 1 \times M) \times (D_p \times D_p \times N)$. Number of parameters in depth-wise convolution = $(M \times N)$.

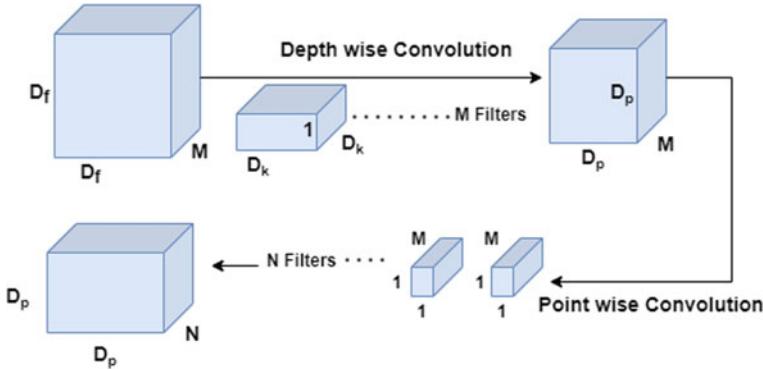


Fig. 3 Depth-wise convolution and point-wise convolution

Total number of multiplication involved in depth-wise convolution = number of multiplication involved in depth-wise convolution + number of multiplication involved in point-wise convolution = $(D_p \times D_p \times M) \times (D_k \times D_k \times 1) + (1 \times 1 \times M) \times (D_p \times D_p \times N) = D_p^2 \times D_k^2 \times M + D_p^2 \times M \times N$ which is lesser than standard convolution ($D_p^2 \times D_k^2 \times M \times N$). Total number of parameters in depth-wise separable convolution = $(D_k \times D_k \times M) + (M \times N)$ which is significantly smaller than standard convolution ($D_k \times D_k \times M \times N$).

4 Training and Implementation

For training and implementation, we have used Keras and TensorFlow Python framework, Google GPU, Adam [18] optimizer, and batch size of 4 for the best possible outcome. The preferred loss function is expressed in Eq. (2) and used in [1, 6] called as binary cross-entropy.

$$\text{Cross entropy loss for } c = 2 \text{ class} = \sum_{l=1}^{c=2} t_l \log(f(s_l)) \quad (2)$$

4.1 Description of Dataset

The 730 H&E stained histopathology images of kidney dataset have been used by [19] and obtained from the National Cancer Institute. Fifty H&E stained histopathology images of a TNBC dataset of human breast tissue used by [5] consist of 4022 cells that are annotated accurately by a pathologist.

5 Performance Metrics and Results

After the simulation, we have been able to express our results in terms of $F1$ -score or dice score used in [20] and aggregated Jaccard index (AJI) used in [12] and compared our proposed model with three standard models which are recently reported their result in the field of biomedical image segmentation.

5.1 Comparison of Methods

Table 1 shows a comparison of the proposed architecture with three other segmentation architectures for the kidney dataset and TNBC dataset. Performance measurement in terms of $F1$ -zcore, AJI score, and the total number of trainable parameters describe the training time and complexity. Results indicated that our architecture is able to retrieve more information compared to others.

Visual segmentation comparison of different models on the kidney dataset and TNBC dataset is shown with two sample images and their overlay images in Figs. 4 and 5, respectively. The top row of both Figs. 4 and 5 has two sample images of ground-truth images and their corresponding overlay images. Overlay image is generated with the help of original histopathology image and predicted image to visually see how much our segmentation model is accurate.

The second row from the top of both the figures has predicted images by the U-Net model and corresponding overlay image. In most of the nuclei detected by the U-Net model, only a few nuclei are not detected or partially detected. Many additional ducts are present in U-Net which are not desirable. A number of partially detected nuclei increase in attention U-Net model. The less number of overlapped nuclei is present in the Dist model, but the number of nuclei not detected increases compared to the other two models. In the proposed model, most of the nuclei identified correctly and no additional ducts were detected, and the distribution of detecting nuclei is almost similar to the ground-truth.

Table 1 Performance comparison of architectures with kidney and TNBC dataset

Model name	$F1$ -score		AJI		Parameters
	Kidney dataset	TNBC dataset	Kidney dataset	TNBC dataset	
U-Net (2015)	0.8537	0.7324	0.7489	0.6559	31,378,945
Attention U-Net (2019)	0.9135	0.7216	0.8590	0.6194	31,902,629
Dist (2019)	0.8992	0.7516	0.8272	0.6727	7,771,873
Proposed model	0.9264	0.7836	0.8653	0.6864	1,099,745

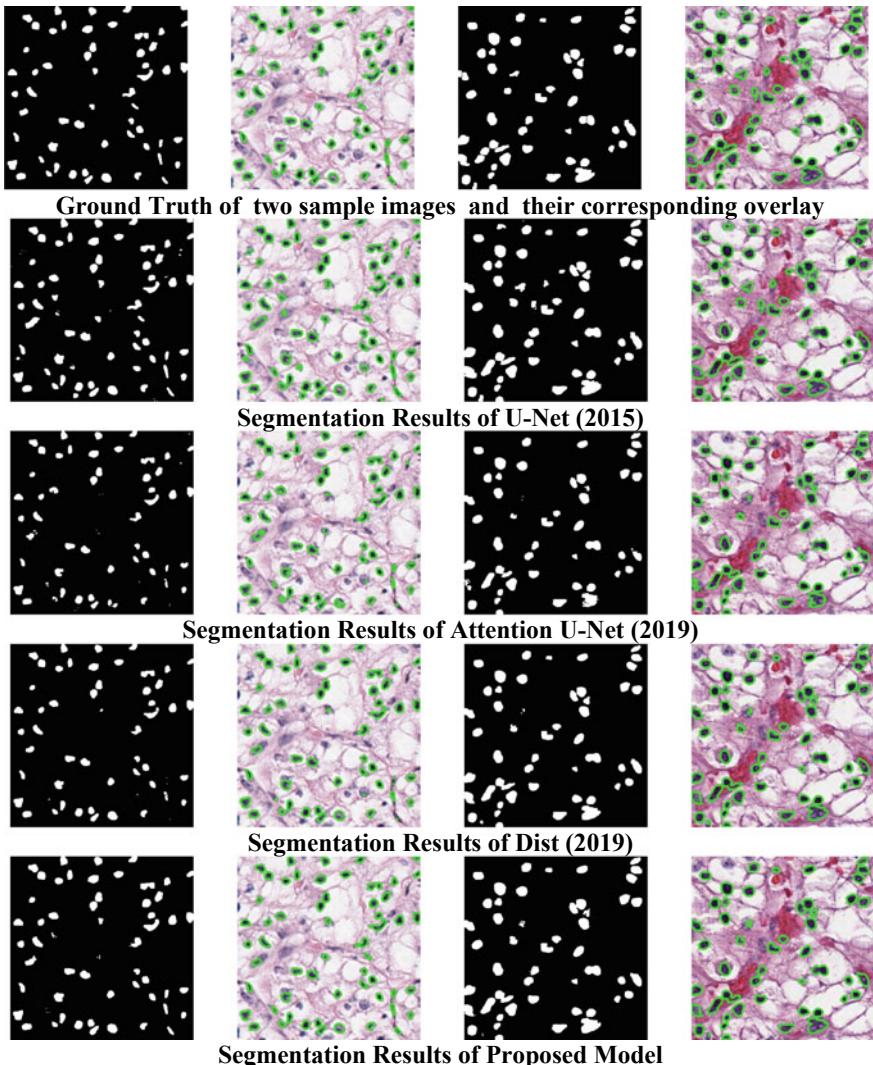


Fig.4 Visual segmentation comparison of different models on two sample images of the kidney dataset

6 Conclusion

Higher computational complexity due to the extraction of multilayer features in very deep architecture takes longer training time and requires expansive GPU. To address this problem, we proposed a convolution strategy called group convolution in encoder-decoder architecture where we applied standard convolution and depth-wise separable convolution that effectively reduces the trainable parameters. Results

