Lecture Notes in Networks and Systems 892

Swagatam Das
Snehanshu Saha
Carlos A. Coello Coello
Jagdish C. Bansal *Editors*

Advances in Data-Driven Computing and Intelligent Systems

Selected Papers from ADCIS 2023, Volume 2

Springer

Lecture Notes in Networks and Systems

Volume 892

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

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

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Swagatam Das · Snehanshu Saha ·
Carlos A. Coello Coello · Jagdish C. Bansal
Editors

Advances in Data-Driven Computing and Intelligent Systems

Selected Papers from ADCIS 2023, Volume 2

Springer

Editors
Swagatam Das
Electronics and Communication Sciences
Unit
Indian Statistical Institute
Kolkata, West Bengal, India

Carlos A. Coello Coello
Department of Computer Science
CINVESTAV-IPN
Mexico City, Mexico

Snehanshu Saha
Department of Computer Science
and Information Systems
Birla Institute of Technology and Science,
Pilani
Sancoale, Goa, India

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

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-981-99-9520-2 ISBN 978-981-99-9521-9 (eBook)
https://doi.org/10.1007/978-981-99-9521-9

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

Paper in this product is recyclable.

Preface

This book contains outstanding research papers as the proceedings of the 2nd International Conference on Advances in Data-driven Computing and Intelligent Systems (ADCIS 2023) organized by BITS Pilani, K. K. Birla Goa Campus, India, and co-organized by National Forensic Sciences University (NFSU), under the technical sponsorship of the Soft Computing Research Society, India. The conference is conceived as a platform for disseminating and exchanging ideas, concepts, and results of researchers from academia and industry to develop a comprehensive understanding of the challenges of the advancements of intelligence in computational viewpoints. This book will help in strengthening congenial networking between academia and industry. We have tried our best to enrich the quality of the ADCIS 2023 through the stringent and careful peer-review process. This book presents novel contributions to Intelligent Systems and serves as reference material for Data-Driven Computing.

We have tried our best to enrich the quality of the ADCIS 2023 through a stringent and careful peer-review process. ADCIS 2023 received many technical contributed articles from distinguished participants from home and abroad. ADCIS 2023 received 1076 research submissions from 18 different countries, viz., Austria, Bangladesh, Fiji, Germany, Greece, Iraq, Italy, Japan, Malaysia, Malta, Morocco, Russia, Saudi Arabia, Serbia, UAE, United Kingdom, and Vietnam. After a very stringent peer-reviewing process, only 162 high-quality papers were finally accepted for the presentation and the final proceedings.

This book presents the second volume. It includes 40 research papers in data science and applications and serves as reference material for advanced research.

Kolkata, India	Swagatam Das
Goa, India	Snehanshu Saha
Mexico City, Mexico	Carlos A. Coello Coello
New Delhi, India	Jagdish C. Bansal

Contents

Deep Learning Models for Classification of Remotely Sensed Data of Sugarcane 1
Mansi Kambli and Bhakti Palkar

Detection and Analysis of Wormhole Attacks in the AODV Routing Protocol with IEEE 802.11p for the Internet of Vehicles 17
Tanuj Meshram and Mou Dasgupta

A Systematic Review of NLP Applications in Clinical Healthcare: Advancement and Challenges 31
Rachit Garg and Anshul Gupta

An Investigational Analysis of Automatic Speech Recognition on Deep Neural Networks and Gated Recurrent Unit Model 45
M. Soundarya and S. Anusuya

Matched Filter and Kirsch's Template Based Approach for Retinal Vessel Segmentation 61
Sonali Dash, Kanwarpreet Kaur, and Gaurav Bathla

Prediction of Abnormality in Kidney Function Using Classification Techniques and Fuzzy Systems 71
Mynapati Lakshmi Prasudha, Sukhavasi Vidyullatha, and Yeluri Divya

Implementation of Parallel Applications on the Hypercube Topology by Using Multistage Network 81
Qusay S. Alsaffar and Leila Ben Ayed

Integrating Artificial Intelligence for Adaptive Decision-Making in Complex System 95
Ajay Verma and Nisha Singhal

Qualitative Research Reasoning on Dementia Forecast Using Machine Learning Techniques 107
Tanvi Kapdi and Apurva Shah

Implementation of Vision Transformers on SPECT Heart Dataset: A Comparative Study 119
Poonam Verma, Vikas Tripathi, and Bhaskar Pant

CSR U-Net: A Novel Approach for Enhanced Skin Cancer Lesion Image Segmentation 129
V. Chakkarapani and S. Poornapushpakala

Automatic Detection and Classification System for Mesothelioma Cancer Using Deep Learning Models with HPO 143
Apeksha Koul, Rajesh K. Bawa, and Yogesh Kumar

A Systematic Literature Survey on IoT in Health Care: Security and Privacy Threats 157
Aryan Bakliwal, Deepak Panwar, and G. L. Saini

Hybrid Deep Learning Framework for Glaucoma Detection Using Fundus Images 173
Royce Dcunha, Aaron Rodrigues, Cassandra Rodrigues, and Kavita Sonawane

Sunflower Optimization with Elite Learning Strategy (SFO-ELS) for Antenna Selection in Massive MIMO Subarray Switching Architecture 185
Snehal Gaikwad and P. Malathi

Machine Learning Models for Human Activity Recognition: A Comparative Study 199
Anshul Sheoran, Ritu Boora, and Manisha Jangra

A K-Means Variation Based on Careful Seeding and Constrained Silhouette Coefficients 211
Libero Nigro, Franco Cicirelli, and Francesco Pupo

Satellite Image Analysis in Health Care—A Systematic Review 227
Bhushan Pawar, Vijay Prakash, Lalit Garg, Charles Galdies, Sandra Buttigieg, and Neville Calleja

Achieving Sustainability in Supply Chain During Disruption Times: Role of Industry 4.0 239
Namit Shrivastava and Manoj K. Srivastava

Emotionally Engaged Neurosymbolic AI for Usable Password Generation 251
Sumitra Biswal

Exploring the Deep Learning Techniques in Plant Disease Detection: A Review of Recent Advances 265
Saurabh Singh and Rahul Katarya

A Robust Driver Distraction Estimation Technique for ADAS Applications 279
Sriman Sathish, S. Ashwin, S. Manish, Nishanth S. Shukapuri, Mayur S. Gowda, and Viswanath Talasila

A Hybrid Approach for Depression Detection Using Word Embedding, Naive Bayes and Bi-LSTM Models 295
Jyoti Singh, Ishan Mangotra, Minni Jain, and Amita Jain

A Deep Learning Approach to Computer-Aided Screening and Early Diagnosis of Middle Ear Disease 309
Ankit Kumar Singh, Ajay Singh Raghuvanshi, Anmol Gupta, and Harsh Dewangan

Pay-by-Palm: A Contactless Payment System 329
Sridevi Saralaya, Pravin Kumar, Mohammed Shehzad, Mohammed Nihal, and Pragnya Nagure

Steganalysis of Reversible Digital Watermarking Algorithm Based on LWT and SVD 345
Geeta Sharma and Vinay Kumar

Analyzing the Effectiveness of Image Augmentation for Soybean Crop and Broadleaf Weed Classification 359
Michael Justina and M. Thenmozhi

Iterative Thresholding-Based Shadow Detection Approach for UAV Images 373
Deeksha and Toshanlal Meenpal

A Subtle Design of Prediction Models Using Machine Learning Algorithms for Advocating Selection and Forecasting Sales of Garments: A Case Study 387
Dillip Rout, Bholanath Roy, and Prasanna Kapse

BGKnow-Medical Chatbot: A Hybrid Approach Based on Knowledge Graph and GPT-2 399
Disha Sunil Nikam, D. Nisha Murthy, Sreeramya Dharani Pragada, and H. R. Mamatha

Video Integrity Checking Using X25519 and Nested HMAC with BLAKE2b 411
Linju Lawrence and R. Shreelekshmi

Cloud-Based Skin Cancer Classification: Training and Deploying a Model on AWS 423
Challa Koti Reddy, Chava Pavan Kumar, A. R. P. S. Gowtham, Rajkumar Maharaju, and Rama Valupadasu

An Automotive ECU-Based Forward Collision Prevention System 435
Fariya Islam, Tajruba Tahsin Nileema, Fazle Rabbi Abir,
Tasmia Tahmida Jidney, and Kazi A. Kalpoma

Credit Card Fraud Detection by Using Ensemble Method of Machine Learning 449
Nihar Ranjan, G. S. Mate, A. J. Jadhav, D. H. Patil,
and A. N. Banubakode

APiCroDD: Automated Pipeline for Crop Disease Detection 461
Pawan K. Ajmera, Sanchit M. Kabra, Anish Mall, Ankur Lhila,
and Aaryan Agarwal

TimeGAN for Data-Driven AI in High-Dimensional Industrial Data 473
Felix Neubürger, Yasser Saeid, and Thomas Kopinski

A Deep Learning Framework for Assamese Toxic Comment Detection: Leveraging LSTM and BiLSTM Models with Attention Mechanism 485
Mandira Neog and Nomi Baruah

Security in VANETs with Insider Attack Resistance and Signature Aggregation 499
Vijaya Lode, Kekhelo Lasushe, and Anil Pinapati

Comparative Study of LevelDB and BadgerDB Databases on the Basis of Features and Read/Write Operations 513
Pragya Vaishnav, Linesh Raja, and Aniket Bhange

An Improved Snow Ablation Optimizer for Stabilizing the Artificial Neural Network 525
Pedda Nagyalla Maddaiah and Pournami Pulinthanathu Narayanan

Author Index 537

Editors and Contributors

About the Editors

Swagatam Das received the B.E. Tel.E., M.E. Tel.E (Control Engineering specialization), and Ph.D. degrees, all from Jadavpur University, India, in 2003, 2005, and 2009, respectively. Swagatam Das is currently serving as an associate professor and the head of the Electronics and Communication Sciences Unit of the Indian Statistical Institute, Kolkata, India. His research interests include evolutionary computing and machine learning. Dr. Das has published more than 300 research articles in peer-reviewed journals and international conferences. He is the founding co-editor-in-chief of *Swarm and Evolutionary Computation*, an international journal from Elsevier. He has also served as or is serving as the associate editor of the IEEE Transactions on Cybernetics, Pattern Recognition (Elsevier), Neurocomputing (Elsevier), Information Sciences (Elsevier), IEEE Trans. on Systems, Man, and Cybernetics: Systems, and so on.

Snehanshu Saha holds Master's Degree in Mathematical and Computational Sciences at Clemson University, USA, and Ph.D. from the Department of Applied Mathematics at the University of Texas at Arlington in 2008. He was the recipient of the prestigious Dean's Fellowship during Ph.D. and Summa Cum Laude for being in the top of the class. After working briefly at his Alma matter, Snehanshu moved to the University of Texas El Paso as a regular full-time faculty in the Department of Mathematical Sciences, University of Texas El Paso. Currently, he is a professor of Computer Science and Engineering at PES University since 2011 and heads the Center for AstroInformatis, Modeling, and Simulation. He is also a visiting professor at the Department of Statistics, University of Georgia, USA, and BTS Pilani, India.

Carlos A. Coello Coello (Fellow, IEEE) received the Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996. He is currently a professor with Distinction (CINVESTAV-3F Researcher), Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico. He has authored and co-authored over 500 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multiobjective Problems* (2nd ed., Springer, 2007) and has edited three more books with publishers such as World Scientific and Springer. His publications currently report over 60,000 citations in Google Scholar (his H-index is 96). His major research interests are evolutionary multi-objective optimization and constraint-handling techniques for evolutionary algorithms.

He has received several awards, including the National Research Award (in 2007) from the Mexican Academy of Science (in the area of exact sciences), the 2009 Medal to the Scientific Merit from Mexico City's congress, the Ciudad Capital: Heberto Castillo 2011 Award for scientists under the age of 45, in Basic Science, the 2012 Scopus Award (Mexico's edition) for being the most highly cited scientist in engineering in the five years previous to the award and the 2012 National Medal of Science in Physics, Mathematics, and Natural Sciences from Mexico's presidency (this is the most important award that a scientist can receive in Mexico). He also received the Luis Elizondo Award from the Tecnológico de Monterrey in 2019.

Dr. Jagdish C. Bansal is an Associate Professor (Senior Grade) at South Asian University New Delhi and Visiting Faculty at Maths and Computer Science, Liverpool Hope University UK. He also holds visiting professorship at NIT Goa, India. Dr. Bansal obtained his Ph.D. in Mathematics from IIT Roorkee. Before joining SAU New Delhi, he worked as an Assistant Professor at ABV- Indian Institute of Information Technology and Management Gwalior and BITS Pilani. His Primary area of interest is Swarm Intelligence and Nature Inspired Optimization Techniques. Recently, he proposed a fission-fusion social structure based optimization algorithm, Spider Monkey Optimization (SMO), which is being applied to various problems in the engineering domain. He has published over 70 research papers in various international journals/conferences. He is the Section Editor (editor-in-chief) of the journal *MethodsX* published by Elsevier. He is the series editor of the book series *Algorithms for Intelligent Systems (AIS)*, *Studies in Autonomic, Data-driven and Industrial Computing (SADIC)*, and *Innovations in Sustainable Technologies and Computing (ISTC)* published by Springer. He is also the Associate Editor of *Engineering Applications of Artificial Intelligence (EAAI) and ARRAY* published by Elsevier. He is the general secretary of the Soft Computing Research Society (SCRS). He has also received Gold Medal at UG and PG levels.

Contributors

Fazle Rabbi Abir Ahsanullah University of Science and Technology, Tejgaon, Dhaka, Bangladesh

Aaryan Agarwal Birla Institute of Technology and Science, Pilani, India

Pawan K. Ajmera Birla Institute of Technology and Science, Pilani, India

Qusay S. Alsaffar University of Sfax, National School of Electronics and Telecommunications of Sfax, Sakiet Ezzit, Tunisia

S. Anusuya Saveetha School of Engineering, SIMATS, Chennai, India

S. Ashwin Department of Electronics and Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India

Leila Ben Ayed National School of Computer Science, Sakiet Ezzit, Tunisia

Aryan Bakliwal Manipal University Jaipur, Jaipur, Rajasthan, India

A. N. Banubakode MET COE, Mumbai, India

Nomi Baruah Dibrugarh University, Dibrugarh, Assam, India

Gaurav Bathla CSED, Chandigarh University, Gharuan, Mohali, India

Rajesh K. Bawa Punjabi University, Patiala, Punjab, India

Aniket Bhange Debit Circle SDN BHD, Kuala Lumpur, Malaysia

Sumitra Biswal Bosch Global Software Technologies (BGSW), Bosch, India

Ritu Boora Guru Jambheshwar University of Science and Technology, Hisar, India

Sandra Buttigieg Faculty of Health Sciences, University of Malta, Msida, Malta

Neville Calleja Faculty of Medicine and Surgery, University of Malta, Msida, Malta

V. Chakkarapani Sathyabama Institute of Science and Technology (Deemed to be University), Chennai, Tamil Nadu, India

Franco Cicirelli CNR—National Research Council of Italy, Institute for High Performance Computing and Networking (ICAR), Rende, Italy

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

Sonali Dash CSED, Chandigarh University, Gharuan, Mohali, India

Royce Dcunha St. Francis Institute of Technology, Mumbai, India

Deeksha Department of Electronics and Communication, National Institute of Technology Raipur, Raipur, India

Harsh Dewangan Department of Electronics and Communication, NIT Raipur, Raipur, India

Yeluri Divya BVRIT HYDERABAD College of Engineering for Women, Hyderabad, Telangana, India

Snehal Gaikwad D. Y. Patil College of Engineering, Pune, India

Charles Galdies Environmental Management and Plan Division, Institute of Earth Systems, University of Malta, Msida, Malta

Lalit Garg Department of Computer Information Systems, Faculty of Information and Communication Technology, University of Malta, Msida, MSD, Malta

Rachit Garg Jain University, Bangalore, India

Mayur S. Gowda Centre for Imaging Technologies, Ramaiah Institute of Technology, Bengaluru, India

A. R. P. S. Gowtham NIT Warangal, Warangal, Telangana, India

Anmol Gupta Department of Electronics and Communication, NIT Raipur, Raipur, India

Anshul Gupta SP Jain School of Global Management, Dubai, UAE

Fariya Islam Ahsanullah University of Science and Technology, Tejgaon, Dhaka, Bangladesh

A. J. Jadhav Department of Information Technology, Rajarshi Shahu College of Engineering, Pune, India

Amita Jain Netaji Subhas University of Technology, Delhi, India

Minni Jain Delhi Technological University, Delhi, India

Manisha Jangra Guru Jambheshwar University of Science and Technology, Hisar, India