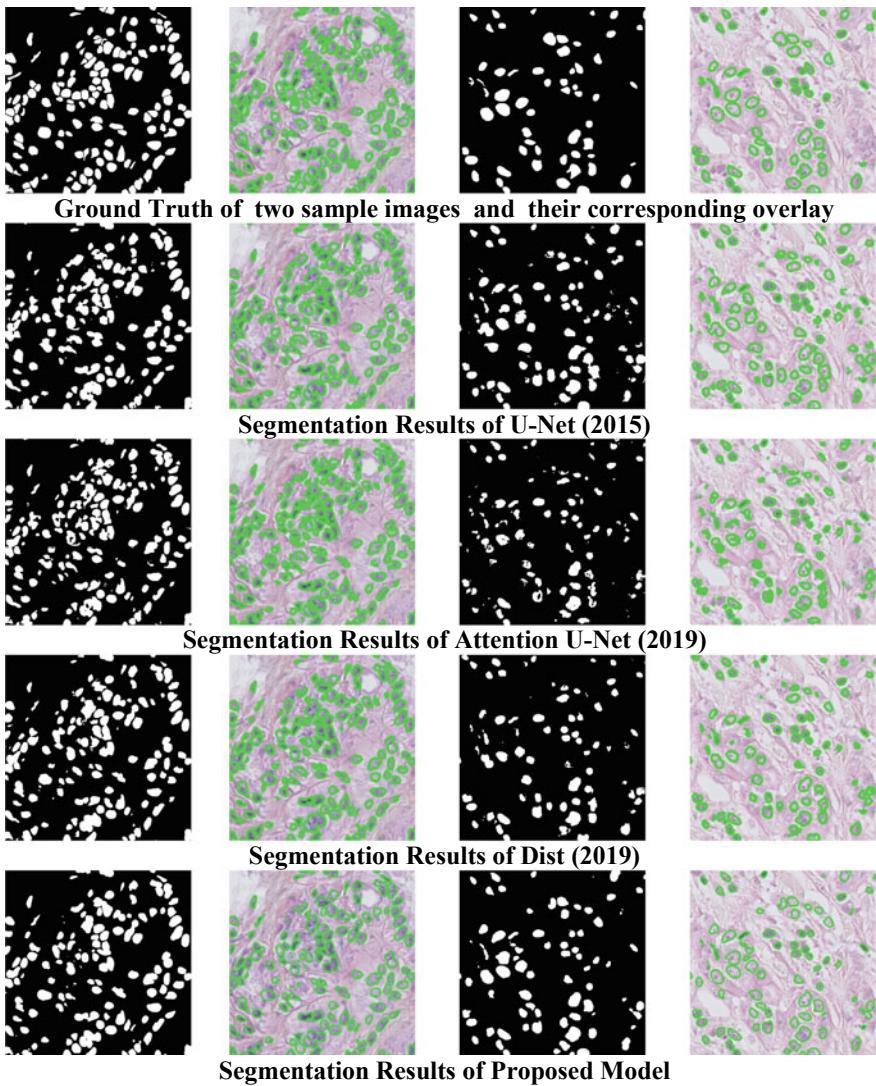


Fig.5 Visual segmentation comparison of different models on two sample images of the TNBC dataset

indicated that for both the datasets, our architecture is better in terms of $F1$ -score and AJI score with less complexity, further visual results also indicated that most of the nuclei identified correctly as compared to other benchmark segmentation results.

Acknowledgements This research work was supported by the Science Engineering and Research Board, Department of Science and Technology, Govt. of India under Grant No. EEG/2018/000323, 2019.

References

1. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Proceedings of medical image computing and computer assisted intervention, MICCAI. Springer, pp 234–241
2. Song Y, Zhang L, Chen S, Ni D, Lei B, Wang T (2015) Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph Partitioning. *IEEE Trans Biomed Eng* 62(10):2421–2433
3. Su H, Liu F, Xie Y, Xing F, Meyyappan S, Yang L (2015) Region segmentation in histopathological breast cancer images using deep convolutional neural network. In: 2015 IEEE 12th international symposium on biomedical imaging (ISBI), New York, NY, pp 55–58. <https://doi.org/10.1109/ISBI.2015.7163815>
4. Xing F, Xie Y, Yang L (2016) An automatic learning-based framework for robust nucleus segmentation. *IEEE Trans Med Imaging* 35(2):550–566. <https://doi.org/10.1109/TMI.2015.2481436>
5. Kumar N, Verma R, Sharma S, Bhargava S, Vahadane A, Sethi A (2017) A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Trans Med Imaging* 36(7):1550–1560. <https://doi.org/10.1109/TMI.2017.2677499>
6. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(12):2481–2495
7. Xie L, Li C (2018) Simultaneous detection and segmentation of cell nuclei based on convolutional neural network. In: Proceedings of the 2nd international symposium on image computing and digital medicine. ACM, Chengdu, pp 129–132
8. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Computer vision—ECCV 2018. ECCV 2018. Lecture notes in computer science, vol 11211. Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_49
9. Ting FF, Tan YJ, Sim KS (2019) Convolutional neural network improvement for breast cancer classification. *Expert Syst Appl* 120:103–115. ISSN 0957-4174. <https://doi.org/10.1016/j.eswa.2018.11.008>
10. Wu H, Zhang J, Huang K, Liang K, Yu Y (2019) FastFCN: rethinking dilated convolution in the backbone for semantic segmentation. [arXiv:1903.11816v1](https://arxiv.org/abs/1903.11816v1) [cs.CV]
11. Schlemper J, Oktay O, Schaap M, Heinrich M, Kainz B, Glocker B, Rueckert D (2019) Attention gated networks: learning to leverage salient regions in medical images. *Med Image Anal* 53:197–207. ISSN 1361-8415. <https://doi.org/10.1016/j.media.2019.01.012>
12. Naylor P, Laé M, Reyal F, Walter T (2019) Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE Trans Med Imag* 38(2):448–459
13. Jung H, Lodhi B, Kang J (2019) An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images. *BMC Biomed Eng* 1:24. <https://doi.org/10.1186/s42490-019-0026-8>
14. Kosaraju SC, Hao J, Koh HM, Kang M (2020) Deep-hipo: multi-scale receptive field deep learning for histopathological image analysis. *Methods* 179:3–13. ISSN 1046-2023. <https://doi.org/10.1016/j.ymeth.2020.05.012>
15. Priego-Torres BM, Sanchez-Morillo D, Fernandez-Granero MA, Garcia-Rojo M (2020) Automatic segmentation of whole-slide H&E stained breast histopathology images using a deep convolutional neural network architecture. *Expert Syst Appl* 151:113387. ISSN 0957-4174. <https://doi.org/10.1016/j.eswa.2020.113387>
16. He Y, Qian J, Wang J (2019) Depth-wise decomposition for accelerating separable convolutions in efficient convolutional neural networks. <https://arxiv.org/abs/1910.09455>
17. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobile nets: efficient convolutional neural networks for mobile vision applications. *Comput Vis Pattern Recognit*. <https://arxiv.org/abs/1704.04861>
18. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. In: International conference on learning representations. <https://arxiv.org/abs/1412.6980>

19. Irshad H, Montaser-Kouhsari L, Waltz G, Bucur O, Nowak JA, Dong F, Knoblauch NW, Beck AH (2015) Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: evaluating experts, automated methods, and the crowd. Pacific Symp Biocomput. <https://doi.org/10.13140/2.1.4067.0721>
20. Lal S, Kanfade A, Alabhyia K, Dsouza R, Kumar A, Chanchal AK, Maneesh M, Peryail G, Kini J (2020) A robust method for nuclei segmentation of H&E stained histopathology images. In: 7th IEEE international conference on signal processing and integrated networks (SPIN2020), Amity University, Delhi NCR, Noida, UP

Chapter 40

Dynamic Reusability Measurement Using Machine Learning Algorithms in Object-Oriented Environment



Manju and Pradeep Kumar Bhatia

1 Introduction

Reusability of code plays an important role to improve the quality of software. In an OO paradigm, to find a class or code is reusable or not is achieved with the help of metrics. Many researchers [1–4] worked on various aspects of metrics like coupling, cohesion, complexity, etc. to identify the software reusability. However, all these studies were mainly based on static metrics by capturing the design patterns for measuring the quality of software. But in modern OO software, there exist features like run time polymorphism [5], dynamic binding, run time complexity of methods, etc. that can be captured only by using dynamic metrics. So, the focus of the software industry moves from static metrics to dynamic metrics to measure software reusability in advance. Therefore, in this paper, authors mainly focus on five factors to measure the reusability factor of a class, i.e., polymorphism, inheritance, number of children, coupling and complexity. Metrics used to compute these factors are described in Table 1. Further, we proposed a fuzzy model based on these five factors and calculated reusability of a design pattern based on static and dynamic metrics as fuzzy logic provides us better flexibility to predict the reusability of a class or code in the form of low, medium and high rather than crisp logic. Various other existing machine learning algorithms [6] were also applied to data and compared with the proposed fuzzy logic approach.

The rest of the paper is divided into four sections. Section 2 explains the literature review. Section 3 contains the proposed formula and proposed fuzzy logic model. In Sect. 4, the experimental study and validation of the proposed model are explained by the AHP technique. Finally, Sect. 5 presents the conclusions and the future directions referenced to this paper.

Manju () · P. K. Bhatia

Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, India

Table 1 Static and dynamic metrics description

Static metrics	Description
SP	Ratio of number of overloading and overridden methods in class to the total number of methods in a class [7]
WMC	Weighted sum of the complexity of the methods of a class [8]
DIT	The maximum length from the node to the root of the tree [8]
NOC	Number of immediate subclasses subordinated to a class in the class hierarchy [8]
CBO	Count of the number of other classes to which it is coupled [8]
Dynamic metrics	Description
DP	Ratio of number of overloading and overridden methods executed at run time to the total number of times methods of a class execute at run time [7]
DWMC	Sum of number of times methods of a class executed at run time [7]
DCBO	Count of the number of other classes to which it is coupled at run time (Mitchell and Power 2004)

2 Related Work

Many empirical studies exist in literature to validate the strong relationship between design patterns and respective reusability. Bhatia and Mann [1] proposed a formula based on CK metrics suite to find the reusability of a class diagram. Sharma et al. [9] proposed ANN to assess the reusability of the software component based on five factors. Sangwan et al. [2], compared the fuzzy, neural and neuro-fuzzy approach to find the reusability of a system based on CK metrics suite and concluded that the neuro-fuzzy model was best among all. Goyal and Gupta [3] proposed an unsupervised neural network model to find the reusability of a class. Kumar and Bhatia [4] proposed a neuro-fuzzy model to find reusability of components using the CK metrics suite. Hence, it is clear from the existing literature survey that most of the studies conducted yet were based on static metrics. Therefore in this paper, authors proposed a fuzzy model to find the reusability factor of a class based on static and dynamic metrics.

3 Proposed Work

3.1 Proposed Formula

A formula is proposed based on metrics described in Sect. 1. Our approach is to derive a formula to measure reusability of a design pattern based on the following principles:

- Higher the degree of polymorphism in class, the greater the potential for reuse. Therefore, SP and DP have a positive impact on reusability.
- If depth of inheritance is more, reuse of code is higher. Therefore, DIT has a positive impact on the reusability of a class [1].
- NOC value up to a particular threshold defines the scope for reuse. Therefore, it has a positive impact on the reusability of a class [1].
- If coupling between classes is high, then it may inhibit reuse. Therefore, it indicates that CBO [1] and WCBO have a negative impact on the reusability of a class.
- If the complexity of a class increases, then it is very hard to maintain and reuse the code. Therefore, WMC[4] and DWMC have a negative impact on class reusability.

A formula is derived using above principles defined as following:

$$\begin{aligned} \text{Reusability of a class based on static metrics (RS)} = & w1 * (\text{SP}) \\ & + w2 * (\text{DIT}) + w3 * (\text{NOC}) \\ & - w4 * (\text{CBO}) - w5 * (\text{WMC}) \end{aligned} \quad (1)$$

$$\begin{aligned} \text{Reusability of a class based on dynamic metrics (RD)} = & w1 * (\text{DP}) \\ & + w2 * (\text{DIT}) \\ & + w3 * (\text{NOC}) \\ & - w4 * (\text{DCBO}) \\ & - w5 * (\text{DWMC}) \end{aligned} \quad (2)$$

where $w1, w2, w3, w4$ and $w5$ are the weights to be calculated by AHP technique described in Sect. 4.

3.2 Proposed Fuzzy Logic Model

The proposed fuzzy model was based on five inputs, i.e., polymorphism, inheritance, number of children, coupling, complexity to the Mamdani's fuzzy inference system (MFIS) and one output, i.e., reusability as shown in Fig. 1. Further, three terms like low, medium, high and five values were taken as input to the model. Therefore, according to formula, 35 rules were developed. After fuzzification process completed, we took the fuzzy sets for the output variable that required defuzzification. For defuzzification process, fuzzy sets act as input that gives singleton values as output. Triangular membership function was used to define the values of membership function as low, medium and high for each input as shown in Fig. 2.

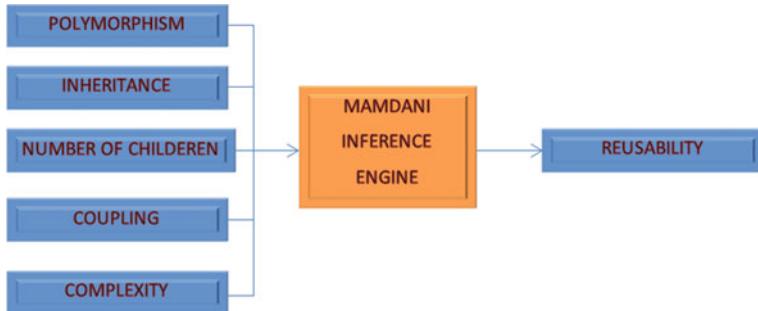


Fig. 1 Proposed fuzzy logic model

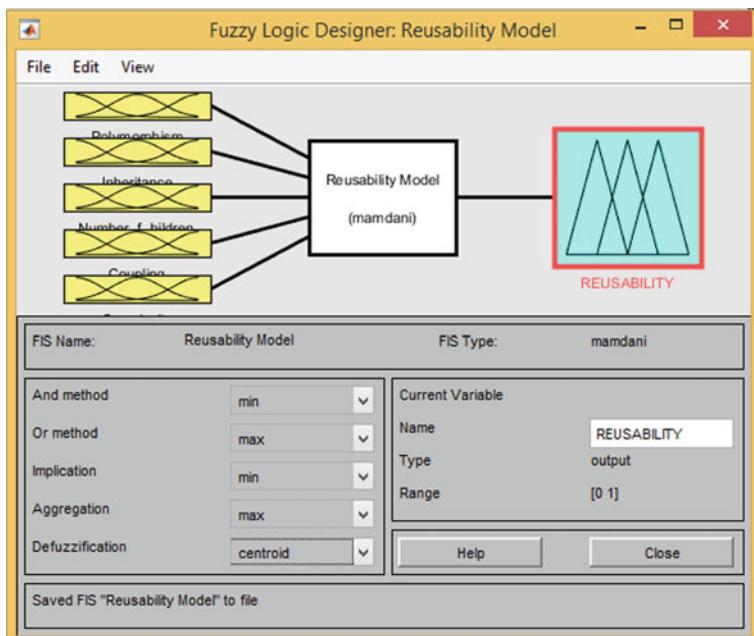


Fig. 2 Proposed Mamdani's fuzzy inference system with triangular membership functions

4 Experimental Study

An experimental study was conducted on 18 design pattern [10, 11] having 92 Java classes taken from the web. Static metrics were extracted using CodeMR [12] tool and dynamic metrics were extracted using AspectJ [13, 14], an implementation of aspect-oriented programming on eclipse [15] platform. Statistical data of static and dynamic metrics are shown in Table 2.

Table 2 Statistical data of static and dynamic metrics

Static metrics	Min	Max	Mean	Med.	Std. Dev.	Dynamic metrics	Min	Max	Mean	Med.	Std. Dev.
SP	0	1	0.43	0.5	0.40	DP	0	1	0.36	0.23	0.39
DIT	0	2	0.63	1	0.56	DIT	0	2	0.63	1	0.56
NOC	0	3	0.58	0	0.92	NOC	0	3	0.58	0	0.92
CBO	0	4	0.78	0	1.08	DCBO	0	4	0.78	0	1.08
WMC	0	9	0.23	2	1.65	DWMC	0	14	3.08	2	3.33

Table 3 AHP decision values

Factors	Polymorphism	Inheritance	NOC	Coupling	Complexity	Eigen vector (w)
Polymorphism	1	3	3	7	9	0.46
Inheritance	1/3	1	3	3	7	0.24
NOC	1/3	1/3	1	3	7	0.16
Coupling	1/7	1/3	1/3	1	7	0.11
Complexity	1/9	1/7	1/7	1/7	1	0.03
Total						1.000

The proposed model was validated by using standard analytic hierarchy process (AHP) technique [16]. Firstly, the consistency index was calculated by finding the average of calculated values using the Eigen vector of five factors as shown in Table 3. After that, consistency ratio was calculated depends on the number of factors, i.e., 5.