Tasmia Tahmida Jidney Ahsanullah University of Science and Technology, Tejgaon, Dhaka, Bangladesh

Michael Justina Department of Computer Science and Engineering, SRMIST, Chennai, India

Sanchit M. Kabra Birla Institute of Technology and Science, Pilani, India

Kazi A. Kalpoma Ahsanullah University of Science and Technology, Tejgaon, Dhaka, Bangladesh

Mansi Kambli Computer department, K. J. Somaiya College of Engineering, Somaiya Vidyavihar University, Mumbai, India

Tanvi Kapdi The Maharaja Sayajirao University Baroda, Vadodara, India

Prasanna Kapse Medi-Caps University, Indore, India

Rahul Katarya Delhi Technological University, New Delhi, India

Kanwarpreet Kaur ECED, Chandigarh University, Gharuan, Mohali, India

Thomas Kopinski South Westphalia University of Applied Sciences, Meschede, Germany

Apeksha Koul Punjabi University, Patiala, Punjab, India

Chava Pavan Kumar NIT Warangal, Warangal, Telangana, India

Pravin Kumar St Joseph Engineering College, Mangaluru, Karnataka, India; Visvesvaraya Technological University, Belagavi, Karnataka, India

Vinay Kumar NIC, Govt. of India, Delhi, India

Yogesh Kumar Pandit Deendayal Energy University, Gandhinagar, Gujarat, India

Mynapati Lakshmi Prasudha BVRIT HYDERABAD College of Engineering for Women, Hyderabad, Telangana, India

Kekhelo Lasushe Department of Computer Science and Engineering, National Institute of Technology, Calicut, Kerala, India

Linju Lawrence Department of Computer Science and Engineering, College of Engineering Trivandrum Affiliated to APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala, India

Ankur Lhila Birla Institute of Technology and Science, Pilani, India

Vijaya Lode Department of Computer Science and Engineering, National Institute of Technology, Calicut, Kerala, India

Pedda Nagyalla Maddaiah Department of Computer Science and Engineering, National Institute of Technology Calicut, Calicut, Kerala, India

Rajkumar Maharaju NIT Warangal, Warangal, Telangana, India

P. Malathi D. Y. Patil College of Engineering, Pune, India

Anish Mall Birla Institute of Technology and Science, Pilani, India

H. R. Mamatha PES University, Banashankari, Bengaluru, Karnataka, India

Ishan Mangotra Netaji Subhas University of Technology, Delhi, India

S. Manish Department of Electronics and Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India

G. S. Mate Department of Information Technology, Rajarshi Shahu College of Engineering, Pune, India

Toshanlal Meenpal Department of Electronics and Communication, National Institute of Technology Raipur, Raipur, India

Tanuj Meshram Department of Computer Applications, National Institute of Technology Raipur, Raipur, India

Pragnya Nagure St Joseph Engineering College, Mangaluru, Karnataka, India; Visvesvaraya Technological University, Belagavi, Karnataka, India

Pournami Pulinthanathu Narayanan Department of Computer Science and Engineering, National Institute of Technology Calicut, Calicut, Kerala, India

Mandira Neog Dibrugarh University, Dibrugarh, Assam, India

Felix Neubürger South Westphalia University of Applied Sciences, Meschede, Germany

Libero Nigro University of Calabria, DIMES, Rende, Italy

Mohammed Nihal St Joseph Engineering College, Mangaluru, Karnataka, India; Visvesvaraya Technological University, Belagavi, Karnataka, India

Disha Sunil Nikam PES University, Banashankari, Bengaluru, Karnataka, India

Tajruba Tahsin Nileema Ahsanullah University of Science and Technology, Tejgaon, Dhaka, Bangladesh

D. Nisha Murthy PES University, Banashankari, Bengaluru, Karnataka, India

Bhakti Palkar Computer department, K. J. Somaiya College of Engineering, Somaiya Vidyavihar University, Mumbai, India

Bhaskar Pant Computer Science and Engineering, Graphic Era University Deemed, Dehradun, India

Deepak Panwar Manipal University Jaipur, Jaipur, Rajasthan, India

D. H. Patil Department of Information Technology, Rajarshi Shahu College of Engineering, Pune, India

Bhushan Pawar Department of Computer Information Systems, Faculty of Information and Communication Technology, University of Malta, Msida, MSD, Malta

Anil Pinapati Department of Computer Science and Engineering, National Institute of Technology, Calicut, Kerala, India

S. Poornapushpakala Sathyabama Institute of Science and Technology (Deemed to be University), Chennai, Tamil Nadu, India

Sreeramya Dharani Pragada PES University, Banashankari, Bengaluru, Karnataka, India

Vijay Prakash Department of Computer Information Systems, Faculty of Information and Communication Technology, University of Malta, Msida, MSD, Malta; Department of Computer Science, Graphics Era Hill University, Dehradun, India

Francesco Pupo University of Calabria, DIMES, Rende, Italy

Ajay Singh Raghuvanshi Department of Electronics and Communication, NIT Raipur, Raipur, India

Linesh Raja Department of Computer Applications, Manipal University Jaipur, Jaipur, Rajasthan, India

Nihar Ranjan Department of Information Technology, Rajarshi Shahu College of Engineering, Pune, India

Challa Koti Reddy NIT Warangal, Warangal, Telangana, India

Aaron Rodrigues St. Francis Institute of Technology, Mumbai, India

Cassandra Rodrigues St. Francis Institute of Technology, Mumbai, India

Dillip Rout C.V. Raman Global University, Bhubaneswar, India

Bholanath Roy Maulana Azad National Institute of Technology, Bhopal, India

Yasser Saeid South Westphalia University of Applied Sciences, Meschede, Germany

G. L. Saini Manipal University Jaipur, Jaipur, Rajasthan, India

Sridevi Saralaya St Joseph Engineering College, Mangaluru, Karnataka, India;
Visvesvaraya Technological University, Belagavi, Karnataka, India

Sriman Sathish Department of Electronics and Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India

Apurva Shah The Maharaja Sayajirao University Baroda, Vadodara, India

Geeta Sharma Jagan Institute of Management Studies, Rohini, Delhi, India

Mohammed Shehzad St Joseph Engineering College, Mangaluru, Karnataka, India;
Visvesvaraya Technological University, Belagavi, Karnataka, India

Anshul Sheoran Guru Jambheshwar University of Science and Technology, Hisar, India

R. Shreelekshmi Department of Computer Applications, College of Engineering Trivandrum Affiliated to APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala, India

Namit Shrivastava Birla Institute of Technology and Science, Pilani, India

Nishanth S. Shukapuri Department of Electronics and Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India

Ankit Kumar Singh Department of Electronics and Communication, NIT Raipur, Raipur, India

Jyoti Singh Netaji Subhas University of Technology, Delhi, India

Saurabh Singh Delhi Technological University, New Delhi, India

Nisha Singhal Department of Mathematics, Indian Institute of Information Technology Bhopal, Bhopal, India

Kavita Sonawane St. Francis Institute of Technology, Mumbai, India

M. Soundarya Saveetha School of Engineering, SIMATS, Chennai, India

Manoj K. Srivastava Management Development Institute, Gurgaon, India

Viswanath Talasila Department of Electronics and Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India;
Centre for Imaging Technologies, Ramaiah Institute of Technology, Bengaluru, India

M. Thenmozhi Department of Computer Science and Engineering, SRMIST, Chennai, India

Vikas Tripathi Computer Science and Engineering, Graphic Era University Deemed, Dehradun, India

Pragya Vaishnav Department of Computer Applications, Manipal University Jaipur, Jaipur, Rajasthan, India

Rama Valupadasu NIT Warangal, Warangal, Telangana, India

Ajay Verma Department of Mechanical Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal, India

Poonam Verma Computer Science and Engineering, Graphic Era University Deemed, Dehradun, India;
School of Computing, Graphic Era Hill University, Dehradun, India

Sukhavasi Vidyullatha BVRIT HYDERABAD College of Engineering for Women, Hyderabad, Telangana, India

Deep Learning Models for Classification of Remotely Sensed Data of Sugarcane

Mansi Kambli and Bhakti Palkar

Abstract The traditional machine learning algorithms are giving way to approaches for deep learning in computer vision, which refers to a computer's capacity to infer meaning from digital images and videos. Sugarcane categorization is important for agricultural management and monitoring. Traditional crop categorization methods based on manual inspection or restricted ground-based data gathering are time-consuming and frequently inaccurate. As a result, an automated and efficient strategy is suggested that requires the use of remote sensing data and the capabilities of deep learning algorithms. A dataset made from multispectral Sentinel imagery is used for the classification of sugarcane. This approach seeks to separate sugarcane-growing regions from other regions in Sentinel-2 images using VGG19, MobileNetV2, and CNN as feature extractors. These findings illustrate the feature extraction utilizing deep learning models with an SVM classifier for sugarcane. By considering variables such as distinct spectral bands, temporal fluctuations, and potential difficulties in separating sugarcane from other land cover types, the objective is to construct and check working of deep learning models for categorizing sugarcane locations using Sentinel-2 data. The sugarcane classification can further be used to find dense and sparse vegetation after the classification is done with deep learning models. The outcomes of this study will help to improve sugarcane categorization techniques and will help farmers, researchers, and agricultural stakeholders make better crop management, yield estimation, and resource optimization decisions in sugarcane farming.

Keywords Deep learning · Sugarcane

M. Kambli (✉) · B. Palkar
Computer department, K. J. Somaiya College of Engineering, Somaiya Vidyavihar University, Mumbai, India
e-mail: mansi.mk@somaiya.edu

B. Palkar
e-mail: bhaktiraul@somaiya.edu

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_1

1 Introduction

Agriculture is main sector that needs to be looked into for the growth of mankind. Sugarcane is a crop that can be used for bioconversion energy and is the second largest crop in India. Sugarcane classification helps in monitoring crop health and warns farmers well before time. It is also useful for sugarcane producers to monitor the cane for sale. Also, the sugar industry can benefit from sugarcane. Ethanol and refined sugar are some sugarcane products that are in great demand. The remote sensing data have spectral, temporal, and spatial resolutions. This is the benefit of using it for agriculture. The traditional structured data, such as images of cats and dogs, require human labeling so that the algorithm can learn to recognize these species based on their unique visual characteristics. Machine learning's branch known as deep learning employs multilayered neural network techniques. The input data is processed through the network layers, with each layer establishing a different set of features and patterns. If you wish to train a model in deep learning to recognize particular attributes such as buildings and roads by feeding it photographs of those features, the model will eventually be able to recognize them by processing the images through its many layers of neural connections. The CaneSat dataset is publicly available at [21] and it contains georeferenced tiff images of sugarcane crop and nonsugarcane. Additionally, some jpeg images may be employed with deep learning models for geographical analysis. CNN, VGG19, and MobileNetV2 are used on the CaneSat dataset and their evaluation metrics are compared.

2 Literature Survey

An innovative deep learning framework approach to the detection of disease is presented in this research of sugarcane plants by analyzing their leaves, stems, and colors. Inception v3, VGG 16, and VGG 19 three popular feature extractors are compared and these relevant models are used to train various classifiers [2]. The major strategy used in this study was to create a sugarcane classification model using RS time series and CaneSat dataset is used for the work [3]. It is observed that when new deep learning methods are applied consistently, they outperform classical machine learning across most tasks. The Long Short-Term Memory (LSTM) Recurrent Neural Networks did not regularly beat Random Forests (RFs) and recent deep learning techniques routinely outperformed conventional machine learning techniques in yield prediction [4]. The deep reinforcement algorithms were used for crop yield optimization which demonstrated a basic plant simulation model accessible through the OpenAI Gym interface and used a state-of-the-art RL algorithm to teach a robot how to maximize harvests [5]. The discussion of deep learning techniques applied to diverse agricultural issues and the current state, benefits, drawbacks, and future possibilities of deep learning in agriculture are explored in this work [1]. A deep learning framework used convolutional neural networks for autonomous palm tree counting. The six new convolutional neural networks' models named Faster R-CNN, YOLOv3, YOLOv4, and Efficient Net were employed to recognize palm trees and other types of trees [6]. The author explains how deep learning was used to analyze data from remote sensors, processing, analysis, and overcoming technical hurdles [7]. The neural network type employed for this study is the feed-forward back-propagation multi-layer perceptron (MLP) in remote sensing and also consideration is applied for fuzzy categorization and multi-source data [8]. The work is shown on two datasets SAT 4 and SAT 6 and classification accuracy is compared based on the models applied [9].

This study compares and evaluates the abilities of the Sentinel-2 (S2) satellite and the Dove nanosatellite constellation, or PlanetScope (PS) data, to locate and map Striga (Striga hermonthica) vegetation in intercropped maize fields [10]. The applications for object identification, picture classification, and automated object clustering are used to demonstrate how to use ImageNet [11]. In this work explored the technologies like Distributed ledger technology, AI, ML for purpose of data security and automation [12]. The studies demonstrate CNNs as deep learning techniques to be particularly useful at representing spatial patterns and thereby extracting a rich set of vegetation properties from remotely sensed images. This summary explains convolutional neural networks and why they work well for vegetative remote sensing [13]. The study was to make sugarcane field map using Sentinel imagery data [14]. The aim was achieved in two steps by detecting sugarcane fields across temporal optical and microwave data using Random Forest and SVM classifiers [15]. The government may plan for an ongoing food supply by using the recommended technology to identify crop types for small farms [16]. The technique classified the different types of sugarcane by employing a dense neural network with multiples of four neurons in each layer and numerous hidden layers, with the number of hidden layers being determined using the greedy layer-wise method [17]. The study to locate the maize fields using multispectral imaging data from Sentinel and Landsat imagery is discussed [18]. The importance of machine learning combined with remote sensing technology in identifying sugarcane crops is discussed in this study [19]. In order to forecast sugarcane leaf nitrogen, this study used Sentinel-2 spectral bands, Random Forest (RF), and support vector regression (SVR) models [20].

3 Methodology

CaneSat dataset [21] contains images of sugarcane crop and non-sugarcane which are created from Sentinel2 imagery data. The tiff as well as jpg in RGB colon bands is there in the dataset. The work is carried on jpg images of nonsugarcane and sugarcane crop. Out of 1627 images, 870 are sugarcane images and 757 are non-sugarcane images. The data is split into 70:30 ratios for training and testing; thus, there are 1138 train data and 489 test data as shown in Table 1.

Table 1 Dataset statistics [21]

Class	Training dataset	Testing dataset	Total
Sugarcane	620	250	870
Non-sugarcane	518	239	757
Total	1138	489	1627

Data Augmentation

It involves creating modified datasets using existing data, adjusting or generating new data points using deep learning. In this work, the data is augmented in the following forms:

(a) Normalization (rescale pixel values of 255): To setting pixel values to [0, 1] by normalizing the values of the image.
(b) Image rotation: The rotation range is kept 10 for the image.
(c) Shift in width: To randomly move the picture horizontally by up to 10% of the width.
(d) Shift in height: To randomly move the picture vertically by up to 10% of the height.
(e) Image zoom (zoom_range = 0.1): To randomly zoom the picture up to 10%.
(f) Horizontal flipping: To flip image in horizontal manner.
(g) Vertical flipping: To flip image in vertical manner.

Accuracy

Deep learning and machine learning models frequently employ accuracy as a statistic to assess the efficiency of a categorization operation. Mathematically, the accuracy can be calculated using the formula as shown in (1).

$$\text{Accuracy} = \frac{\text{TrueNegtive} + \text{TruePositive}}{\text{TruePositive} + \text{FalsePositive} + \text{TrueNegative} + \text{FalseNegative}} \tag{1}$$

Precision

A binary classification model's performance is measured by a statistic called precision. It is a measurement of the proportion of real positive cases (TP + FP) among all anticipated positive examples. In other words, it evaluates how well the model can accurately identify positive instances while reducing false positives. Mathematically, precision is defined as shown in (2).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall

Recall measures the proportion of true positives (TP) among positive and negative instances, demonstrating the model's capacity to recognize positive examples while reducing false negatives, and uses this information to assess the effectiveness of binary classification models as shown in (3).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

3.1 Convolutional Neural Network (CNN)

The convolutional neural network (CNN) model is trained using CaneSat dataset as shown in Fig. 1. As input, Sentinel-2 imagery is used, which is represented as feature vectors. To extract information from the input, each convolutional layer employs a 3 × 3 kernel and a certain number of filters (3, 6, and 9). Each convolutional layer's output feature maps have the same size as the input, maintaining the original information of the picture patches. To down sample the feature maps, a layer with a 2 × 2 filter size is implemented after the third convolutional layer. In order to introduce nonlinearity into the network, Rectified Linear Unit (ReLU) is used in the convolutional layers. To calculate the difference between predicted and actual outputs, the model employs category cross entropy as the loss function. After the fully connected layer, dropout regularization with a probability of 0.25 is applied to prevent over fitting. A softmax layer follows the fully connected layer, transforming the output of the network into a probability distribution with two classes.