Consistency index (CI) = $(\lambda_{\max} - n)/(n - 1)$ where $n = 5$, i.e., $(5.28 - 5)/4 = 0.07$.

Consistency ratio (CR) = $0.07/1.11 = 0.06 < 0.1$, for $n = 5$ index of consistency = 1.11. If the value of consistency ratio was less than 0.1, then the decision value was accepted; otherwise, it is rejected. Therefore, $w_1 = 0.46$, $w_2 = 0.24$, $w_3 = 0.16$, $w_4 = 0.11$ and $w_5 = 0.03$. Therefore, by putting the weight values in Eqs. (1) and (2) described in Sect. 3.1, value of reusability was calculated.

Various other existing machine learning algorithms were also applied on static and dynamic metrics data by using Weka 3.8 tool and compared with proposed fuzzy logic approach on the basis of mean absolute error (MAE) and root mean square error (RMSE) values as shown in Table 4.

Table 4 Comparison of machine learning algorithms

Machine learning algorithms	Static metrics		Dynamic metrics	
	MAE	RMSE	MAE	RMSE
Proposed fuzzy logic model	0.13	0.15	0.03	0.05
Linear regression	0.06	0.07	0.07	0.08
SMOreg	0.06	0.10	0.07	0.11
Random forest	0.04	0.06	0.04	0.06
SLR	0.11	0.14	0.12	0.15
M5P	0.04	0.06	0.04	0.05
Gaussian processes	0.07	0.09	0.07	0.10

5 Conclusion

This paper proposed Mamdani's fuzzy inference system (MFIS) to measure the reusability of object-oriented software systems. The proposed model has five inputs, i.e., polymorphism, inheritance, number of children, coupling and complexity and one output, i.e., reusability. A total of 243 rules were generated for evaluating the reusability factor of various design patterns. Authors validated the proposed fuzzy model and proposed formula by using the AHP technique. Besides, the proposed model calculated the reusability of 18 design patterns having 92 java classes taken from the web. Further, authors compared the proposed fuzzy model with six machine learning algorithms, and it is found that the proposed fuzzy model gives satisfactory results based on dynamic metrics among all machine learning algorithms. Hence, we can say that dynamic metrics are a better predictor of the reusability factor. In this paper, we have worked on a small set of design patterns with a small set of metrics. Therefore, more work is required by taking a large set of metrics on large projects.

References

1. Bhatia PK, Mann R (2008) An approach to measure software reusability of OO Design. In: 2nd national conference on challenges and opportunities in information technology (COIT-2008) RIMT-IET, Mandi Gobindgarh, pp 26–30
2. Sangwan OP, Bhatia PK, Singh Y (2011) Software reusability assessment using soft computing techniques. ACM SIGSOFT Softw Eng Notes 36(1):1–7
3. Goyal N, Gupta D (2014) Reusability calculation of object oriented software model by analyzing CK metric. Int J Adv Res Comput Eng Technol 3(7):2466–2470
4. Kumar G, Bhatia PK (2015) Neuro-fuzzy model to estimate and optimize quality and performance of component based software engineering. ACM SIGSOFT Softw Eng Notes 40(2):1–6
5. Choi KHT, Tempero E (2007) Dynamic measurement of polymorphism. Thirtieth Australasian Comput Sci Conf Victoria Australia 62:211–220
6. Han J, Kamber M (2006) Data mining concepts and techniques, 2nd edn. Elsevier

7. Manju BPK (2020) Impact of dynamic polymorphism on quality of a system. *J Sci Technol* 5(2):54–59
8. Chidamber SR, Kemerer CF (1994) A metrics suite for object oriented design. *IEEE Trans Softw Eng* 20(6):476–493
9. Sharma A, Kumar R, Grover PS (2009) Reusability assessment for software components a neural network based approaches. *ACM SIGSOFT Softw Eng Notes USA* 34(2):1–6
10. Design Patterns. <https://www.geeksforgeeks.org/chain-responsibility-design-pattern/>, last accessed 2020/05/06
11. Inheritance in Java Homapage. <https://www.geeksforgeeks.org/inheritance-in-java/>, last accessed 2020/05/06
12. CodeMR Guide. <https://www.codemr.co.uk/docs/codemr-intellij-userguide.pdf>, last accessed 2020/05/06
13. AspectJ Tutorial. <https://o7planning.org/en/10257/java-aspect-oriented-programmingtutorial-with-aspect>, last accessed 2020/05/06
14. AspectJ Homepage. <https://www.eclipse.org/aspectj>, last accessed 2020/05/06
15. Eclipse Guide. <https://www.eclipse.org/aspectj/doc/next/progguide/printable.html>, last accessed 2020/05/06
16. Saaty TL (1990) How to make a decision: the analytic hierarchy process. *Eur J Oper Res* 48:9–26

Chapter 41

Comparison of Different Machine Learning and Deep Learning Emotion Detection Models



Akanksha Aggarwal, Sahil Garg, Raghav Madaan, and Rajender Kumar

1 Introduction

According to the Association for the Advancement of Artificial Intelligence—AAAI, the emotion detection AI is a \$20 billion industry as of today. A daunting issue of automated human impact detection has become a study area involving rising numbers of scientists trained in various fields such as artificial intelligence, computer vision, psychology and physiology. Like other non-verbal elements, facial gestures are one of the key sources of awareness in interpersonal contact through holding emotional meaning [1]. It is only normal that work into facial expression has got a plenty of interest over the past few years. Use of facial expressions is popular during a conversation to express emotions. They convey the same emotion through ages, and certain facial gestures are common. The emotion detection has various challenges and offers numerous opportunities in fields like psychology, criminal assessment, gaming industry, software development, etc [2]. This paper provides an insight into comparing the performance of different modules by which human facial emotions can be detected. Two different approaches were used, namely machine learning and deep learning. For machine learning, four different models have been used. For deep learning approach, three different models were used.

2 Literature Survey

Face gesture-based emotion recognition systems allow for the study, labelling and inference of cognitive affective conditions from face video recording in real time. An attempt to manipulate the facial expression gets triggered for a short duration

A. Aggarwal · S. Garg (✉) · R. Madaan · R. Kumar
National Institute of Technology, Kurukshetra, Haryana 136119, India

when human faces an emotion and is known as facial action [3]. From each of these emotions, a particular collection of facial behaviours is extracted using computer vision techniques which identify independent motion of the face. Movements of facial muscles are interpreted as shifts of head, nose and mouth locations. This method is applied by computer systems recording pictures of the user's facial gestures and head motions. Such systems identify adjustments in the head, nose and mouth location as shifts in a coordinate system's dot pattern [4]. And, by evaluating certain shifts, the emotion experience can be determined.

3 Introduction to Technologies Used

It has been observed that deep learning provides better results than conventional machine learning because it processes the data in its original form unlike conventional machine learning algorithms [5]. So an attempt is made to compare accuracy for machine learning-based models—logistic regression, Naïve Bayes, KNN and SVM and deep learning-based models—neural networks, convolution neural networks and transfer learning-based MobileNet model. Same set of test and train images were fed as input to different models, and confusion matrix was used to find out accuracy score for each model. This accuracy score was sole criteria for comparing accuracy of above-mentioned emotion detection models.

3.1 Data set

The data set used for training and testing is Kaggle facial recognition challenge (FER 2013). It contains 48-by-48 pixel grayscale images of faces, all labelled with one of the five emotion classes: anger, happy, sad, surprised and neutral. Total image samples available in data set are 30,219 with facial expression, labels out of which 24,282 samples are used for training and rest 5937 are used for testing [6]. Emotion classes are considered in lexicographical order for usage. Also, data set is shuffled randomly so that the samples in data set are available in random order [6].

3.2 Libraries

Following libraries are used at various levels: NumPy, Pandas, Matplotlib, operating system, Time, Pickle, OpenCV, scikit-learn, Keras and Pathlib.

3.3 Machine Learning and Deep Learning Models Used

To detect the emotions of the images used in the data set, following different types of models based upon machine learning and deep learning were used.

Machine Learning Models Used

1. Logistic Regression: This predictive analysis machine learning algorithm is used for the classification problems. The hypothesis of logistic regression limits the cost function between 0 and 1 [7].
2. Naïve Bayes Classifier: It is a probabilistic machine learning model based on Bayes theorem. The assumption made here is that the predictors/features are independent, i.e. naïve [8].
3. K-Nearest Neighbours (KNN): It is a nonlinear classifier. K-nearest neighbours are chosen according to the Euclidean distance, and numbers of data points in each category are counted [9].
4. Support Vector Machine (SVM): It is based on the concept of finding a hyperplane which will better divide the features into different domains. When the data is not linear, kernel is used [10].