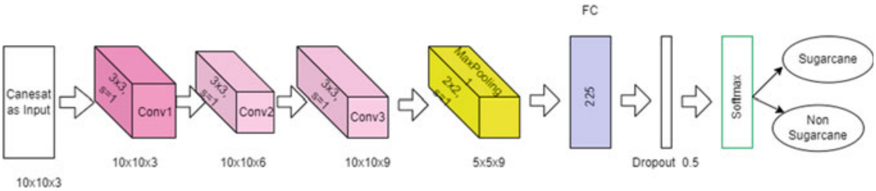


Fig. 1 CNN model architecture [3]

3.2 MobileNetV2

MobileNetV2 uses convolutional layer, 3 × 3 kernels, and two strides for spatial input constraints. The model uses an "Inverted Residual Block" to achieve a compromise between accuracy and effectiveness. This block consists of a depth-wise separable convolution, expansion layer, and projection layer. The middle flow consists of successive stacks of repeating inverted residual blocks, allowing the network to learn more expressive representations. The final layers sharpen the characteristics further, with a batch normalization step, ReLU activation function, and 1 × 1 convolutional layers. The final feature map is spatially compressed, pooled, and classified using a global average pooling layer. As a feature extractor, MobileNetV2 model is used, with the depth-wise and point-wise convolutions' weights frozen. The output of MobileNetV2's final convolutional layer is used as the input for other layers (such as fully connected layers) that are added for task-specific categorization or fine-tuning.

By employing MobileNetV2 as shown in Fig. 2 as a feature extractor, it is possible to take use of the high-level features that the model has learnt on a big dataset (like ImageNet) and use them in other image-related tasks with constrained computing resources. ML classifier used is SVM, whereas other classifiers can also be used for better accuracy.

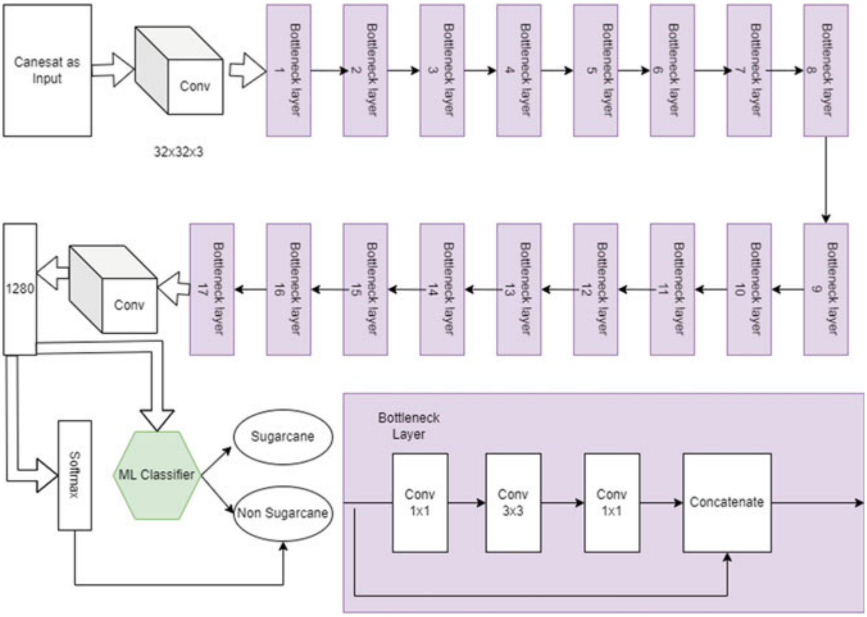


Fig. 2 MobileNetV2 as FE architecture [20]

3.3 Visual Geometry Group (VGG19)

VGG19 architecture was proposed by Oxford University for deep CNN model. It has convolutional, pooling, and completely connected layers and so its named as VGG19. The VGG19 architecture is described in great depth below.

The VGG19 network's input layer can take an image of any size as shown in Fig. 6. The standard input size is 224 by 224 pixels. As for the VGG19 network's convolutional layers, they are labeled "Conv1" through "Conv16" and total 16. These layers extract features by convoluting the input images with a series of trainable filters for feature extraction. The convolutional layers undergo nonlinear activation with ReLU function. The convolutional layers followed by max pooling layers, sampling feature map the spatial dimensions. The VGG19 network employs max pooling with a stride of 2 and a 2 × 2 filter. Three 4096-unit levels (labeled "FC1", "FC2", and "FC3") are fully connected in the VGG19 design. All of the neurons in one layer are linked to those in the next layer via these interconnecting layers. The final categorization is carried out by the fully connected layers, which also collect the higher-level features. VGG19 network's softmax layer converts output to probability distribution across classes. Each class label is given a probability that represents how likely it is that the input fits into that category. The VGG19 network is depicted in Fig. 3 with explanation of the layers in the network. The VGG19 framework is often praised for its clarity and consistency. The network may learn complex hierarchical features since the convolutional layer's model with 3 × 3 filters are employed in the network. VGG19 as fine extractor (FE) architecture is shown in Fig. 3.When utilizing VGG19 as a feature extractor, only the fully connected layers are updated and trained on the new dataset, leaving the pre-trained weights learnt on sizable datasets (like ImageNet) fixed. With this method, information gained from a big, diverse dataset may be used to a smaller, less training dataset.

4 Results

Figure 4 shows the confusion matrix, a 2 × 2 matrix that summarizes the actual and predicted class labels for a set of examples where value 1 is sugarcane and value 0 is non-sugarcane. For class = sugarcane, true positive (TP) is 210, whereas false positive (FP) is 68. False negative (FN) is 160 and true negative (TN) is 43.

Figures 5 and 6 show the accuracy graph and loss graph for 100 epochs. The training accuracy is 83% and validation accuracy is 77%.

Training_Accuracy: 0.8330404162406921

Val_Accuracy: 0.7730061411857605

Precision: 0.7580071174377224

Recall: 0.83203125.

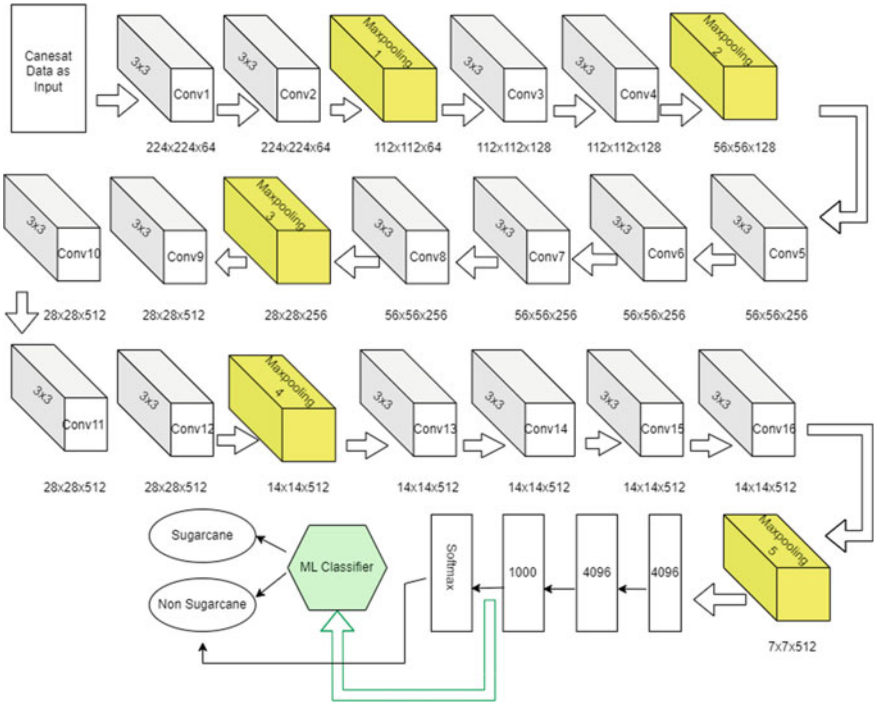
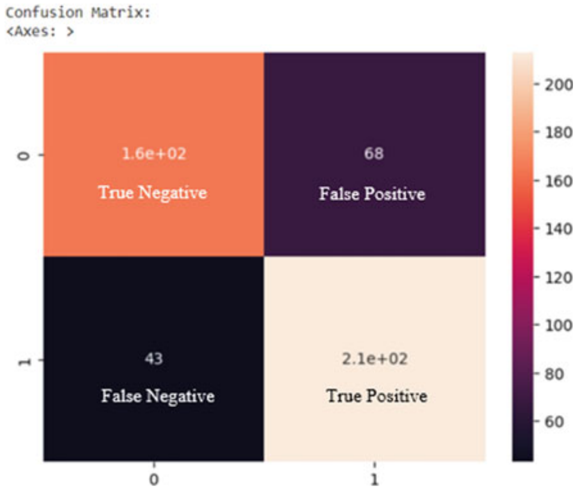


Fig. 3 VGG19 as FE architecture [22]

Fig. 4 CNN model confusion matrix


```
Confusion Matrix:
<Axes: >
```

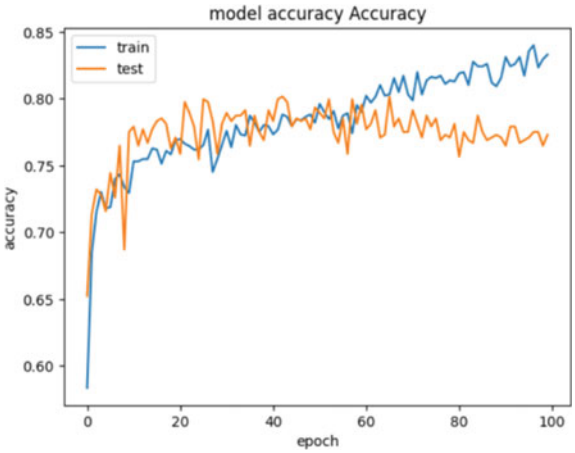


Fig. 5 CNN accuracy graph

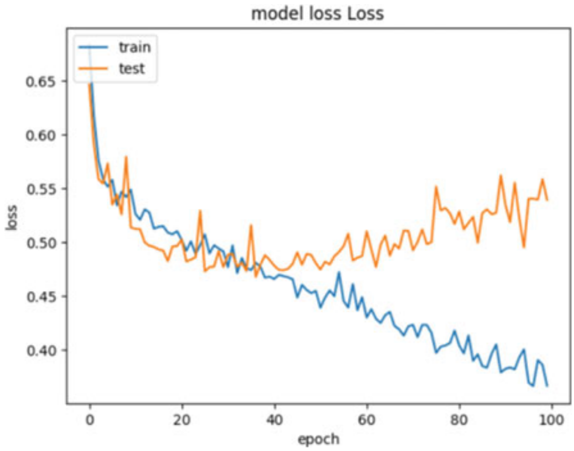


Fig. 6 CNN loss graph for model

Model Accuracy and Loss for MobileNetV2

Fine-tuned: The model accuracy graph values for training accuracy of 70% and loss accuracy of 56% are obtained as shown in Figs. 7 and 8, respectively**.** As shown in Fig. 9, the true positive values are 180 samples and true negative are 56, whereas false negative are 79 and false positive are 170 samples. In Fig. 10, true positive are 210 and true negative are 80, whereas false positive are 150 and false negative are 39. The accuracy and precision are better as compared to fine-tuned model.

Training accuracy: 0.7085514664649963

Validation accuracy: 0.7657840847969055

Training loss: 0.5642507672309875

Fig. 7 Accuracy graph

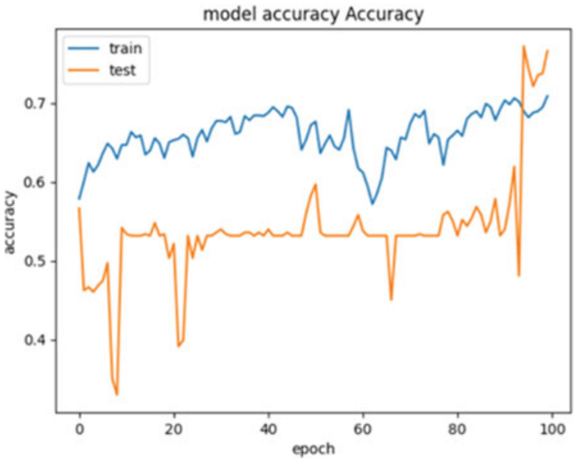
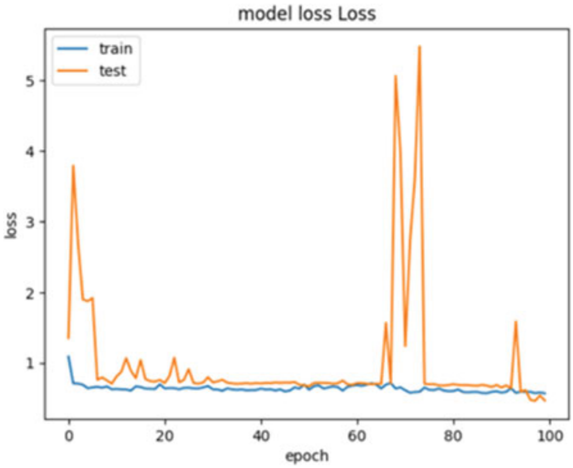


Fig. 8 Loss graph



Validation loss: 0.46231648325920105

Precision score: 0.4660692294848683

Recall score: 0.4847250509164969

MobileNetV2 as FE

Accuracy: 0.7225

Precision: 0.732797783933518

Recall: 0.8450492023715415

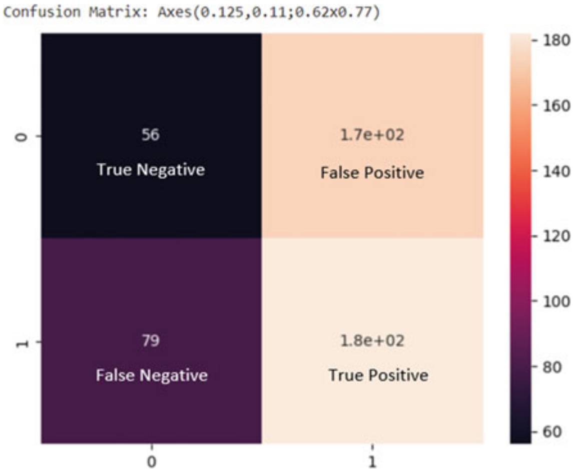


Fig. 9 MobileNetV2 fine-tuned CM

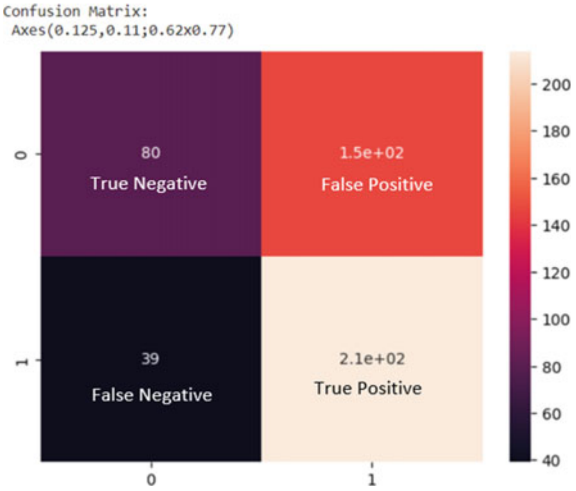


Fig. 10 MobileNetV2 FE confusion matrix

Model Accuracy and Loss for VGG19

VGG19 Fine-Tuned Model

The accuracy obtained for training is 75% for VGG19 and validation accuracy is 71% as shown in Fig. 11, and training loss is 47%, whereas validation loss is 50% as shown in Fig. 12.

Training accuracy VGG19: 0.7469459176063538

Validation accuracy VGG19: 0.7087576389312744

Training loss VGG19: 0.4711935818195343

Fig. 11 VGG19 accuracy graph

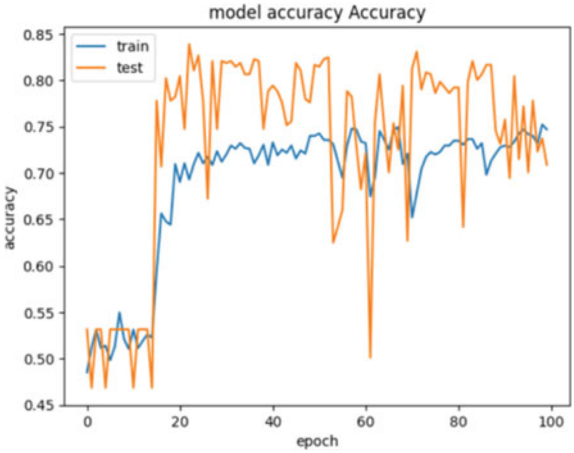
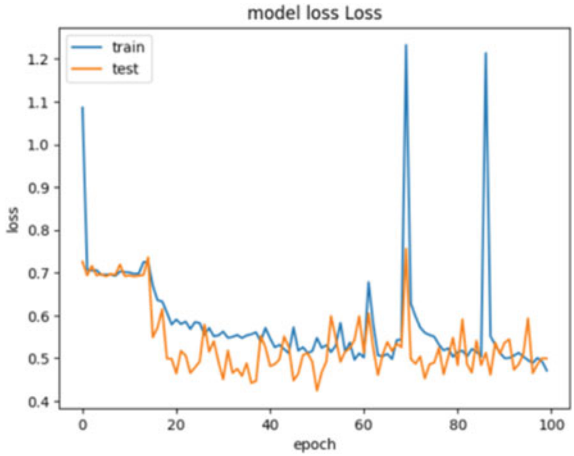


Fig. 12 VGG19 loss graph



Validation loss VGG19: 0.4995840787887573.

VGG19 Fine-Tuned Model Precision and Recall

Precision score: 0.518131123729308

Recall score: 0.5193482688391039.

Confusion Matrix

As shown in Fig. 13, the true positive samples are 150 and true negative are 110, whereas false positive are 120 and false negative are 110. As shown in Fig. 14, true positive are 220 and true negative are 140, whereas false positive are 77 and false negative are 16.

Fig. 13 VGG19 fine-tuned model CM

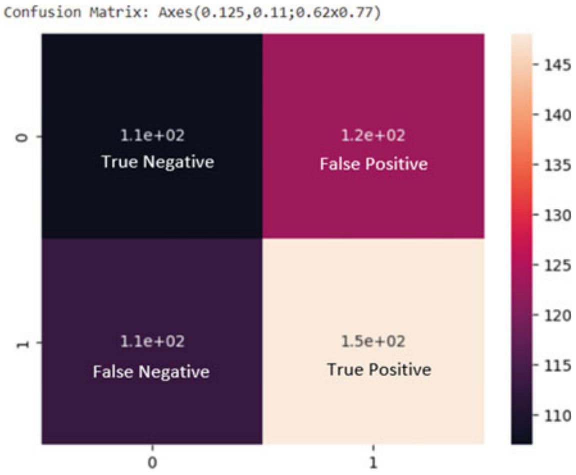
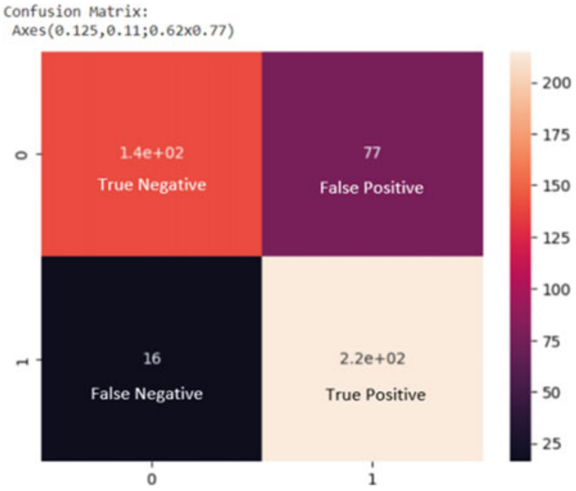


Fig. 14 VGG19 as FE confusion matrix



VGG19 as FE Model Performance Measure

Accuracy: 0.7924107142857143

Precision: 0.7363013698630136

Recall: 0.9307359307359307.

Table 2 compares the deep learning models that were used to categorize sugarcane and non-sugarcane. SVM and RF are the classifiers used for evaluation in terms of accuracy, precision, and recall. VGG19 evaluates deep learning models' accuracy, precision, and recall in Sentinel-2 dataset for sugarcane classification effectiveness.

Table 2 Deep learning model evaluation table

Models	Accuracy	Precision	Recall	Accuracy as FE		Precision as FE		Recall as FE	
				SVM	RF	SVM	RF	SVM	RF
CNN	83.30	75.80	83.20	–		–		–	
MobileNetV2	70.85	46.6%	48.47	72.25	71.95	73.27	72.61	84.58	79.37
VGG19	74.69	51.81	51.93	79.24	77.50	73.63	75.34	93.07	85.93

5 Conclusion

According to the findings, the CNN model with network layers achieves an accuracy of 83.40%. Despite the challenges posed by a limited sugarcane sample size, the average overall accuracy of VGG19 as feature extractor is 79.24% and MobileNetV2 is 72.25%. MobileNetV2 (70.85 and 72.25%) is marginally less competent than every other model. Overall VGG19 model as feature extractor (FE) gives better result than that of CNN and MobileNetV2 model as there is difference in its precision and recall when compared with other model's precision and recall. The future scope can be increased dataset of sugarcane or hyperspectral imagery dataset to increase the accuracy when used with deep learning models. Further, the classified sugarcane crop can be used for dense and sparse vegetation analysis or variety of sugarcane can be found from it.

References

1. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: a survey. Comput Electron Agric 147:70–90
2. Srivastava S, Kumar P, Mohd N, Singh A, Gill FS (2020) A novel deep learning framework approach for sugarcane disease detection. SN Comput Sci 1(1):1–7
3. Virnodkar SS, Pachghare VK, Patil VC, Jha SK (2022) CaneSat dataset to leverage convolutional neural networks for sugarcane classification from Sentinel-2. J King Saud Univ Comput Inf Sci 34(6):3343–3355
4. Victor B, He Z, Nibali A (2022)A systematic review of the use of deep learning in satellite imagery for agriculture. arXiv preprint arXiv:2210.01272
5. Ashcraft C, Karra K (2021) Machine learning aided crop yield optimization. arXiv preprint arXiv:2111.00963
6. Ammar A, Koubaa A, Benjdira B (2021) Deep-learning-based automated palm tree counting and geolocation in large farms from aerial geotagged images. Agronomy 11(8)
7. Zhang X, Zhou Y, Luo J (2022) Deep learning for processing and analysis of remote sensing big data: a technical review. Big Earth Data 6(4):527–560
8. Atkinson PM, Tatnall ARL (1997) Introduction neural networks in remote sensing. Int J Remote Sens 18(4):699–709
9. Basu S, Ganguly S, Mukhopadhyay S, DiBiano R, Karki M, Nemani R (2015) DeepSat: a learning framework for satellite imagery. In: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, pp 1–10

10. Mudereri BT (2019) A comparative analysis of PlanetScope and Sentinel-2 space-borne sensors in mapping Striga weed using Guided Regularised Random Forest classification ensemble. Int Arch Photogramm Remote Sens Spat Inf Sci 42:701–708
11. Deng J, Dong W, Socher R, Li J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. IEEE
12. Mitra A, Alakananda S, Vangipuram LT, Bapatla AK, Bathalapalli VKVV, Mohanty SP, Kougianos E, Ray C (2022) Everything you wanted to know about smart agriculture. arXiv preprint arXiv:2201.04754
13. Kattenborn T, Leitloff J, Schiefer F, Hinz S (2021) Review on convolutional neural networks (CNN) in vegetation remote sensing. ISPRS J Photogramm Remote Sens 173:24–49
14. Virnodkar S, Pachghare VK, Patil VC, Jha SK (2021) Performance evaluation of RF and SVM for sugarcane classification using sentinel-2 NDVI time-series. In: 9th international proceedings on proceedings, pp 163–174. Springer, Singapore
15. Nihar A, Patel NR, Pokhariyal S, Danodia A (2021) Sugarcane crop type discrimination and area mapping at field scale using Sentinel images and machine learning methods. J Indian Soc Rem Sens 1–9
16. Khan HR, Gillani Z, Jamal MH, Athar A, Chaudhry MT, Chao H, He Y, Chen M (2023) Early identification of crop type for smallholder farming systems using deep learning on time-series sentinel-2 imagery. Sensors 23(4):1779
17. Kai PM, Oliveira BM, da Costa RM (2022) Deep learning-based method for classification of sugarcane varieties. Agronomy 12(11)
18. Wijayanto AW. Triscowati DW, Marsuhandi AH (2020) Maize field area detection in East Java, Indonesia: an integrated multispectral remote sensing and machine learning approach. In: 2020 12th international conference on information technology and electrical engineering (ICITEE), pp 168–173. IEEE
19. Virnodkar SS, Pachghare VK, Patil VC, Jha SK (2020) Application of machine learning on remote sensing data for sugarcane crop classification: a review. In: ICT analysis and applications: proceedings of ICT4SD 2019, vol 2, pp 539–555
20. Soltanikazemi M, Minaei S, Shafizadeh-Moghadam H, Mahdavian A (2022) Field-scale estimation of sugarcane leaf nitrogen content using vegetation indices and spectral bands of Sentinel-2: application of random forest and support vector regression. Comput Electron Agric 200:107130
21. Virnodkar S, Pachghare V, Patil V, Jha SK (2022). CaneSat. https://doi.org/10.21227/vzbn-qj64
22. Saini R, Ghosh SK (2018) Exploring capabilities of Sentinel-2 for vegetation mapping using random forest. Int Arch Photogramm Remote Sens Spat Inf Sci 42:1499–1502

Detection and Analysis of Wormhole Attacks in the AODV Routing Protocol with IEEE 802.11p for the Internet of Vehicles

Tanuj Meshram and Mou Dasgupta

Abstract The Internet of Vehicles (IoV) is a network of vehicles that consists of vehicles, sensors, and technologies that allow communication between them. Its primary purpose is to enable vehicles to connect and share information via the Internet. The connectivity between all these entities is the most important. So, routing becomes the most crucial thing in IoV to establish connections. AODV is an essential protocol for efficiently connecting vehicles and other environments. In vehicular networks, the wormhole is a routing attack that severely threatens communication and data security. The objective of our research is to compare the performance analysis for metrics like Packet Delivery Ratio (PDR), End-to-End Delay (EED), and Throughput during wormhole attacks with those during normal operations. The study looks at how wormhole attacks affect the Ad hoc On-Demand Distance Vector (AODV) routing protocol. We have used NS 2.35 for simulation and getting results. We have also done statistical testing with one-way Analysis of Variance (ANOVA) to determine the significance of the difference. The study employs the IEEE 802.11p standard and aims to find evidence of the existence of wormhole attacks through statistical testing. The detection of wormhole attacks on IoV performance is explored through simulation results and statistical analysis.

Keywords IoV · Wormhole · AONVA · IEEE 802.11p and NS2.35

1 Introduction

The advent of Vehicular ad hoc Networks (VANETs) has brought about a significant transformation in the vehicle industry, leading to the emergence of the Internet of Vehicles (IoV). Because vehicles are interconnected and share data, the resulting VANET is a vital component of the IoV and ultimately improves road safety. The concept of IoV is based on the foundation of VANET, a branch of the Internet of

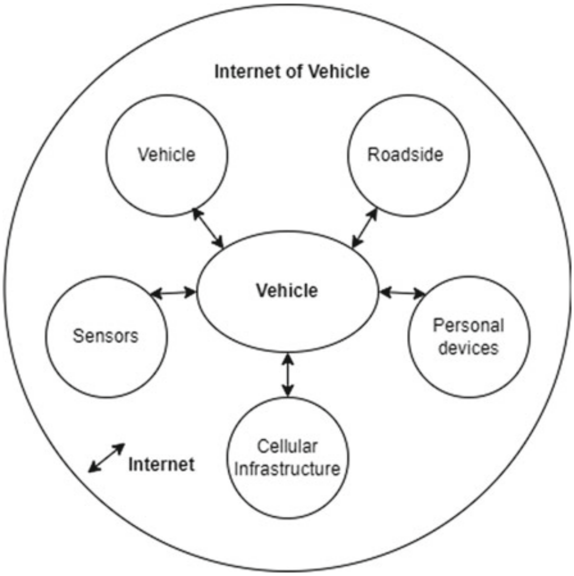
T. Meshram (✉) · M. Dasgupta
Department of Computer Applications, National Institute of Technology Raipur, Raipur, India
e-mail: tanujmmeshram@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_2

Things. It has become a crucial enabling technology to bring about autonomous and intelligent transportation scenarios in smart cities. IoV is made up of a plethora of vehicles, things, and networks, and it is highly controllable, operational, and credible. The fundamental IoV architecture is illustrated in Fig. 1 and described [1]. This architecture's essential and indispensable feature is the connectivity among all the entities. Hence, routing becomes the primary need for establishing a connection in IoV. AODV is essential for establishing efficient relationships between vehicles and other components [2]. AODV is a protocol for reactive routing. Every node has its routing table, including information about its connections. There are two phases of the AODV mechanism: the first is route discovery, and the second is route maintenance. This mechanism makes it simpler for every node to send a packet to its intended destination [3]. Despite this, VANETs have disadvantages, including unpredictable vehicle movement, changeable network topology, a limited communication range, and many routing vulnerabilities that pose significant routing threats [4].

Wormhole attacks are one of the most severe and evident threats to VANET security. They reduce the network's dependability and make it more difficult for the AODV routing protocol to perform other security-related tasks [5]. In a wormhole attack, a malicious vehicle node creates a high-speed virtual tunnel between two peer nodes and forwards packets through this tunnel to alter the network [6]. This attack can alter communication patterns, disrupt routing protocol activity, and manipulate potentially sensitive data [7]. This characteristic renders it well-suited for examining the wormhole attack within the AODV routing protocol, identifying its origins to enhance network security, and establishing trust among nodes to ease the exchange of data in the IoV.

Fig. 1 Fundamental IoV architecture



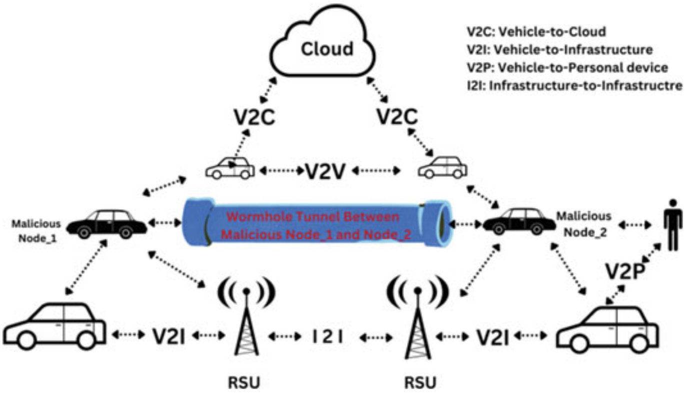

Cloud
V2C: Vehicle-to-Cloud
V2I: Vehicle-to-Infrastructure
V2P: Vehicle-to-Personal device
I2I: Infrastructure-to-Infrastructre
V2C
V2C
V2V
Malicious Node_1
Wormhole Tunnel Between Malicious Node_1 and Node_2
Malicious Node_2
V2P
V2I
I2I
V2I
RSU
RSU

Fig. 2 Wormhole attacks in IoV architecture

1.1 Wormhole Attack in IoV

The wormhole attack tends to be the most dangerous in IoV. When one malicious node intercepts data, it is forwarded to another malicious node somewhere in the network. A wormhole attack could result in unauthorized access, the disruption of routing systems, the launching of denial-of-service attacks, and other similar operations [8]. There is a connection between the two malicious nodes used in this attack. The first malicious node collects the data packets and transfers them to a second malicious node, which then distributes them within its immediate vicinity [9]. The wireless nature of the network makes it feasible for malicious nodes to establish a wormhole. The packets pass through a wormhole tunnel to the cooperating node at the opposite end of the wormhole.

In contrast to the other nodes, the wormhole grants the attacker command authority [10]. A malicious node launches a two-phase wormhole attack. The two malicious tunnels that end in the initial phase may pass routing traffic to attract routes. Wormhole nodes might use data in the second phase. Forwarding all packets through a wormhole tunnel may be helpful for other nodes, but it will deliberately attack nodes to disrupt routing protocols. They may disrupt data flow by intentionally dropping, altering packets, and often shutting off the wormhole link [11]. An example of a wormhole attack in IoV architecture is shown in Fig. 2.

1.2 Contributions

This paper presents the following contributions.

- Our research aims to detect the wormhole and establish evidence of its presence in the AODV routing protocol within the framework of the IoV, using the specialized IEEE 802.11p standard. For analysis, we have used the simulation NS 2.35 to evaluate the real-world performance of AODV employing SUMO mobility-generated traffic under attacks and compare it with its performance under normal conditions.
- We then ran an in-depth analysis of the performance factors, looking particularly at the PDR, EED, and Throughput, and conducted ANOVA tests to establish statistical significance. This study expands our understanding of IoV security vulnerabilities and highlights the crucial need to resolve these issues for a reliable and secure Internet of Vehicles.