Deep Learning Models Used

1. Neural Networks: Neural networks are the classic example of machines mimicking how the brain works. These are the layers of highly interconnected processing elements called neurons that transform the data to develop its own understanding of data and give corrected outputs [11].
2. CNN: It is mainly used for image processing, classification, segmentation, etc. Convolution, rectified linear unit, pooling and classification are the main four operations used in convolutional neural network [12].
3. Transfer Learning: The advantages of using transfer learning are as follows: There is no need for very large training data set, and it requires less computing power. Training process becomes fast because as only a few dense layers were trained [13].

4 Modelling and Simulation

Modelling includes the network structure of MobileNet model used. It shows the layered architecture used to obtain the efficient emotion recognition model. Simulation consists of the hardware used to obtain the results and the outputs obtained. This model can be used for real-time emotion recognition also.

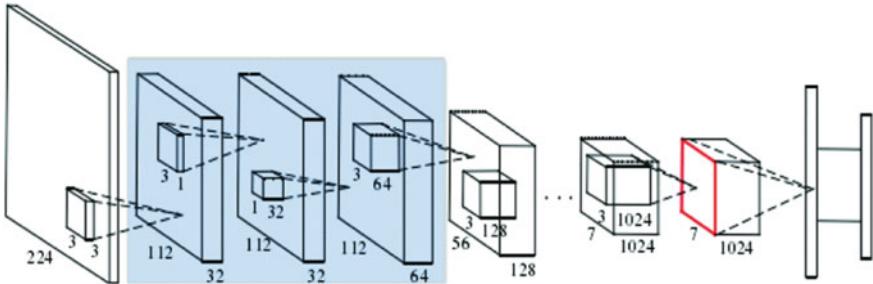


Fig. 1 MobileNet network structure

4.1 Mobilenet Network Structure

This network is centred on simplified design. The central concept is to substitute conventional convolution with deeply separable convolution, essentially reducing the redundant representation of the convolution kernel and creating a balance between the amount of output channels and the kernel scale. In Fig. 1, the icon red of the MobileNet network layout reflects the network's final classification and can be separated into 1024 groups [13].

4.2 Pseudo-code

Pseudo-code of transfer learning model:

- Step 1. **Select** a pre-trained deep learning model MobileNet. Use dataset nomenclature variable array CLASSES containing emotion names in lexicographical order- anger, happy, neutral, sad, surprise
- Step 2. **Remove** the last few output layers of selected model
 $IMG_ROW, IMG_COL \leftarrow 224, 224$
MobileNet contains ImageNet weights and removes the top layer and considers the image shape in above written variables format only
- Step 3. **Freeze** the weights of the remaining layers
- Step 4. **Add** few convolution or dense layers on the top of the model to make custom classifiers
define the function addTopModelMobileNet with arguments
BOTTOM_MODEL and NUM_CLASSES
creates the top or head of the model that will be placed on top of the bottom layers
RETURN TOP_MODEL
- Step 5. **Train** the model on the training dataset
- Step 6. **Validate** the model on the validation dataset
- Step 7. If needed, **unfreeze** more layers and fine-tune the hyper parameters

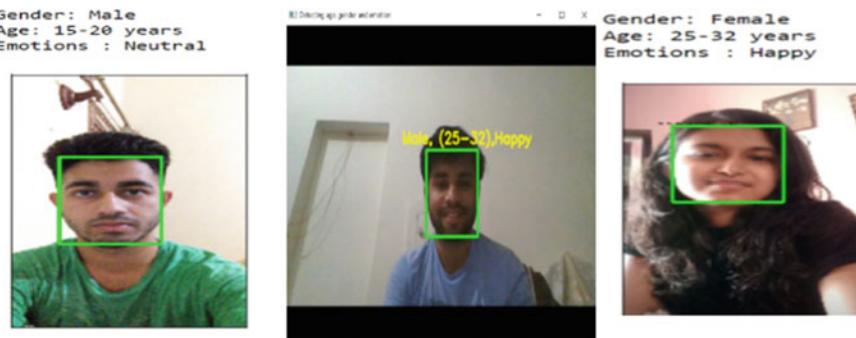


Fig. 2 Emotion detection outputs

4.3 Simulation and Outputs

The simulation was performed on a Jupyter notebook environment, and hardware used was based upon Core i5 8th Gen Processor. Output accuracy, i.e. the percentage of predicting emotions correctly using transfer learning-based MobileNet model was found to be 68%. Various other models were also used, for which the detailed analysis is present in results section. Figure 2 represents the outputs of MobileNet based model.

5 Results and Discussion

Confusion matrix is used for the calculation of accuracy of the emotion detection models used. The accuracy percentage for each ML and DL model as shown in Figs. 3 and 4, respectively, was calculated using the formula:

$$\text{Accuracy Percentage} = \frac{\text{Number of correctly identified emotion outputs}}{\text{Total number of input images}} \times 100 \quad (1)$$

5.1 Machine Learning-based Algorithms Accuracy

The machine learning algorithms are displayed relatively low as compared to machine learning because the deep learning algorithms work on exact data set provided, whereas machine learning algorithms required data set to be modified so that it can

Angry	222	312	154	217	55
Happy	73	1378	137	195	42
Neutral	73	328	533	327	45
Sad	71	311	211	507	39
Surprise	31	156	95	70	455
Support Vector Machine accuracy = 51%					
Angry	84	144	75	303	354
Happy	99	477	167	447	635
Neutral	48	201	186	366	415
Sad	81	169	135	451	303
Surprise	33	76	61	86	341
Gaussian Naïve Bayes accuracy = 29%					
Angry	237	238	149	73	263
Happy	377	728	291	147	282
Neutral	167	307	367	107	268
Sad	235	261	228	162	253
Surprise	106	148	131	72	340
Multinomial Naïve Bayes accuracy = 31%					
Angry	273	290	253	100	44
Happy	175	965	473	147	65
Neutral	145	373	522	110	66
Sad	174	355	324	247	39
Surprise	69	197	174	67	290
K-Nearest Neighbors accuracy = 38%					
Angry	190	290	171	217	92
Happy	177	1100	214	257	77
Neutral	125	326	416	242	107
Sad	155	307	235	356	86
Surprise	56	147	99	92	403
Logistic Regression accuracy = 41%					

Fig. 3 Confusion matrix and accuracy for ML models used

Angry	408	265	104	137	52
Happy	78	1370	140	197	40
Neutral	76	317	541	234	48
Sad	79	303	210	507	40
Surprise	29	158	90	75	445
Neural Network accuracy = 55%					
Angry	481	195	102	137	45
Happy	54	1454	127	162	28
Neutral	63	277	616	215	45
Sad	65	268	187	598	21
Surprise	27	125	79	65	501
Convolutional Neural Network accuracy = 61%					
Angry	622	92	94	92	60
Happy	63	1428	117	185	32
Neutral	53	178	783	137	65
Sad	91	181	100	708	59
Surprise	32	76	65	40	584
MobileNet accuracy = 70%					

Fig. 4 Confusion matrix and accuracy for DL models used

be used as an input. Table 1 shows testing accuracy calculated using Eq. (1) of machine learning models used for emotion deduction.

Table 1 Testing accuracy of machine learning-based models

S. No.	Models used	Testing accuracy (%)
1	Gaussian Naïve Bayes	29
2	Multinomial Naïve Bayes	31
3	K-nearest neighbours	38
4	Logistic regression	41
5	Support vector machine	51

Table 2 Increasing order of testing score with corresponding models

S. No.	Models used	Testing accuracy (%)
1	Neural network	55
2	Convolutional neural network	61
3	MobileNet	70

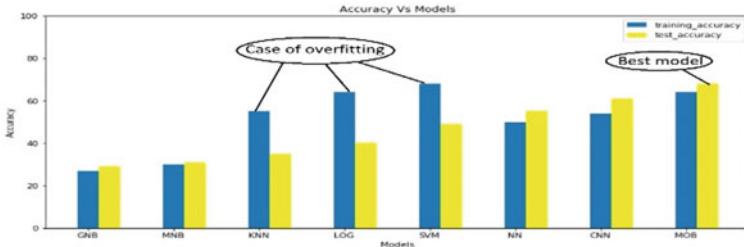


Fig. 5 Both training accuracy and testing accuracy of various models

5.2 Deep Learning-based Models Accuracy

The machine learning-based algorithms were unable to provide good accuracy scores for both training as well as test images, so approach was shifted towards deep learning-based models.

Table 2 shows the testing accuracy calculated using Eq. (1) for various used deep learning models. Least accuracy obtained using Gaussian Naïve Bayes machine learning algorithm and best accuracy is obtained using MobileNet transfer learning algorithm. Figure 5 represents the graph depicting both training accuracy and testing accuracy of various models.

6 Conclusion

The task of human emotion detection using facial expression was best accomplished by deep learning-based MobileNet model with an accuracy level of 70%. The performance for machine learning-based KNN, logistic regression and SVM algorithms was also studied, and they were found to be less efficient because of considerable difference between testing and training accuracy as well as low testing accuracy. DL-based approaches give more consistent results between both training and testing images. Out of the three different DL models, the transfer learning-based MobileNet model outperforms both other NN and CNN-based models with 70% test images accuracy.

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. Kołakowska A, Landowska A, Szwoch M, Szwoch W, Wróbel M (2014) Emotion recognition and its applications. *Adv Intell Syst Comput* 300:51–62. https://doi.org/10.1007/978-3-319-08491-6_5
3. Chiranjeevi P, Gopalakrishnan V, Moogi P (2015) Neutral face classification using personalized appearance models for fast and robust emotion detection. *IEEE Trans Image Process* 24(9):2701–2711. <https://doi.org/10.1109/TIP.2015.2421437>
4. Soleymani M, Asghari-Esfeden S, Fu Y, Pantic M (2016) Analysis of EEG signals and facial expressions for continuous emotion detection. *IEEE Trans Affect Comput* 7(1):17–28. <https://doi.org/10.1109/TAFFC.2015.2436926>
5. Chauhan NK, Singh K (2018) A review on conventional machine learning versus deep learning. In: 2018 international conference on computing, power and communication technologies (GUCON), Greater Noida, Uttar Pradesh, India, pp 347–352. <https://doi.org/10.1109/GUCON.2018.8675097>
6. Oheix J (2018) Face expression recognition dataset. Retrieved from www.kaggle.com/jonathanohex/face-expression-recognition-dataset
7. Gutiérrez PA, Hervás-Martínez C, Martínez-Estudillo FJ (2011) Logistic regression by means of evolutionary radial basis function neural networks. *IEEE Trans Neural Netw* 22(2):246–263. <https://doi.org/10.1109/TNN.2010.2093537>
8. Yang F (2018) An implementation of Naive Bayes classifier. In: 2018 international conference on computational science and computational intelligence (CSCI), Las Vegas, NV, USA, pp 301–306. <https://doi.org/10.1109/CSCI46756.2018.00065>
9. Zhang S, Li X, Zong M, Zhu X, Wang R (2018) Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans Neural Netw Learn Syst* 29(5):1774–1785. <https://doi.org/10.1109/TNNLS.2017.2673241>
10. PD WM, Haryoko (2019) Optimization of parameter support vector machine (SVM) using genetic algorithm to review Go-Jek's services. In: 2019 4th international conference on information technology, information systems and electrical engineering (ICITISEE), Yogyakarta, Indonesia, pp 301–304. <https://doi.org/10.1109/ICITISEE48480.2019.9003894>
11. Verma A, Singh P, Rani Alex JS (2019) Modified convolutional neural network architecture analysis for facial emotion recognition. In: 2019 international conference on systems, signals and image processing (IWSSIP), Osijek, Croatia, pp 169–173. <https://doi.org/10.1109/IWSIP.2019.8787215>
12. Samudre P, Shende P, Jaiswal V (2019) Optimizing performance of convolutional neural network using computing technique. In: 2019 IEEE 5th international conference for convergence in technology (I2CT), Bombay, India, pp 1–4. <https://doi.org/10.1109/I2CT45611.2019.9033876>
13. Zhou Y, Liu Y, Han G, Fu Y (2019) Face recognition based on the improved MobileNet. In: 2019 IEEE symposium series on computational intelligence (SSCI), Xiamen, China, pp 2776–2781. <https://doi.org/10.1109/SSCI44817.2019.9003100>

Chapter 42

An Improvised Particle Swarm Optimization Using Balanced Local Best



Bubul Doley, Arpita Nath Boruah, Sazidur Rahman, Saroj Kumar Biswas, Manomita Chakraborty, Sunita Sarkar, and Biswajit Purkayastha

1 Introduction

In recent times, machine learning is the most promising and attractive area of research. Machine learning analyses the given data, identifies patterns and makes decisions with less human interaction. It plays a significant role in various fields like medical, business and banking sectors. The main aim is to build an optimized model for decision making for nonlinear problem. Optimization is one of the most essential components of many machine learning approaches. Optimized machine learning approaches play an important role in extracting knowledge from a large volume of data.

In the present era, for finding the global optimal solution the most widely applied technique is the stochastic search. Various global optimizers came in existence such as evolutionary algorithm (EA), genetic algorithm (GA), ant colony (AC) and particle swarm optimization (PSO). Among these, PSO is the simplest algorithm and contributes to a new research area called swarm intelligence [1]. Inspired by bird flocking and schooling in nature, PSO is a meta-heuristic population-based stochastic search algorithm. The basic idea of PSO is inspired by a flying swarm of birds searching for food. PSO has already been successfully applied in many areas. Easiness in its implementation is the main advantage of PSO compared to the evolutionary algorithms. But PSO has a slow rate of convergence compared to the optimization techniques, and also for complex multimodal problems, there is the

B. Doley · A. N. Boruah (✉) · S. Rahman · S. K. Biswas · M. Chakraborty · B. Purkayastha
Department of Computer Science and Engineering, National Institute of Technology Silchar,
Silchar, Assam 788010, India

B. Purkayastha
e-mail: biswajit@nits.ac.in

S. Sarkar
Department of Computer Science, Assam University, Silchar, Assam, India

problem of local optimum value. Therefore to overcome these drawbacks, this paper proposes an improved PSO named improvised particle swarm optimization using balanced local best (IPSOB-LBEST), which tunes the components of the velocity equation to overcome the problems like premature convergence, getting trapped in local maxima/minima and hence produces better, faster and more accurate results. The paper is systematically arranged as, Sect. 2 gives an overview of the related works of PSO, Sect. 3 explains basic PSO, Sect. 4 describes the proposed methodology in details, Sect. 5 discusses experimental results. Section 6 draws conclusion.

2 Literature Review

Lots of research for PSO and its modified algorithms have already been done. Sun et al. [2] described the optimization problems in machine learning and their applications. Oliveira et al. [3] gave an overview of various optimization techniques used in machine learning. Poli et al. [4] put forwarded an overview of PSO. Khan et al. [5] put forwarded the PSO algorithm with certain modification to adjust with the updating parameters of the algorithm and hence kept the balance between exploration and exploitation searches. Alhussein et al. [6] proposed a modified PSO with two main changes to the original PSO. The proposed method made modification to the velocity of a particle and if in case the summation of the velocity and position vector results in breaching the boundary limits of search space, then particle velocity needs to be penalized. He et al. [7] put forwarded a parameters selection based on its performance of fault diagnosis method for PSO algorithm. Wen [8] designed a dynamic inertia weight with a decrease function which improved the performance of the PSO. Rui [9] modified PSO by constructing an adaptive inertia weight and adding a new mutation operator. Li [10] modified PSO by inducing selection operator of GA and using chaos in classical PSO. Li et al. [11] forwarded a new swarm which was generated by the contest of two Swarms evolution where one swarm used the linear decreasing weight and the other swarm adopted the random inertia weight.

3 The Original Version of PSO

The initial PSO has a population of particles that are moved around in the search space according to a few simple formulae line velocity and position to find the accurate position. To balance the exploration and exploitation, local search process and global search process are applied. The fitness function is used to evaluate the fitness value associated with each particle. The position and velocity equation of PSO are used to move the particles around the search space.

$$\text{Velocity: } v_i^{t+1} = wv_i^t + c_1r_1^t[P_{\text{best},i}^t - x_i^t] + c_2r_2^t[G_{\text{best}} - x_i^t] \quad (1)$$

$$\text{Position: } x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

In (1), r_1 and r_2 are two random numbers from uniform distribution between 0 and 1, c_1 and c_2 are the positive acceleration constants which are used to level the contribution of cognitive and social components, respectively [12, 13], w is inertia weight known as the controlling parameter [14]. v_i^t velocity vector of the particle i at time t . x_i^t position vector of particle i at time t . P_{best, i^t} personal best of particle i found from initialization through time t . G_{best} global best of all particles found from initialization through time t . In (2), x_i^{t+1} position vector of i th particle at time $t + 1$. x_i^t position vector of i th particle at time t and v_i^{t+1} velocity vector of the i th particle at time $t + 1$.