1.3 Organization

The remaining sections of this work are structured as follows. In Sect. 2, we take a brief look at some of the research done against wormhole attacks. In Sect. 3, we examine the performance metric, concisely summarize the detection results against the wormhole, explore the possible consequences of these results, and analyze the one-way ANOVA and performance analysis of the ns2 simulation results. The concluding section offers a conclusion.

2 Related Works

This section examines the existing research carried out on the detection of wormhole attacks in the VANET. The taxonomy for identifying wormhole attacks is illustrated in Fig. 3. In this paper, Amish and Vaghela [9] presented research which thoroughly examines established methodologies for identifying wormhole attacks in Wireless Sensor Networks (WSNs) and introduces a novel strategy for their detection and mitigation. The utilization of the Ad hoc On-Demand Multi-path Distance Vector (AOMDV) routing protocol was employed by this technique, wherein the Round Trip Time (RTT) mechanism serves as a fundamental component. The suggested methodology's efficacy surpasses other methods documented in existing literature. All simulations were conducted using the NS2 simulator. In this paper, Qian et al. [12] proposed that the identification of attacks can be accomplished through the utilization of a direct and uncomplicated method that relies on statistical analysis of multi-path (SAM). In contrast to previous methodologies, such as packet leash, the proposed approach does not necessitate supplementary infrastructure such as

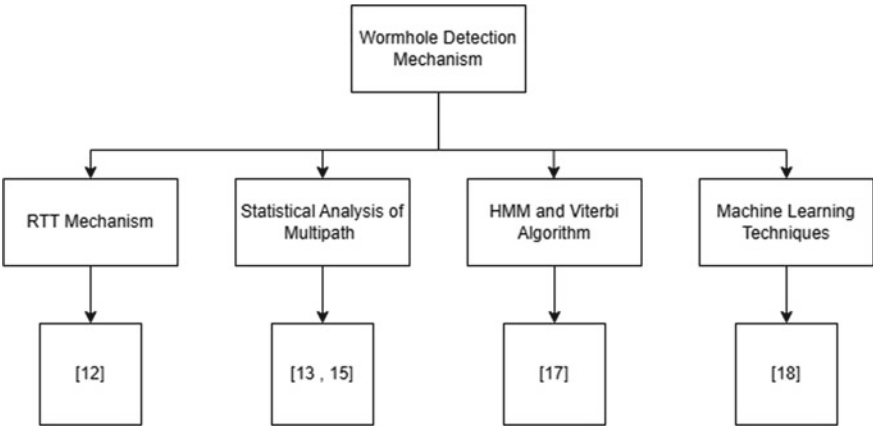

Wormhole Detection Mechanism
RTT Mechanism
Statistical Analysis of Multipath
HMM and Viterbi Algorithm
Machine Learning Techniques
[12]
[13 , 15]
[17]
[18]

Fig. 3 Taxonomy for detecting wormhole attacks

time synchronization or GPS. Furthermore, SAM has the potential to function as a component within local detection agents in an Intrusion Detection System (IDS) designed specifically for wireless ad hoc networks.

In this paper, Tiwari et al [8]. demonstrated how wormhole impacts AODV routing protocol and the influence of wormhole attacks on PDR, EED, and Throughput was analyzed. A proposed approach was introduced to identify and mitigate wormhole attacks in VANET by utilizing the multi-path idea. This method was designed to operate over a real map with dynamic vehicle densities. The purpose of their investigation is to make the VANET more secure. In this paper, Tian et al. [13] presented an innovative approach new statistical analysis to identify instances of wormhole attacks. The proposed approach involves utilizing a sensor to identify fake neighbors that are introduced by wormholes during the neighbor discovery process. Subsequently, a method based on k-means clustering is employed to identify and mitigate wormhole attacks. Using this method, it is possible to detect wormholes based solely on neighbor information without any additional requirements. The performance of this strategy was evaluated through a series of experiments, which yielded results indicating its ability to provide desirable outputs. In this paper, Khalil et al. [14] introduced a lightweight defense against wormhole attacks that exploit the ability to eavesdrop on nearby communications for multi-hop wireless networks with limited resources. With this technique, the wormhole can be located, and then the malicious nodes can be taken away. The simulation findings demonstrate that, across various conditions, every wormhole is quickly recognized and isolated. In this paper, Obado et al. [15] used Hidden Markov Model (HMM) and Viterbi algorithm which were utilized to detect wormhole assaults by determining the hidden state transitions with the highest estimated probabilities. The determination of the distance between a source node and a destination node is achieved by identifying the shortest path that exhibits the minimum number of hops. The Viterbi method utilizes a given observation sequence to determine the states that correspond to the shortest pathways, hence identifying viable wormhole channels based on their minimal overall cost. In this paper, Ali et al. [16] have used machine learning techniques to detect wormhole attacks in multi-hop VANET communication. The flow monitor-generated statistics were collected using the AODV routing protocol on the NS3 simulator, and the SUMO simulator generated the overall mobility traces to simulate the wormhole attack. The collected data is preprocessed, and then KNN and Random Forest algorithms are applied to this data to generate a model capable of learning wormhole attacks. With the proposed detection and prevention technique and ML-based approach, VANET can be immune to wormhole attacks.

3 Wormhole Attack Detection and Analysis

Our goal is to compare the result-based performance analysis during wormhole attacks to that during normal operation. Subsequently, we generate mobility using Sumo and look into the consequences of wormhole attacks on the AODV routing protocol. Table 1 displays the simulation parameter. One of the most widely used open-source simulation tools is NS 2.35, which we have successfully employed. In our research, we used the performance metrics of PDR, EED, and Throughput. We also used one-way ANOVA statistical testing to test the significance of the variation in performance metrics. Simulation findings and statistical analysis are employed to identify wormhole attacks on the performance of the IoV. The 50 nodes are depicted in Fig. 5, with Nodes 8, 14, 16, and 18 serving as sources and Nodes 3, 9, 10, and 11 serving as destinations.

Table 1 Simulation parameter

Parameter	Value
Max. floor size	2345 × 1423 m^2
Network/traffic simulator	NS-2.35/SUMO-0.32.0
Node density	20, 25, 30, 35, 40, 45, 50
Stop time	35–70 s
Transport protocol/type of traffic	UDP/constant bit rate
Propagation delay	Constant speed
Detection of attacks	Wormhole attack
MAC layer Mac/802_11	Mac/802_11Ext
Tunnel length	2 node
Packet size	512 B

3.1 Attack Model and Assumption

The final wormhole simulation is shown in Fig. 4, using the agent vehicle, i.e., the source node, performing the role of a wormhole node.

Let us assume there are no wormholes in the network. We assume that if a single node sends data in a specific direction, the packet has not been altered, dropped, or forwarded between the two good neighbors for any reason other than a wormhole attack. We assume the packet has been forwarded between the two good neighbors only because of a wormhole attack. Let us assume that the adversary pair of wormhole nodes is <16, 14> and <18, 8>. In this study, we demonstrate the efficacy of our proposed methodology in effectively identifying and detecting the occurrence of wormhole assaults. As a consequence of the wormhole attack, when node <16, 18>

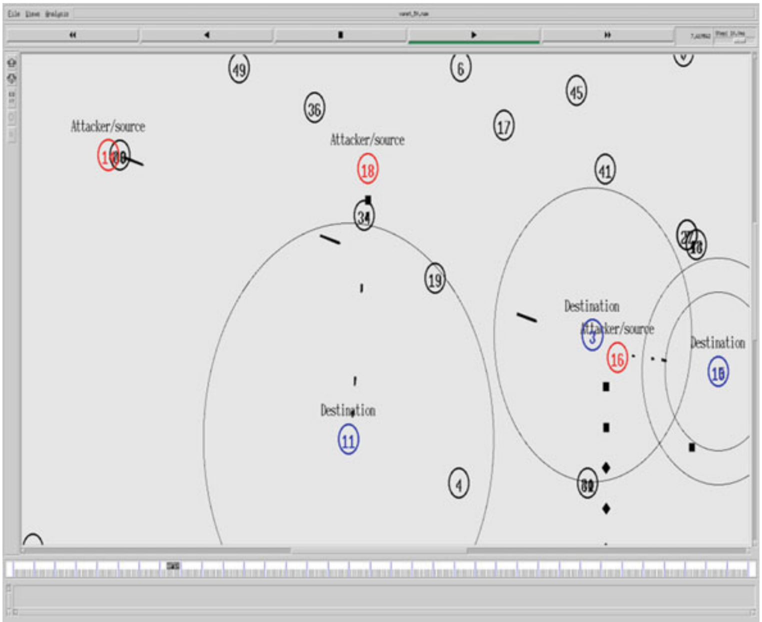
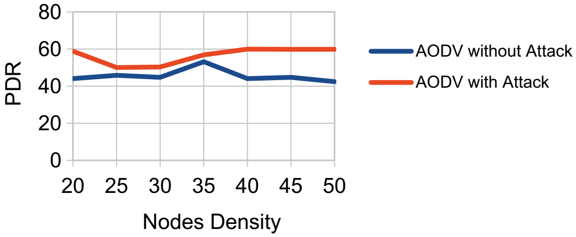


Fig. 4 Simulation of 50 nodes under wormhole attack

Fig. 5 PDR_comparison



transmits a data packet to node <14, 8>, PDR and Throughput will increase, and EED will decrease. Without the wormhole attack, PDR and Throughput would decrease, and EED would increase.

3.2 Metrics for Performance

A. Packet Delivery Ratio (PDR): It is the metric used to measure the possibility that a given destination node will receive all of the packets sent to it.

$$\text{PDR} = \frac{\sum_{a=1}^{a=n} \text{packet(received)}}{\sum_{a=1}^{a=k} \text{packet(sent)}}. \tag{1}$$

In this context, n represents the total number of network nodes [17].

B. End-to-End Delay (EED): It is a measure of how long it takes for a packet to go from its source node to its destination node over a given network. Due to the fact that only one synchronous path exists between the sender and the target, this measurement is commonly referred as RTT in networking.

$$\text{EED} = \frac{\sum_{a=1}^{a=n} (\text{Data}_A)}{k}. \tag{2}$$

In this context, Data_A represents the Ath packet sent from the sending node to the receiving node [17].

C. Throughput: It provides information regarding the total number of successful packets obtained when reaching the destination.

$$\text{Throughput} = \sum_{p=1}^{p=k} \text{packet(Destination)}. \tag{3}$$

In this context, packet refers to the successful Ath packet that was received by the destination node [17].

3.3 Performance Analysis

IEEE 802.11p is the standard that has been established to create a VANET simulation. IEEE has proposed a dedicated short-range communication (DSRC) system that runs on the 5.9 GHz band and employs the 802.11p access protocol mechanism [18]. Comparison of the PDR in percentage (%) between AODV without an attack and AODV with a wormhole attack is shown in Fig. 5. Here is a summary of the findings:

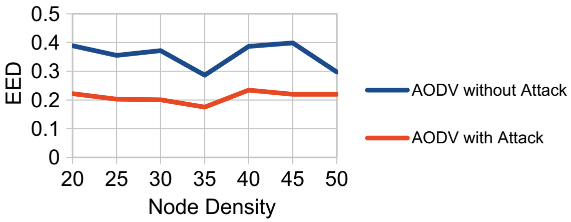


Fig. 6 EED_comparison

The PDR is higher in the presence of a wormhole attack compared to the absence of an attack. For node distances between 20 and 50, the PDR without an attack varies from 42.42 to 53.14%. When a wormhole is used, the PDR rises from 50.02 to 59.93% at the same node density. Increases in PDR show that the wormhole attack improves the efficiency with which the AODV routing system delivers packets.

A comparison of the EED in milliseconds (ms) between AODV without an attack and AODV with a wormhole attack is shown in Fig. 6. Here is a summary of the findings: The average delay for networks with 20–50 nodes is between 0.286 and 0.398 ms when there is no attack. When a wormhole attack happens, the average delay drops to 0.175 and 0.234 ms. The reduction in delay suggests that the wormhole attack allows for faster data transmission between distant nodes by establishing a shortcut in the network. In general, a wormhole attack results in a notable decrease in EED compared to the standard functioning of the Ad hoc On-AODV routing protocol.

Comparison of the Throughput in Kbps between AODV without an attack and AODV with a wormhole attack is shown in Fig. 7. Here is a summary of the findings: Figure indicates that AODV with an attack scenario outperforms AODV without an attack scenario regarding Throughput for all nodes. This could mean that the attack is optimizing the performance of routing protocol, resulting in more efficient data transmission and higher Throughput.

In summary, the presence of a wormhole attack within a network results in a notable reduction in the EED and a considerable enhancement in both the Packet Delivery Ratio (PDR) and Throughput. Further analysis and investigation are required to fully understand the impact and consequences of a wormhole attack on the AODV routing protocol and the overall performance of the Internet of Vehicles. As a result, the one-way ANOVA test will be used.

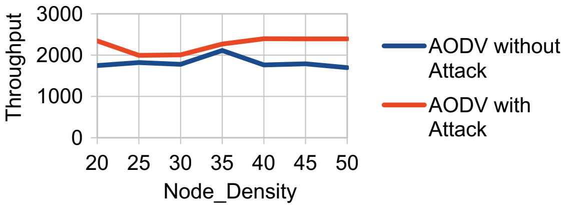


Fig. 7 Throughput_comparison

3.4 One-Way ANOVA Test for Detecting Wormhole

To detect the presence of wormhole attacks, we used statistical measures to analyze the variance within the two groups (AODV with an attack and AODV without an attack) and to identify the variance between the two groups in terms of varied node density (i.e., Nodes 10, 20, 30, 40, and 50). The AONVA test, used for evaluating variance analysis, was successfully applied to our research, which employs two scenarios. Thus, the degree of freedom is 1, so we conducted the ANOVA test using (= 0.05). The following hypothesis is being researched to detect the wormhole attack.

- In this case, null hypothesis (H_0) typically states that there is no difference in PDR/EED/Throughput scores between the two levels of the independent variable (i.e., two groups: AODV with an attack and AODV without an attack).
- Alternative hypothesis (H_1) would propose that there is a significant difference in PDR/EED/Throughput scores due to the presence of a wormhole attack.

To assess the null hypothesis, we examine the F-value and the associated SL. The F-value is calculated by comparing the between-group variability to the within-group variability. In Table 2 SL (< 0.001) indicates the probability of observing the obtained F-value by chance alone if the null hypothesis was true. In this study, we examined whether wormhole attacks exist and how they affect the AODV routing protocol in the context of IoV. Therefore, we reject the null hypothesis and conclude that there is a significant difference in PDR scores between the two levels of the independent variable.

In Table 3, the SL is less than 0.001, meaning that the probability is extremely low. Therefore, we reject the null hypothesis and conclude that there is a significant difference in EED scores between the two levels of the independent variable.

In Table 4, SL is less than 0.001, meaning that the probability is extremely low. Therefore, we reject the null hypothesis and conclude that there is a significant difference in Throughput scores between the two levels of the independent variable. In summary, based on the ANOVA results, we have evidence to support the alternative hypothesis that there is a significant difference in PDR, EED, and Throughput between the two levels of the independent variable being studied.

Table 2 PDR ANOVA test for two groups

PDR					
	Sum of squares (SoS)	Degrees of freedom (DoF)	Mean square (MS)	F-value	Significance level (SL)
Between groups	419.487	1	419.487	26.023	< 0.001
Within groups	193.436	12	16.120		
Total	612.924	13			

Table 3 EED ANOVA test for two groups

EED					
	Sum of squares (SoS)	Degrees of freedom (DoF)	Mean square (MS)	F-value	Significance level (SL)
Between groups	0.073	1	0.073	59.658	< 0.001
Within groups	0.015	12	0.001		
Total	0.088	13			

Table 4 Throughput ANOVA test for two groups

Throughput					
	Sum of squares (SoS)	Degrees of freedom (DoF)	Mean square (MS)	F-value	Significance level (SL)
Between groups	683,469.098	1	683,469.098	26.563	< 0.001
Within groups	308,757.069	12	25,729.756		
Total	992,226.167	13			

In conclusion, the research findings provide evidence of the impact of a wormhole attack on performance metrics. The statistical analysis and the rejection of the null hypothesis indicate a significant difference caused by the wormhole attack, highlighting the importance of addressing and mitigating this security threat in the Internet of Vehicles. If an attacker was to maintain a wormhole link and never drop any packets, the wormhole would serve as an advantage for the network by speeding up packet delivery. However, the attacker can deliberately slow the network down by dropping packets [11]. In addition, the attacker can cause a denial-of-service (DoS) attack by simply turning the wormhole link on and off, which causes route oscillation throughout the network [19].

4 Conclusion

In this study, we examined whether wormhole attacks exist and how they affect the AODV routing protocol in the context of the IoV. By simulation and statistical analysis, wormhole attack has been examined, significantly affecting the performance of PDR, EED, and Throughput. Implementing the AODV routing protocol in VANET is significantly impacted by wormhole attacks, as the research shows. Analyzing the simulation results and applying statistical testing, we observed notable differences in PDR, EED, and Throughput scores between scenarios with and without wormhole attacks. The rejection of the null hypothesis and the support for the alternative view indicate the presence of a substantial difference caused by the wormhole attack. Further research and analysis are necessary to fully understand the consequences of wormhole attacks on IoV and develop effective countermeasures to mitigate their impact. Exploring other performance metrics and evaluating the effectiveness of different security mechanisms against wormhole attacks would be valuable areas for future research.

References

1. Hatim SM, Elias SJ, Ali RM, Jasmis J, Aziz AA, Mansor S (2020) Blockchain based Internet of Vehicles (BIoV): an approach towards smart cities development. In: 2020 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE), Jaipur, India, 2020, p 1. https://doi.org/10.1109/ICRAIE51050.2020.9358355
2. Marinov T (2022) Comparative analysis of AODV, DSDV and DSR routing protocols in VANET. In: 2022 57th international scientific conference on information, communication and energy systems and technologies (ICEST), Ohrid, North Macedonia, pp 1–4. https://doi.org/10.1109/ICEST55168.2022.9828684
3. Kushwaha US, Gupta PK (2014) AOMDV routing algorithm for Wireless Mesh Networks with local repair (AOMDV-LR). In: 2014 international conference on communication and signal processing, Melmaruvathur, India, pp 818–822. https://doi.org/10.1109/ICCSP.2014.6949957
4. Liu X, Fang Z, Shi L (2007) Securing vehicular ad hoc networks. In: 2007 2nd international conference on pervasive computing and applications, Birmingham, UK, pp 424–429. https://doi.org/10.1109/ICPCA.2007.4365481
5. Ali S, Nand P (2016) Comparative performance analysis of AODV and DSR routing protocols under wormhole attack in mobile ad hoc network on different node's speeds. In: 2016 international conference on computing, communication and automation (ICCCA), Greater Noida, India, pp 641–644. https://doi.org/10.1109/CCAA.2016.7813800
6. Al-Sultan S, Al-Doori MM, Al-Bayatti AH, Zedan H (2014) A comprehensive survey on vehicular Ad Hoc network. J Netw Comput Appl 37:380–392. https://doi.org/10.1016/j.jnca.2013.02.036
7. Al-Karaki JN, Kamal AE (2004) Routing techniques in wireless sensor networks: a survey. IEEE Wirel Commun 11(6):6–28. https://doi.org/10.1109/MWC.2004.1368893
8. Ali S, Nand P, Tiwari S (2020) Impact of wormhole attack on AODV routing protocol in vehicular ad-hoc network over real map with detection and prevention approach. Int J Veh Inf Commun Syst 5(3):354. https://doi.org/10.1504/ijvics.2020.110997
9. Amish P, Vaghela VB (2016) Detection and prevention of wormhole attack in wireless sensor network using AOMDV protocol. Procedia Comput Sci 79:700–707. https://doi.org/10.1016/j.procs.2016.03.092
10. Ali S, Nand P, Tiwari S (2017) Secure message broadcasting in VANET over Wormhole attack by using cryptographic technique. In: 2017 international conference on computing, communication and automation (ICCCA), Greater Noida, India, pp 520–523. https://doi.org/10.1109/CCAA.2017.8229856
11. Azer M, El-Kassas S, El-Soudani M (2009) A full image of the wormhole attacks-towards introducing complex wormhole attacks in wireless ad hoc networks. arXiv preprint arXiv:0906.1245
12. Qian L, Song N, Li X (2007) Detection of wormhole attacks in multi-path routed wireless ad hoc networks: A statistical analysis approach. J Netw Comput Appl 30(1):308–330. https://doi.org/10.1016/j.jnca.2005.07.003

13. Tian B, Li Q, Yang Y, Dong L, Yang X (2012) A ranging based scheme for detecting the wormhole attack in wireless sensor networks. J Chin Univ Post Telecommun 19:6–10. https://doi.org/10.1016/s1005-8885(11)60478-0
14. Khalil I, Bagchi S, Shroff NB (2007) LiteWorp: detection and isolation of the wormhole attack in static multi hop wireless networks. Comput Netw 51(13):3750–3772. https://doi.org/10.1016/j.comnet.2007.04.001
15. Obado V, Djouani K, Hamam Y (2012) Hidden Markov model for shortest paths testing to detect a wormhole attack in a localized wireless sensor network. Procedia Comput Sci 10:1010–1017. https://doi.org/10.1016/j.procs.2012.06.140
16. Ali S, N and P, Tiwari S (2022) Detection of wormhole attack in vehicular ad-hoc network over real map using machine learning approach with preventive scheme. In: DOAJ (DOAJ: Directory of Open Access Journals)
17. Dhanaraj RK, Islam SH, Rajasekar V (2022) A cryptographic paradigm to detect and mitigate blackhole attack in VANET environments. Wirel Netw 28(7):3127–3142. https://doi.org/10.1007/s11276-022-03017-6
18. Shringar Raw R, Kumar M, Singh N (2013) Security challenges, issues and their solutions for vanet. Int J Netw Secur Appl 5(5):95–105. https://doi.org/10.5121/ijnsa.2013.5508
19. Poovendran R, Lazos L (2006) A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. Wirel Netw 13(1):27–59. https://doi.org/10.1007/s11276-006-3723-x

A Systematic Review of NLP Applications in Clinical Healthcare: Advancement and Challenges

Rachit Garg and Anshul Gupta

Abstract This systematic literature review examines the advancements and challenges of natural language processing applications in clinical healthcare. Authors provide an overview of NLP applications, including clinical text classification, named entity recognition, information extraction, clinical dialogue systems, and clinical decision support. These applications have improved clinical documentation, patient care, and research outcomes. Authors critically evaluate challenges such as data privacy, lack of standardized datasets, and domain-specific language models. Ethical considerations, interoperability, and potential biases in NLP algorithms are also discussed. This review highlights the current state of NLP in clinical healthcare, identifies areas for improvement, and suggests future research directions. By synthesizing existing literature, this paper contributes to a deeper understanding of NLP's potential in transforming clinical practice.

Keywords NLP · Healthcare · Clinical healthcare · Clinical decision support · Systematic literature

1 Introduction

Natural language processing (NLP) technology has revolutionized logistics [1–3], finance, legal, and business, including healthcare. NLP, a branch of AI, studies human–computer interaction to help robots understand, produce, and interpret human language. NLP has become a powerful tool for data-driven decision-making, improved patient outcomes, and better healthcare delivery. Natural language processing (NLP) use has increased dramatically in the healthcare industry recently.

R. Garg (✉)
Jain University, Bangalore, India
e-mail: rachit.garg.nitttr@gmail.com

A. Gupta
SP Jain School of Global Management, Dubai, UAE
e-mail: anshul.gupta@spjain.org

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_3

The promise benefits and downsides of this innovation have been the subject of several studies. Despite the abundance of research in this field, there is a need for a thorough synthesis and assessment of the existing literature to give gist of a thorough overview of the current trailblazing applications of NLP in the healthcare industry.

This study objective is to gain a thorough understanding of the present position of NLP implementations in the healthcare industry through a methodical examination and integration of the existing literature. The present analysis holds the promise of augmenting the extant corpus of scholarly literature, in addition to functioning as a beneficial tool for scholars, policymakers, and healthcare professionals who aim to enhance the utilization of NLP in healthcare settings. Our goal with this systematic review is to fill any existing research gaps, highlight the problems and limits of present NLP applications, and recommend prospective future research and development routes. The fundamental point of this study is to advocate for the utilization of evidence-based decision-making in order to improve patient outcomes, therefore furthering the utilization of linguistic processing in healthcare sector.

The following are the research objectives of the paper:

1. To furnish a comprehensive gist of the utilization of NLP in clinical documentation, highlighting its role in automating clinical coding, improving data extraction, and enhancing the coherence of healthcare information management.
2. To explore the inherent of NLP in patient engagement and communication, investigating how NLP techniques can enable conversational agents and chatbots to effectively interact with patients, provide personalized health information, and support remote monitoring.
3. To examine the utilization of NLP in clinical research and evidence-based medicine, assessing its contribution to systematic reviews, efficient updating of literature, eligibility criteria search for clinical trials, and adverse event detection.
4. To evaluate the shortcomings and opportunities in connection with the implementation of NLP in healthcare, considering ethical considerations, data privacy, interoperability, and scalability.
5. To provide insights into the future prospects of NLP in healthcare, discussing emerging trends, technological advancements, and potential areas of exploration for further research and development.

This review paper's goals are to improve the knowledge base of healthcare practitioners, academics, and policymakers about NLP, encourage its widespread use, and make it easier for them to make well-informed decisions.

2 Applications of NLP in Healthcare: An Overview

Natural language processing is a game-changing innovation that has found several uses in the healthcare industry and beyond. Clinical documentation benefits greatly from the application of NLP since it allows for the exploration of important knowledge from patient records. Furthermore, NLP enables conversational agents and chatbots to interpret and reply to patient requests, deliver individualized health information, and provide remote monitoring, which all contribute to improved patient engagement and communication. When it comes to clinical research and evidence-based medicine, NLP is crucial because of its ability to facilitate the systematic evaluation of literature, the quick updating of systematic reviews, the search for competence in medical trials, and the detection of adverse events. Collectively, these use cases demonstrate the enormous potential of natural language processing to revolutionize healthcare delivery, improve patient care, and further medical research and knowledge.

2.1 NLP in Clinical Documentation

The automation of clinical coding, the mining of knowledge from unformatted data, the improvement of clinical decision support systems (CDSS), and the facilitation of clinical data mining and analysis are all areas in which natural language processing hold a vital presence in modernizing clinical documentation. The use of NLP in clinical documentation has the potential to enhance productivity, precision, and health outcomes in the medical field. As an added bonus, natural language processing makes it possible to mine and analyze massive clinical datasets, which aids in areas like comparative effectiveness research, disease surveillance, adverse event identification, and prediction modeling. Table 1 shows the application of NLP in clinical documentation.

Table 1 Application of NLP in clinical documentation

Application	Description	References
Automated clinical coding and classification	Automates clinical coding tasks by extracting relevant clinical concepts from unstructured clinical text	Kaur et al. [4], Jiang et al. [5]
Extraction of clinical information	Converts unstructured data into structured formats, aiding clinical decision-making and research	Meystre et al. [6], Rajkomar et al. [7]
Clinical decision support systems using NLP	Enhances CDSS with real-time alerting, risk prediction, treatment recommendations, and drug interaction detection	Demner et al. [8], Hao et al. [9]
NLP-based clinical data mining and analysis	Mines large-scale clinical datasets for insights, patterns, and associations in EHRs and other repositories	Weng et al. [10], Harpaz et al. [11]

Table 2 Application of NLP in patient engagement and communication

Application	Description	References
Self-management	Empowering patients to actively participate in their healthcare by collecting and sharing their health data, leading to improved self-management and decision-making	Demiris et al. [12]
Clinical NLP	Advancements in NLP for semantic analysis in patient communication	Velupillai et al. [13]
Automated conversational agents in healthcare	Shortcoming and possibilities of using conversational agents in healthcare	Milne et al. [14]
Conversational agents in health service	Use of virtual agents for patient engagement in health services	Laranjo et al. [15]
Diabetes-specific health literacy	Measures the degree of health education specific to diabetes in elderly people with pre-diabetes residing in rural China	Hu et al. [16]
Learning patient-specific information in healthcare	NLP techniques for extracting patient-specific information from multiple sources	Zeng-Treitler et al. [17]

2.2 NLP for Patient Engagement and Communication

NLP has exciting potential to improve healthcare communication and patient engagement. Utilizing natural language processing methods, healthcare professionals can track and manage their patients' health, use chatbots and virtual assistants powered by NLP to offer individualized care, and increase their patients' knowledge and understanding of their conditions. Patient results, patient-centered treatment, and the patient experience could all benefit from these NLP applications. Table 2 shows the application of NLP in patient engagement and communication.

2.3 NLP for Clinical Research and Evidence-Based Medicine

The fields of clinical research and evidence-based medicine can both benefit significantly from the applications that NLP has to offer. NLP has the ability to automate procedures such as literature review and evidence synthesis, as well as improve clinical trial recruitment and eligibility screening, facilitate automated adverse event detection, and increase pharmacovigilance and drug safety monitoring activities. These applications of natural language processing help to clinical research techniques that are more evidence-based and efficient. Table 3 shows NLP in clinical research and medicine.

Table 3 NLP in clinical research and medicine

Application	Description	References
Efficient updating of systematic reviews in genetics	Modernizing systematic review pipelines through data mining	Wallace et al. [18]
Topic extraction and categorization	Extracts key topics and categorizes feedback based on common themes	Khanbhai et al. [19]
OHDSI for observational research	Opportunities for observational researchers through OHDSI	Hripcsak et al. [20]
NLP for competent criteria search in medical trials	Using NLP to enable eligibility criteria search in oncology trials	Zhang et al. [21]
Ontology-based representation of adverse events	Ontology-based representation of adverse events in pharmacovigilance	Yu et al. [22]
NLP methods for identifying adverse drug reactions	Literature review and comparison of NLP methods for ADR detection	Zhang et al. [23]
NLP in pharmacovigilance	Review of NLP applications in pharmacovigilance	Luo et al. [24]
Adverse drug reaction investigation from EHR	Utilizing hierarchical attention networks for ADR detection	Chu et al. [25]

3 Search Criteria, Inclusion/Exclusion Criteria, and Research Questions

For the purpose of pertaining that relevant researches are considered and that a synthesis of the findings is performed, this review will adhere to a methodology that is both organized and exhaustive. The authors have carried out a systematic literature search across multiple electronic databases, including PubMed, Scopus, ACM, and IEEE Xplore, using relevant keywords related to NLP, healthcare, and related terminologies. Figure 1 depicts the criteria for research papers inclusion and exclusion in filtering it.

To assure the worth and significance of the included papers, authors have applied specific criteria for considering and excluding. Only papers that focus on the application of NLP in healthcare, written in English, and peer-reviewed will be considered for inclusion. Authors have included various study designs, including laboratory research, large-scale observational studies, case reports, and systematic reviews. Research that primarily focus on non-healthcare applications of NLP or do not provide sufficient details on the NLP techniques employed will be excluded. This review ought to answer given questions of research (RQ).

RQ1: What are the present trailblazing NLP techniques and applications in the healthcare domain?

RQ2: What are the benefits and challenges of integrating NLP into clinical decision support systems?

RQ3: What are the key considerations regarding privacy, security, and regulatory compliance when implementing NLP in healthcare settings?