4 Proposed Methodology

The premature convergence and getting trapped in local maxima/minima are some of the drawbacks of the standard PSO. To overcome these drawbacks, the proposed method IPSOB-LBEST has tuned the components of the velocity equation and thereby produce better and more accurate results. It is achieved by modifying selection of weight (w), random values $r1$ and $r2$ and the updating the cognitive component. The diagrammatic representation of proposed IPSOB-LBEST is shown in Fig. 1.

Analysis of Weight (w): Weight (w) plays the role of controlling the individual velocity of each particle. Greater value of w means the particles will have greater inertia and the search space increases at the cost of slower convergence and hence move away from local and global optimum values. Whereas smaller w value will lead to higher dependency towards the social and cognitive component and increases the convergence rate at the cost of reduced search space increasing the chance of getting caught up in a local optimum. It is found that the optimum value is obtained comparatively faster when w is in the range (0.5, 0.9). Therefore, starting off with higher value of w , i.e. 0.9 to have higher searching abilities and reduce it gradually with lower bound 0.5. Doing this gives the particles the much-needed velocity at the beginning stages that help them to explore more search space and to slow down when they reach the optimum value.

Particle Initialization: Particles are initialized with uniformly distributed random variables, i.e. dividing the entire search space into some same size of blocks and each block contains same number of particles.

For random values $r1$ and $r2$: Chaotic variables are used to relatively regulate the parameters used for updating the particle velocity. The use of chaotic variables in optimization search prevents the evolutionary algorithms falling into local optimum.

The chaotic optimization of parameters $r1$ and $r2$ is given in (3).

$$r_i(t) = 4.0 * r_i(t) * (1 - r_i(t))$$

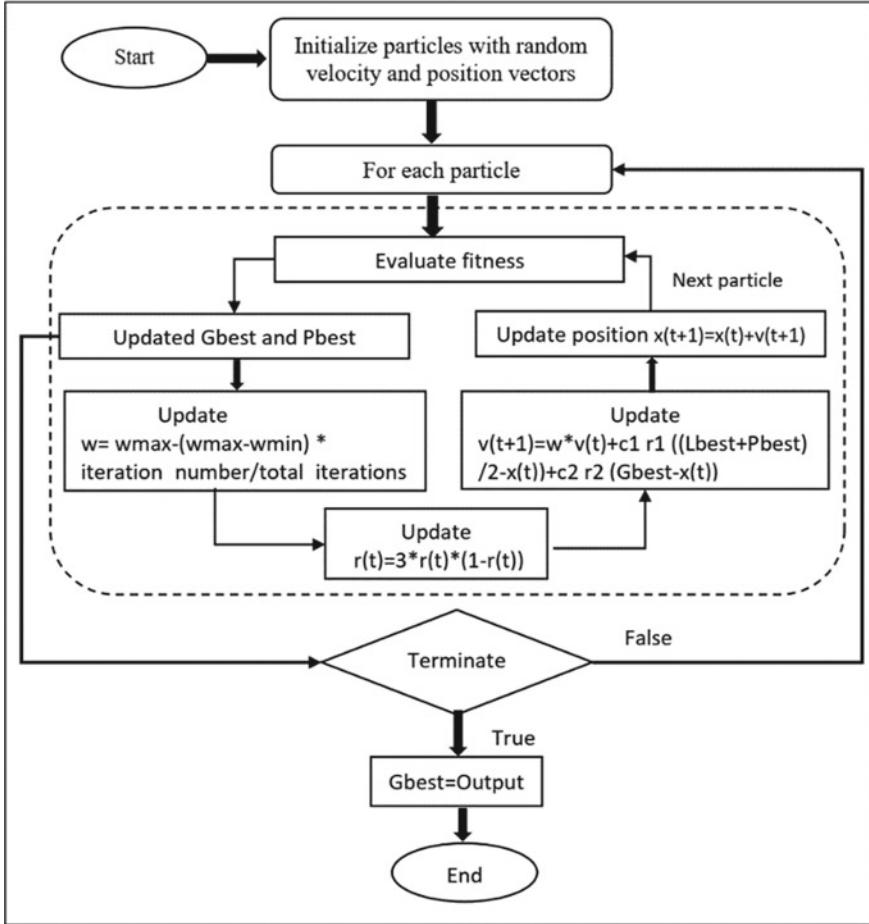


Fig. 1 Improvised particle swarm optimization using balanced local best (IPSOB-LBEST)

$$r_i(t) \in (0, 1.0), i = 1, 2 \quad (3)$$

For cognitive component: Let the local best (L_{best}) play the role in deciding the movement of the particles in the cognitive component. While updating the cognitive component of each particle, the average of the best particle value of the last iteration (L_{best}) and the best of the particle (P_{best}) is taken under consideration.

Equation (4) gives the updated velocity

$$\begin{aligned}
 v(t + 1) = & w * v(t) + c_1 r_1 \left(\frac{(L_{\text{best}} + P_{\text{best}})}{2} - x(t) \right) \\
 & + c_2 r_2 (G_{\text{best}} - x(t))
 \end{aligned} \quad (4)$$

5 Results and Discussions

Some of the benchmark minimization functions are considered to compare the IPSOB-LBEST with the standard PSO and the velocity clamping model.

1. Ackley Function, range: $(-5 \leq x, y \leq 5)$

$$f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos cx_i\right) + a + \exp(1)$$

2. Holder Table Function, range: $(-10 \leq x, y \leq 10)$

$$f(x) = -\left| \sin(x_1) \cos(x_2) \exp\left(1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right) \right|$$

3. Himmelblau Function, range: $(-5 \leq x, y \leq 5)$

$$f(x) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

4. Beale Function, range: $(-4.5 \leq x, y \leq 4.5)$

$$\begin{aligned} f(x) = & (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 \\ & + (2.625 - x_1 + x_1 x_2^3)^2 \end{aligned}$$

5. Rastrigin Function, range: $(-5.12 \leq x, y \leq 5.12)$

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$$

6. Matyas Function, range: $(-10 \leq x, y \leq 10)$

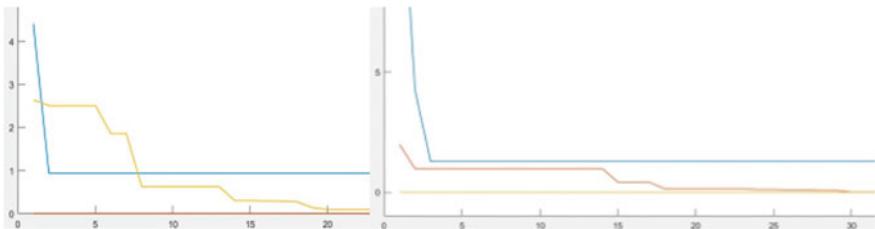
$$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$$

From Table 1, it is observed that for almost all the considered minimization functions, the proposed model IPSOB-LBEST performs better producing more accurate and precise value than the standard PSO and the velocity clamping model.

Figures 2, 3, 4, 5 and 6 depict the convergence characteristics of the different minimization functions considered for comparison of IPSOB-LBEST with the standard PSO and velocity clamping model. From the mentioned figures, it is observed that in comparison with the standard PSO and the velocity clamping model, IPSOB-LBEST converges faster for all the functions.