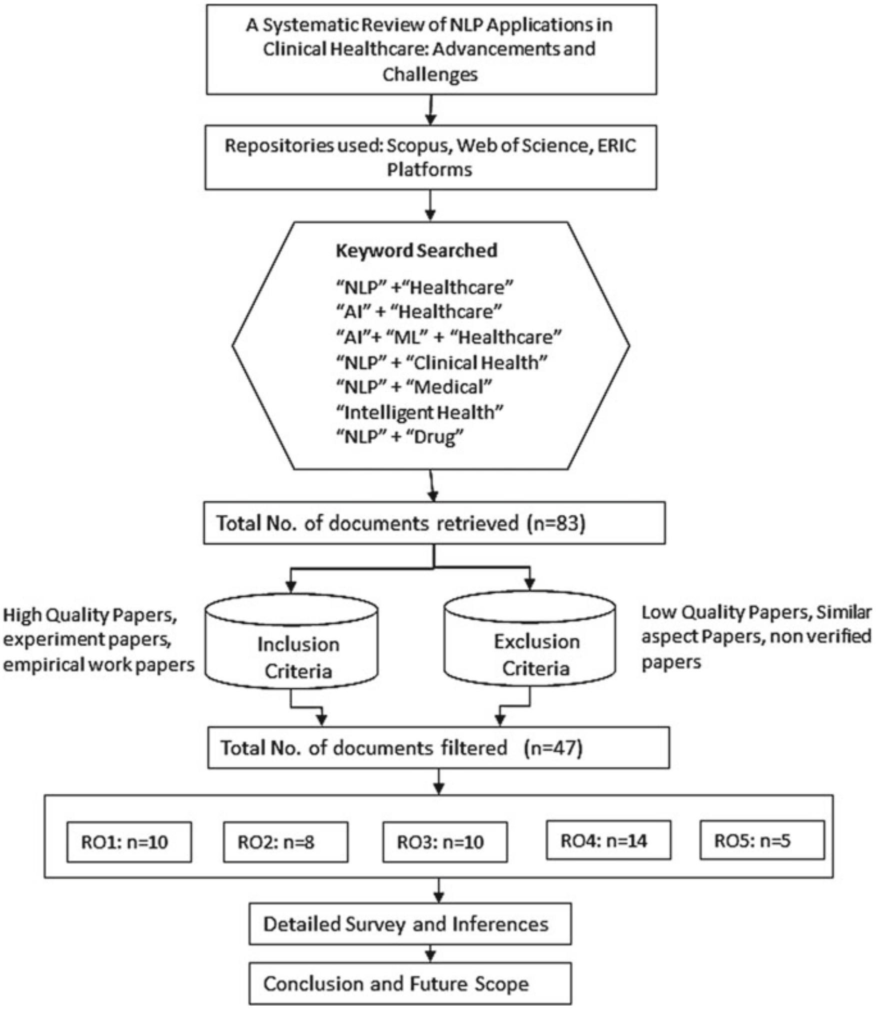

A Systematic Review of NLP Applications in Clinical Healthcare: Advancements and Challenges
Repositories used: Scopus, Web of Science, ERIC Platforms
Keyword Searched
"NLP" +"Healthcare"
"AI" + "Healthcare"
"AI"+ "ML" + "Healthcare"
"NLP" + "Clinical Health"
"NLP" + "Medical"
"Intelligent Health"
"NLP" + "Drug"
Total No. of documents retrieved (n=83)
High Quality Papers, experiment papers, empirical work papers
Inclusion Criteria
Exclusion Criteria
Low Quality Papers, Similar aspect Papers, non verified papers
Total No. of documents filtered (n=47)
RO1: n=10
RO2: n=8
RO3: n=10
RO4: n=14
RO5: n=5
Detailed Survey and Inferences
Conclusion and Future Scope

Fig. 1 Inclusion and exclusion criteria

4 NLP in Healthcare: Current Scenario

In this section, authors addressed the research questions to facilitate an extensive understanding of the applications of NLP in healthcare. By exploring each of the research questions, author aim to delve deeper into the specific ways in which NLP is transforming healthcare.

4.1 NLP Techniques and Applications in Healthcare (RQ1)

Using natural language processing (NLP) approaches, Caccamisi et al. [26] performed a thorough research on the topic of determining smoking status from electronic health records (EHRs). This paper has shown how linguistic processing may be used to automatically find smoking-related information from unstructured clinical writing, which can then be used for population-level analysis and clinical decision support.

The utilization of machine learning in medicine, including processing natural language, was recently reviewed extensively by Rajkomar et al. [27]. They stressed the importance of natural language processing in identifying cohorts, detecting adverse events, and predicting outcomes from clinical narratives and electronic health records.

Researchers and clinicians can both benefit from mining EHRs, as was stated by Jensen et al. [28]. They highlighted the significance of processing linguistic data in converting unstructured EHR data into structured information, which in turn aids data analysis and supports clinical decision-making. The potential of clinical NLP to enhance healthcare service and research was emphasized in a call for a comprehensive evaluation by Kreimeyer et al. [29]. Named entity recognition, relationship extraction, and information retrieval were among the many NLP techniques they covered. The effectiveness of automated de-identification of clinical note and their effect on information mining were assessed by Deleger et al. [30]. Their research proved the viability and efficacy of NLP-based de-identification methods, which are essential for maintaining patient privacy while allowing for the subsidiary use of medical data. A thorough analysis of deep learning in healthcare was presented by Miotto et al. [31]. They talked on the potential of deep learning models for NLP tasks in healthcare, such as classification of clinical document, recognizing named entity, and relation extraction. These models include Recurrent Neural Networks and CNN.

To autonomously identify clinically meaningful cancer characteristics from the reports of radiology department, Savova et al. [32, 33] presented a natural language processing approach. Their results proved the utility of natural language processing in extracting specific data from reports of radiology for use in cancer research and clinical decisiveness. The medical industry has benefited greatly from the advanced innovations in NLP techniques. The specialized method for healthcare text analysis is highlighted by the introduction of ClinicalBERT by Hao et al. [9], a pre-trained language model developed exclusively for clinical notes. Remmer et al. [34] offer a new method that integrates NLP techniques in healthcare classification problems by proposing clinical and linguistic embeddings for multi-label classification of medical discharge summaries. In addition, Le et al. [35] offer a thorough analysis of NLP's uses in mental health, illuminating the wide variety of applications and prospective benefits in this field.

The reviewed papers presented cutting-edge examples of NLP applications and methods in the medical field. Among these are the following: de-identification of clinical notes; use of deep learning in healthcare; identification of research hypotheses;

Table 4 Current scenario of NLP in healthcare

References	Techniques/methods	Applications/findings
Caccamisi et al. [26]	Processing linguistics data using NLP	Systematic review of smoking status
Rajkomar et al. [27]	Machine learning	Overview of machine learning in medicine
Jensen et al. [28]	EHR mining	Improved research and clinical healthcare
Kreimeyer et al. [29]	Information extraction, named entity recognition, text classification	Capturing and standardizing unstructured clinical information using NLP systems
Deleger et al. [30]	De-identification of clinical note	Impact on information extraction
Miotto et al. [31]	Deep learning	Review, opportunities, and challenges
Savova et al. [32, 33]	DeepPhe, NLP	Extraction of cancer phenotypes
Hao et al. [9]	Enriching pre-trained language models with clinical notes	ClinicalBERT for clinical language modeling
Remmer et al. [34]	Clinical and linguistic embeddings	Multi-label classification of discharge summaries
Le et al. [35]	Natural language processing (NLP)	Applications in mental health

extraction of cancer phenotypes; representation of medical concepts, and detection of medication errors from electronic health records (EHRs). These developments allow for more effective and precise analysis of clinical text data, which in turn improves healthcare service, research, and patient safety. Present NLP approaches encompasses clinical coding and extracted data from unstructured and enhance decision-making Table 4 shows the finding of related studies where NLP has been applied in healthcare.

4.2 *Benefits and Challenges of Integrating NLP into Clinical Decision Support Systems (RQ2)*

There are several upsides and caveats to incorporating NLP into CDSS, but the benefits far outweigh the difficulties. The publications that were reviewed clarified these points. The MIMIC-III critical care database was introduced by Johnson et al. [36], who showed how NLP may be used to mine useful data from clinical notes that is unstructured in nature. Clinicians can access and use a tremendous amount of clinical data by incorporating NLP into CDSS, which enhances decision-making and patient care. The identification of adverse medication events from healthcare news stories exemplifies how natural language processing (NLP) can aid CDSS in monitoring and detecting undesirable events, opening the door to preventative steps, and safeguarding patients. Hou et al. [37] gave an analysis of processing the language in healthcare, focusing on its historical, contemporary, and prospective uses. They talked about how NLP can improve CDSS by boosting decision support and advancing personalized medicine through better information extraction from clinical writing. The potential of NLP to transform unstructured clinical text into well-structured data was highlighted in an introduction to NLP provided by Nadkarni et al. [38]. The incorporation of NLP into CDSS improves the utilization of clinical information and paves the way for evidence-based decision-making. The use of unstructured clinical notes for comparative effectiveness research was discussed by Capurro et al. [39]. By facilitating the collection and analysis of data from clinical narratives, NLP can play a pivotal role in CDSS and contribute to the development of evidence-based medicine. Data mining is crucial in clinical decision support systems (CDSS). Aleksovska et al. [40] stressed the importance of NLP in clinical data extraction. NLP can show patterns, correlations, and insights that improve clinical decision-making. Ravì et al. [41] examined deep learning in healthcare informatics. The discourse revolved around the potential of deep learning and NLP techniques to augment CDSS by enabling a more precise and efficient comprehension of clinical literature and enhancing decision-making support.

The history, current, and future of artificial intelligence (AI) in healthcare were examined by Jiang et al. [5], including the integration of NLP into CDSS. They underlined how AI and NLP may enhance clinical judgment and customized care. Tervonen et al. [42] provided a methodology for evaluating the benefits and drawbacks of clinical risk prediction models, emphasizing the demand for trustworthy, and precise data. By increasing data accuracy and quality, NLP can help CDSS by upgrading risk prediction models. There are advantages and disadvantages to using NLP methods into CDSS. While highlighting the need for more study and validation, Gulum et al. [43] emphasize the potential advantages of deep learning, a subset of NLP, in providing clinical decision assistance for radiologists. Overall, incorporating NLP into CDSS has many positive outcomes, such as increased access to clinical data, better decision-making assistance, earlier identification of adverse events, more precise and individualized treatment, and better data for comparative effectiveness research. The promise of NLP in healthcare is immense, but it is not yet being fully realized because of obstacles including poor data quality, inaccurate models, and a lack of interoperability. The strengths and challenges of NLP discussed in sector of healthcare is illustrated in Table 5.

4.3 Compliance in Implementing NLP in Healthcare (RQ3)

The use of NLP in clinical healthcare environments poses key questions and concerns with relation to patients' privacy, data security, and regulatory compliance. The papers that were looked over offer some new perspectives on these essential factors. Table 6 shows the ethical consideration of implementing NLP in healthcare. Meystre et al. [6] explored the reuse of clinical data and emphasized the significance of patient

Table 5 Strength and challenges of NLP in health service

References	Strengths	Challenges
Johnson et al. [36]	Freely accessible critical care database (MIMIC-III)	Privacy and data security concerns
Hou, et al. [37]	Review of NLP applications in healthcare	Lack of standardized data and interoperability
Nadkarni et al. [38]	Introduction to natural linguistic processing	Ambiguity and context sensitivity of natural language
Capurro et al. [39]	Utilization of unstructured clinical notes for comparative effectiveness research	Difficulty in extracting structured data from unstructured text
Aleksovska et al. [40]	Data mining in clinical decision support systems	Integration and interoperability with existing healthcare systems
Ravì et al. [41]	Reviewing applications of deep learning in healthcare informatics	Deep learning models training: need for large labeled data
Jiang et al. [5]	Overview of artificial intelligence in healthcare	Ethical considerations and potential biases in AI-based decision-making
Tervonen et al. [42]	Framework for assessing strengths and weaknesses of clinical risk prediction models	Generalizability of risk prediction models to diverse patient populations
Gulum et al. [43]	Competence of deep learning for clinical decision support in radiology	Deep learning models provide a difficulty in terms of their interpretability and ability to be explained

confidentiality and privacy. Healthcare businesses must make sure that patient data is adequately de-identified and safeguarded while employing NLP in order to adhere to privacy laws.

Samwald et al. [44] looked at how patients were exposed to various medications and stressed the importance of protecting important pharmacogenomic data. To safeguard genomic material and stop illegal access, NLP implementation should have strong security features. The use of AI in healthcare was examined by Yu et al. [45], also covered the moral and legal issues around data security and privacy. When using NLP, they highlighted the significance of openness, consent, and secure data storage in order to preserve patient trust and adhere to legal standards. The difficulty of electronically sharing clinical data while safeguarding patient privacy is an important aspect. Safeguarding private medical information during NLP implementation calls for the use of data sharing agreements, encryption mechanisms, and stringent access controls. In its 2018 publication of ethics principles for reliable AI, the European Commission emphasized the significance of privacy, security, and openness. To ensure the moral and appropriate use of patient data, NLP implementations should adhere to these standards.

When implementing NLP in healthcare settings, it is important to keep patient confidentiality, data security, and government regulations in mind. Soleymani et al.

Table 6 Ethical consideration of NLP in healthcare

References	Key considerations	Recommendations
Meystre and Lovis [6]	Need for privacy protection and regulatory compliance in reusing clinical data for NLP applications	Develop data sharing policies, obtain patient consent, ensure data anonymization and de-identification
Samwald et al. [44]	Focus on the significance of privacy and security in handling pharmacogenomic data and ensuring patient confidentiality	Implement strong data encryption, access controls, and audit trails to protect sensitive genomic information
Yu et al. [45]	Highlight the ethical and legal implications of AI in healthcare, including privacy, security, and regulatory compliance	Develop transparent and accountable AI frameworks, adopt privacy-preserving techniques such as federated learning, comply with regulations and guidelines
Soleymani et al. [46]	Addresses the challenges of maintaining privacy and security while enabling analytics in healthcare data in real-time	Use secure data storage and transmission methods, implement access controls and data encryption, comply with privacy regulations

[46] highlight the significance of data security and privacy in NLP applications by discussing the difficulties of constructing privacy-preserving data systems for real-time analytics, with a focus on public health emergencies. Privacy, security, and regulatory compliance must all be taken into account when implementing NLP in healthcare settings. Protecting patient information, obtaining informed permission, complying with privacy requirements, and addressing ethical concerns all need rigorous approaches [47]. By keeping these things in mind, healthcare institutions can use NLP's advantages without compromising patient privacy or losing the public's trust.

5 Conclusion and Future Scope

In conclusion, the utilization of human linguistic processing in the healthcare industry has a great deal of potential in terms of radically transforming clinical decision support systems and the patient's treatment. Techniques that are considered to be state-of-the-art, such as ClinicalBERT and DeepNLP-PI, have shown that they are useful in radiology, in the classification of adverse events, and in the prediction of outcomes. However, challenges such as data quality, bias, interpretability, and privacy must be addressed. Future research should focus on improving data quality, developing privacy-preserving techniques, and fostering interdisciplinary collaborations. By overcoming these challenges, we can fully harness the potential of linguistic processing to transform health services and improve clinical outcomes.

References

1. Garg R, Kiwelekar AW, Netak LD, Ghodake A (2021) i-Pulse: a NLP based novel approach for employee engagement in logistics organization. Int J Inf Manag Data Insights 1(1):100011.https://doi.org/10.1016/j.jjimei.2021.100011
2. Garg R, Kiwelekar AW, Netak LD (2021) Logistics and freight transportation management: an NLP based approach for shipment tracking. Pertanika J Sci Technol 29(4). https://doi.org/10.47836/pjst.29.4.28
3. Garg R, Kiwelekar AW, Netak LD, Bhate SS (2021) potential use-cases of natural language processing for a logistics organization. pp 157–191. https://doi.org/10.1007/978-3-030-68291-0_13
4. Kaur R, Ginige JA, Obst O (2023) AI-based ICD coding and classification approaches using discharge summaries: a systematic literature review. Expert Syst Appl 213:118997.https://doi.org/10.1016/j.eswa.2022.118997
5. Jiang F et al (2017) Artificial intelligence in healthcare: past, present and future. Stroke Vasc Neurol 2(4):230–243. https://doi.org/10.1136/svn-2017-000101
6. Meystre SM, Lovis C, Bürkle T, Tognola G, Budrionis A, Lehmann CU (2017) Clinical data reuse or secondary use: current status and potential future progress. Yearb Med Inform 26(01):38–52. https://doi.org/10.15265/IY-2017-007
7. Rajkomar A et al (2018) Scalable and accurate deep learning with electronic health records. npj Digit Med 1(1):18. https://doi.org/10.1038/s41746-018-0029-1
8. Demner-Fushman D, Chapman WW, McDonald CJ (2009) What can natural language processing do for clinical decision support? J Biomed Inform 42(5):760–772. https://doi.org/10.1016/j.jbi.2009.08.007
9. Hao B, Zhu H, Paschalidis IC (2020) Enhancing clinical BERT embedding using a biomedical knowledge base. In: Coling 2020—28th international conference computer linguistics proceeding conference, pp 657–661. https://doi.org/10.18653/v1/2020.coling-main.57
10. Weng W-H, Wagholikar KB, McCray AT, Szolovits P, Chueh HC (2017) Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. BMC Med Inform Decis Mak 17(1):155. https://doi.org/10.1186/s12911-017-0556-8
11. Harpaz R et al (2014) Text mining for adverse drug events: the promise, challenges, and state of the art. Drug Saf 37(10):777–790. https://doi.org/10.1007/s40264-014-0218-z
12. Demiris G, Iribarren SJ, Sward K, Lee S, Yang R (2019) Patient generated health data use in clinical practice: a systematic review. Nurs Outlook 67(4):311–330. https://doi.org/10.1016/j.outlook.2019.04.005
13. Velupillai S, Mowery D, South BR, Kvist M, Dalianis H (2015) Recent advances in clinical natural language processing in support of semantic analysis. Yearb Med Inform 24(1):183–193. https://doi.org/10.15265/IY-2015-009
14. Milne-Ives M et al (2020) The effectiveness of artificial intelligence conversational agents in health care: systematic review. J Med Internet Res 22(10):e20346.https://doi.org/10.2196/20346
15. Laranjo L et al (2018) Conversational agents in healthcare: a systematic review. J Am Med Informatics Assoc 25(9):1248–1258. https://doi.org/10.1093/jamia/ocy072
16. Hu Z, Qin L, Xu H (2019) Association between diabetes-specific health literacy and health-related quality of life among elderly individuals with pre-diabetes in rural Hunan Province, China: a cross-sectional study. BMJ Open 9(8):e028648.https://doi.org/10.1136/bmjopen-2018-028648
17. Zeng-Treitler A, Goryachev Q, Kim S, Keselman H (2007) Making texts in electronic health records comprehensible to consumers: a prototype translator. In: AMIA 2007 symposium proceedings, vol 21, no 1, pp 846 pmid: 18693956
18. Wallace BC et al (2012) Toward modernizing the systematic review pipeline in genetics: efficient updating via data mining. Genet Med 14(7):663–669. https://doi.org/10.1038/gim.2012.7