Table 1 Performance comparison

Function name	Actual optimum	Measures	Standard PSO	Velocity clamping model	IPSOB-LBEST
Ackley	(0, 0)	Calc. value	(−0.009, −0.1509)	(0, 0)	(−0.0101, 0.0107)
		Mean	−0.039975	0	0.00015
		Std. deviation	0.213024	0	0.0007646
Holder table	($\pm 8.05502, \pm 9.66459$)	Calc. value	(−8.0528, 10)	(14.4665, −9.145)	(−7.9992, 9.7004)
		Mean	0.8891925	0.9279825	0.8276925
		Std. deviation	9.02679464	11.9322343	8.8498296
Himmelblau function	(3, 2)	Calc. value	(3.0898, 1.6729)	(3.5848, −1.8479)	(3, 2)
		Mean	1.7141	1.83025	1.7508
Beale function	(3, 0.5)	Calc. value	(2.8385, 0.5179)	(3.2629, 0.5581)	(3.0031, 0.5001)
		Mean	1.160855	1.354778	1.1525
		Std. deviation			
Rastrigin function	(0, 0)	Calc. value	(1.005, 0.0142)	(−0.0988, −0.1521)	(0, 0)
		Mean	0.37855	−0.062725	0
		Std. deviation	0.51244	0.0681516	0
Matyas function	(0, 0)	Calc. value	(0.3733, 0.233)	(−0.1681, −0.1244)	(0.0367, 0.0387)
		Mean	0.151575	−0.073125	0.01885
		Std. deviation	0.1670209	0.2204604	0.056558

**Fig. 2** Graphical representation of Ackley and Himmelblau Function for IPSOB-LBEST versus standard PSO and velocity clamping model

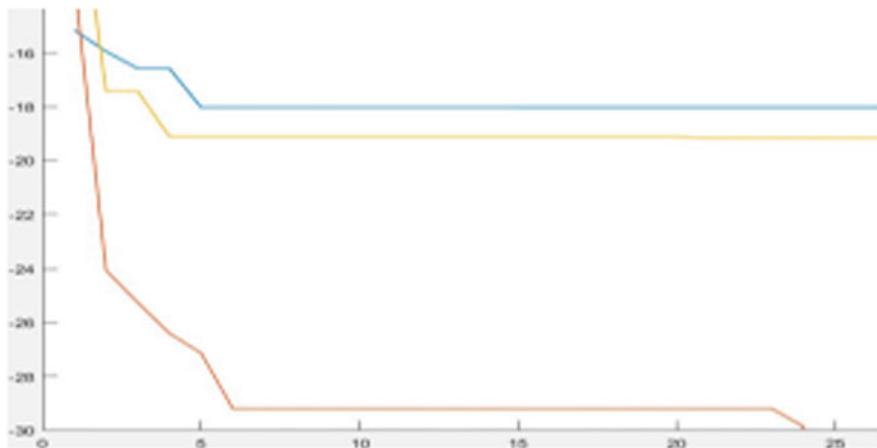


Fig. 3 Graphical representation of holder table function for IPSOB-LBEST versus standard PSO and velocity clamping model



Fig. 4 Graphical representation of Beale function for IPSOB-LBEST versus standard PSO and velocity clamping model

6 Conclusion

An improvised model has been proposed with three modifications over the standard PSO model. The modifications are targeted towards improvement of convergence rate and also increasing the search space. The IPSOB-LBEST changes the value of inertial weight, ' w ', using the known result that the optimum value is obtained easily when w lies between 0.9 and 0.5. The random parameters, r_1 and r_2 , are changed according to a chaotic model, which brings uniformity towards the randomness.

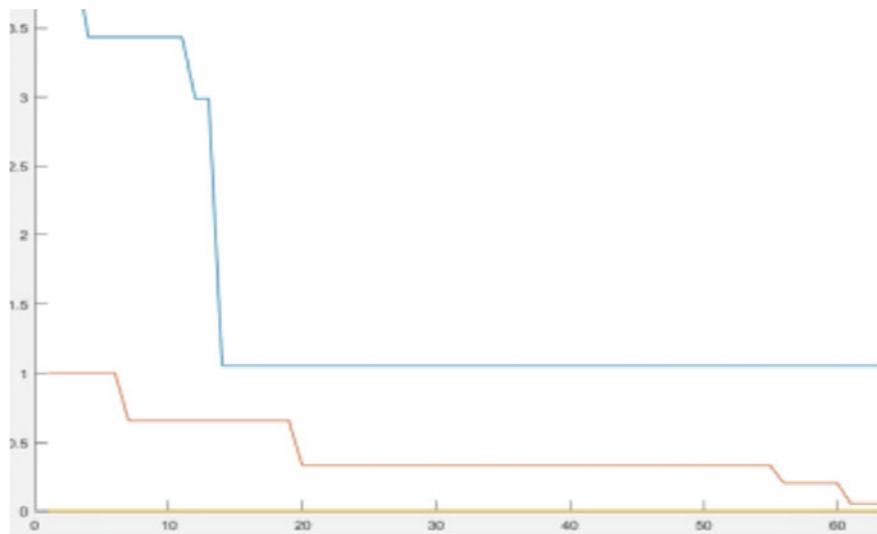


Fig. 5 Graphical representation of Rastrigin FUNCTION for IPSOB-LBEST versus standard PSO and velocity clamping model

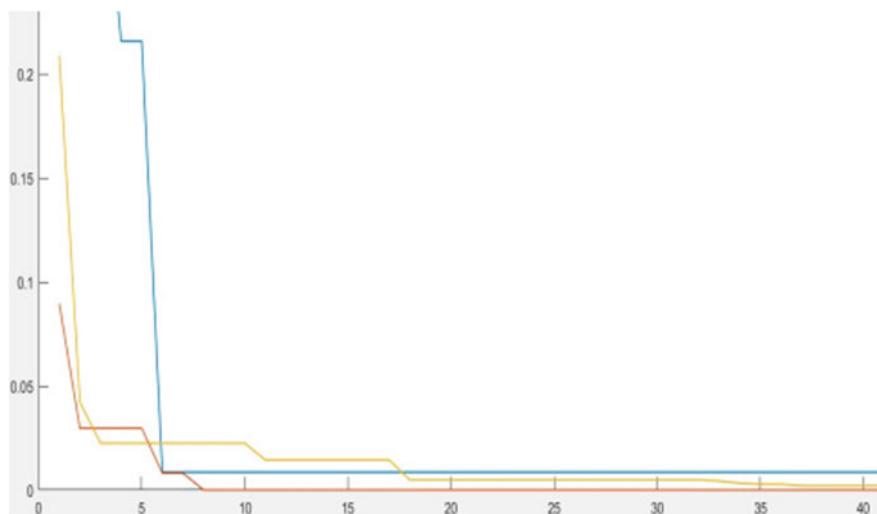


Fig. 6 Graphical representation of Matyas function for IPSOB-LBEST versus standard PSO and velocity clamping model

Thereby the cognitive component is changed. The IPSOB-LBEST has considered the local best value in deciding the movement of the particle. The IPSOB-LBEST performs better than the standard model for multimodal functions. The efficiency of the model can be further increased by tuning the values of acceleration coefficients, c_1 and c_2 .

References

1. Yang XS (2008) Nature-inspired metaheuristic algorithms, 1st edn. Luniver Press, UK
2. Sun S, Cao Z, Zhu H, Zhao J (2019) A survey of optimization methods from a machine learning perspective. *IEEE Trans Cybern*
3. Oliveira P, Portela F, Santos MF, Abelha A, Machado J Machine-learning an overview of optimization techniques. *Recent Adv Comput Sci*
4. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization an overview. *Swarm Intell* 1:33–57
5. Khan S, Kamran M, Rehman OU, Liu L, Yang S (2018) A modified PSO algorithm with dynamic parameters for solving complex engineering design problem. *Int J Comput Math* 95(11):2308–2329
6. Alhussein M, Haider SI (2015) Improved particle swarm optimization based on velocity clamping and particle penalization. In: 2015 3rd international conference on artificial intelligence, modelling and simulation (AIMS), pp 61–64
7. He Y, Ma WJ, Zhang JP (2016) The parameters selection of PSO algorithm influencing on performance of fault diagnosis. *Proc. MATEC Web Conf* 63:02019
8. Wen L (2012) A modified dynamic particle swarm optimization algorithm. *Fifth Int Symp Comput Intell Des* 2012:432–435
9. Rui S (2016) A modified adaptive particle swarm optimization algorithm. *Int Conf Comput Intell Secur* 2196:209–214
10. Li J (2014) A modified particle swarm optimization based on genetic algorithm and chaos. In: Proceeding of the 11th world congress on intelligent control and automation shenyang, China, June 29–July 4 2014
11. Li W, Jianfeng Z, Xin L, Guoqiang S (2015) Particle swarm optimization algorithm based on two swarm evolution. *Control Decision Conf IEEE*:1200–1204
12. Eberhart RC, Shi YH (2001) Particle swarm optimization developments application and resources. *Proc. IEEE Congr Evol Comput* 1:81–86
13. Shi YH, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of IEEE international conference on evolutionary computation, Anchorage, AK, pp 69–73
14. Ratnaweera A, Saman K, Halgamuge KS (2004) Self-organizing hierarchical particle swarm optimizer with time varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255