19. Khanbhai M, Anyadi P, Symons J, Flott K, Darzi A, Mayer E (2021) Applying natural language processing and machine learning techniques to patient experience feedback: a systematic review. BMJ Heal Care Inform 28(1):e100262.https://doi.org/10.1136/bmjhci-2020-100262
20. Hripcsak G et al (2015) Observational health data sciences and informatics (OHDSI): opportunities for observational researchers. Stud Health Technol Inform 216:574–578. https://doi.org/10.3233/978-1-61499-564-7-574
21. Zhang K, Demner-Fushman D (2017) Automated classification of eligibility criteria in clinical trials to facilitate patient-trial matching for specific patient populations. J Am Med Inform Assoc 24(4):781–787. https://doi.org/10.1093/jamia/ocw176
22. Yu H et al (2019) ODAE: Ontology-based systematic representation and analysis of drug adverse events and its usage in study of adverse events given different patient age and disease conditions. BMC Bioinformatics 20(S7):199. https://doi.org/10.1186/s12859-019-2729-1
23. Zhang T et al (2021) Identifying adverse drug reaction entities from social media with adversarial transfer learning model. Neurocomputing 453:254–262. https://doi.org/10.1016/j.neucom.2021.05.007
24. Luo Y et al (2017) Natural language processing for EHR-based pharmacovigilance: a structured review. Drug Saf 40(11):1075–1089. https://doi.org/10.1007/s40264-017-0558-6
25. Chu J, Dong W, He K, Duan H, Huang Z (2018) Using neural attention networks to detect adverse medical events from electronic health records. J Biomed Inform 87:118–130. https://doi.org/10.1016/j.jbi.2018.10.002
26. Caccamisi A, Jørgensen L, Dalianis H, Rosenlund M (2020) Natural language processing and machine learning to enable automatic extraction and classification of patients' smoking status from electronic medical records. Ups J Med Sci 125(4):316–324. https://doi.org/10.1080/03009734.2020.1792010
27. Rajkomar A, Dean J, Kohane I (2019) Machine learning in medicine. N Engl J Med 380(14):1347–1358. https://doi.org/10.1056/NEJMra1814259
28. Jensen PB, Jensen LJ, Brunak S (2012) Mining electronic health records: towards better research applications and clinical care. Nat Rev Genet 13(6):395–405. https://doi.org/10.1038/nrg3208
29. Kreimeyer K et al (2017) Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review. J Biomed Inform 73:14–29. https://doi.org/10.1016/j.jbi.2017.07.012
30. Deleger L et al (2013) Large-scale evaluation of automated clinical note de-identification and its impact on information extraction. J Am Med Inform Assoc 20(1):84–94. https://doi.org/10.1136/amiajnl-2012-001012
31. Miotto R, Wang F, Wang S, Jiang X, Dudley JT (2018) Deep learning for healthcare: review, opportunities and challenges. Brief Bioinform 19(6):1236–1246. https://doi.org/10.1093/bib/bbx044
32. Savova GK et al (2017) DeepPhe: a natural language processing system for extracting cancer phenotypes from clinical records. Cancer Res 77(21):e115–e118. https://doi.org/10.1158/0008-5472.CAN-17-0615
33. Savova GK et al (2019) Use of natural language processing to extract clinical cancer phenotypes from electronic medical records. Cancer Res 79(21):5463–5470. https://doi.org/10.1158/0008-5472.CAN-19-0579
34. Remmer S, Lamproudis A, Dalianis H (2021) Multi-label diagnosis classification of Swedish discharge summaries—ICD-10 code assignment using KB-BERT. In: Proceedings of the conference recent advances in natural language processing—deep learning for natural language processing methods and applications, pp 1158–1166. https://doi.org/10.26615/978-954-452-072-4_130
35. Le Glaz A et al (2021) Machine learning and natural language processing in mental health: systematic review. J Med Int Res 23(5):e15708. https://doi.org/10.2196/15708
36. Johnson AEW et al (2016) MIMIC-III, a freely accessible critical care database. Sci Data 3(1):160035.https://doi.org/10.1038/sdata.2016.35

37. Hou JK, Imler TD, Imperiale TF (2014) Current and future applications of natural language processing in the field of digestive diseases. Clin Gastroenterol Hepatol 12(8):1257–1261. https://doi.org/10.1016/j.cgh.2014.05.013
38. Nadkarni PM, Ohno-Machado L, Chapman WW (2011) Natural language processing: an introduction. J Am Med Inform Assoc 18(5):544–551. https://doi.org/10.1136/amiajnl-2011-000464
39. Capurro D, Yetisgen M, Eaton E, Black R, Tarczy-Hornoch P (2014) Availability of structured and unstructured clinical data for comparative effectiveness research and quality improvement: a multi-site assessment. eGEMs (Generating Evid. Methods to Improv. patient outcomes) 2(1):11. https://doi.org/10.13063/2327-9214.1079
40. Aleksovska-Stojkovska L, Loskovska S (2013) Data mining in clinical decision support systems, pp 287–293
41. Ravi D et al (2017) Deep learning for health informatics. IEEE J Biomed Heal Inform 21(1):4–21. https://doi.org/10.1109/JBHI.2016.2636665
42. Tervonen et al (2015) Applying multiple criteria decision analysis to comparative benefit-risk assessment. Med Decis Mak 35(7):859–871. https://doi.org/10.1177/0272989X15587005
43. Gulum MA, Trombley CM, Kantardzic M (2021) A review of explainable deep learning cancer detection models in medical imaging. Appl Sci 11(10):4573. https://doi.org/10.3390/app11104573
44. Samwald M et al (2016) Incidence of exposure of patients in the united states to multiple drugs for which pharmacogenomic guidelines are available. PLoS One 11(10):e0164972.https://doi.org/10.1371/journal.pone.0164972
45. Yu K-H, Beam AL, Kohane IS (2018) Artificial intelligence in healthcare. Nat Biomed Eng 2(10):719–731. https://doi.org/10.1038/s41551-018-0305-z
46. Soleymani SA, Goudarzi S, Anisi MH, Jindal A, Kama N, Ismail SA (2023) A privacy-preserving authentication scheme for real-time medical monitoring systems. IEEE J Biomed Heal Inform 27(5):2314–2322. https://doi.org/10.1109/JBHI.2022.3143207
47. El Emam K, Jonker E, Arbuckle L, Malin B (2011) A systematic review of re-identification attacks on health data. PLoS One 6(12):e28071. https://doi.org/10.1371/journal.pone.0028071

An Investigational Analysis of Automatic Speech Recognition on Deep Neural Networks and Gated Recurrent Unit Model

M. Soundarya and S. Anusuya

Abstract For thousands of years, communication has played a crucial role in human existence, development, and globalization. Speech recognition has several uses, including biometric analysis, education, security, health care, and smart cities. Many scientists have spent years studying how machine learning may be applied to speech processing, particularly voice recognition. But in recent years, researchers have concentrated on ways to apply deep learning to problems involving human speech. In this post, we discuss our work using deep neural networks like CRNN and GRU to recognize audio samples in spoken language. Seven different classes of audio samples (Walk & footsteps, Kids speaking, Filling with water, Bass drum, Scissors, Clock, and Cough) were employed in Free Sound Datasets. Mel-spectral coefficients, along with other spectral and intensity-related factors, are among the feature parameters utilized for recognition. White noise and a retuned voice were employed as data augmentation. An average recognition rate of accuracy 93.25% and WER—Word Error Rate—of 7.84% were obtained by the GRU model, according to the findings.

Keywords Deep learning · Speech recognition · MFCC · Gated recurrent unit · Feature extraction · And data augmentation

1 Introduction

Speech, being the most fundamental and innate mode of human communication, can communicate crucial information rapidly and correctly. People nowadays are investing the time and energy necessary to master the skill of voice control for a wide

M. Soundarya (✉) · S. Anusuya
Saveetha School of Engineering, SIMATS, Chennai, India
e-mail: soundaryam2009.sse@saveetha.com

S. Anusuya
e-mail: anusuyas.sse@saveetha.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_4

variety of smart devices. Over fifty percent of the global population communicates in just few languages, despite the widespread perception that there are many more languages spoken than there actually are. As a result of their widespread use and the abundance of data available for them, systems based on Artificial Intelligence (AI) have been developed for speech identification, text-to-speech synthesis, natural language processing, and computational linguistics in these dialects [1]. However, less-used languages sometimes lack the funding necessary to conduct specialized technological research and development. Therefore, it is a tough and noteworthy challenge to design analogous techniques for low-resource languages. Language has been used by humans for a long time as a means of communication and interaction [2]. It acts as a conduit for the exchange of ideas across different cultures, so fostering their growth and progress. Artificial Intelligence technology based on deep learning has also progressed rapidly and achieved a significant transformation from the theoretical research level to the practical application level in the past few years, thanks in large part to the rise of data technologies such as cloud computing, big data, and the Internet of Things.

The automated voice identification system has likewise gone from “unusable” to “obtainable” with the use of AI technology, demonstrating very high application value and strong development possibilities [3]. It has been challenging for automated speech recognition technologies to reach everyday life since the typical automatic speech recognition model has an intricate framework and demands a lot of computation and storage capabilities. Nevertheless, with the advent of deep learning, the amount of information of training models has been substantially decreased, and even a deep learning automated speech recognition model, like Google’s voice assistant model, which is 80M in size, can be put into a mobile device. The fast advancement of automatic voice recognition can be directly linked to the usage of algorithms that depends on deep learning in recognize languages [4].

Most state-of-the-art automatic speech recognition systems employ some combination of HMMs—Hidden Markov Models, GMMs—Gaussian Mixture Models, and DNNs—deep neural networks. Due in large part to the development of specialized neural network models in various training and categorization strategies, DNNs are an integral aspect of ASR system construction. In addition to these uses, they have been implemented for issues such feature extraction, audio signal classification, text recognition and TTS, processing of disordered speech, and vocabulary-based voice recognition [5]. However, ASR system performance is heavily influenced by the speech datasets used to train DNN models. Speech recognition technology is an area of computer science concerned with creating computer systems that can understand human speech. It is important to keep in mind that the term “voice recognition” refers simply to the ability of a computer to transcribe spoken words. The branch of computer science concerned with understanding human languages is known as natural language processing. There are a variety of speech recognition tools now on the market. Words are no problem for the most advanced systems [6]. To get the most out of them, though, you will need to put in some serious time training the computer to recognize your voice and accent. It is believed that these kinds of systems rely too much on the speaker. The speaker must also enunciate each phrase clearly and pause briefly between them, as this is a requirement of many systems. Discrete speech systems are what they sound like. Recent years have seen significant development in continuous speech systems, or natural-sounding voice recognition software. Several different continuous speech systems [7] are now accessible for use on desktop PCs.

Voice recognition systems have always been reserved for very specific applications due to their high price and limited functionality. Such systems are helpful, for instance, when the user is unable to use a keyboard to enter information because his or her hands are otherwise engaged or because the user has a disability. A headset allows the user to voice instructions rather than type them. However, as the price of voice recognition systems drops and their performance increases, they are becoming more widely used as a viable alternative to keyboards. It is common knowledge that a speaker's voice reflects the speaker's individuality through characteristics such as the speaker's shape of vocal tract, size of larynx, rhythm, and various accent [8]. Therefore, a computer can be used to automatically identify the speaker based on their voice. The focus of this study is on this method, which is known as automated speaker recognition. Human-made speaker recognition systems are not discussed. There are many practical uses for the rudimentary but important job of speaker detection in the field of voice processing. It is used for voice-based identification of mobile devices, cars, and computers, for instance. It ensures the safety of online banking and money transfers. It has found extensive use in fields like forensics, where it is used to determine whether a person is guilty, as well as in surveillance and automatic identification tagging. It is useful for finding things like phone conversations, meetings, and radio broadcasts using audio-based databases. It may also be used as an ASR frontend to boost the accuracy of transcriptions of conversations involving several speakers. The high survivability in voice signals makes ASR, or the translation of uttered words into text, a difficult process. People can talk in a variety of ways, including with various accents, dialects, pronunciations, styles, rates, and emotions. Noise and reverberation in the recording space, as well as the use of several microphones and playback devices, all contribute to an already variable final product. In addition, voice processing, ASR's many practical applications in areas like security, education, smart healthcare, and smart cities make it a dynamic and vital research topic. It is a set of methods that work together to turn sound waves into written words by applying text matching to the identified speech signal. By using a consistent foundation for in-depth semantic learning, automatic speech recognition (ASR) aims to transform audio signals into text.

ASR incorporates several area of studies, such as computer science, DSP, acoustics, AI, languages, statistics, and more. HMMs—Hidden Markov Models—based on the GMM—Gaussian Mixture Model [9]—are typically used in traditional speech recognition systems to describe the temporal organization of speech data. Since a speech signal may be interpreted as a short-time or piecewise stationary signal, HMMs are useful in this application [10]. On a small enough time, scale, human speech may be modeled as a steady state. For many stochastic goals, the human voice may be seen as a Markov model. Each HMM state typically employs a Gaussian mixture to simulate the sound wave's spectral representation. Systems that recognize language based on HMMs are easy to use, take little time to train, and are theoretically viable. Gaussian mixture models, however, have a significant drawback in that they are numerically unproductive when trying to represent data that lies on or near an irregular manifold in the data space. While HMMs assume certain statistical qualities of features, neural networks do not. Neural networks provide efficient and natural discriminative training for estimating the probability of a voice feature segment [11]. While neural networks outrival in classification of discrete time units such as words and phones, but they suffer from the issue of temporal associations. So, as an alternative, neural networks may be used to do preprocessing, such as feature modification and dimensionality reduction, before HMM-based recognition is performed. The deep neural networks [12] are widely used for recognition of image and voice in an efficient manner. We focused on speech recognition and will provide our latest findings on using deep neural networks based on the deep neural network and the Gated Recurrent Unit Model.

2 Related Works

Many researchers are hard at work about computer science known as image processing and speech synthesis. An image is a grid of pixels, each of which has a numerical value that indicates its significance in the overall picture. Images are obtained using a wide variety of techniques, and the general public has little trouble recognizing the persons shown. Character recognition in pictures is impossible for the blind and the uneducated. As a result, in article [13], the author offers a prototype for a system that can read aloud the characters/text in the photographs. Optical Character Recognition and speech synthesis models form the backbone of the system. Through steps including preprocessing, segmentation, and classification, the Optical Character Recognition model can decipher the text contained in photos and turn it into a format that can be edited. The voice synthesis model takes the user-edited text as input and outputs a signal that may be understood as speech. Both the deep and machine learnings are utilized to train these models. There is a about 90% degree of accuracy with these models, and the generated speech sounds very much like the input speech.

Some areas, such as study into how people interact with computers, have begun to incorporate emotional considerations as a result of globalization. Expressions on people's faces are often the only clue we have as to how they are feeling. Another option is to deduce a person's state of mind from their vocalizations. In study [14], the Mel-Frequency Cepstral Coefficient (MFCC) was employed as a feature extraction approach in a human emotion recognition system based on acoustic data. MFCC was selected to simulating the response of the human auditory system. Many research projects employ support vector machines (SVMs), a kind of supervised machine learning, to categorize human speech. The RBF kernel is a popular choice for SVM Multi-class that has been utilized in several prior researches. This is since SVM improves precision utilizing the RBF—Radial Basis Function kernel. By a 0.001-s size of frame, 1.0 C values, the ratio of accuracy maximum for the investigation is 0.72.

The development of machine learning techniques has paved the way for the creation of increasingly practical intelligent systems. Intelligent systems are those that can communicate with their users, learn from their input, and improve over time. Therefore, a novel computational algorithm is developed for audio classes. A feature extraction procedure is applied to a speech signal. The LSTM approach [15] is used to classify a speech corpus, and its results are compared to those of the more traditional support vector machine (SVM) method [16]. No obvious research has been done to compare feature-generation techniques with ML models to establish which one achieved best on a generic public database. By means of the method SVM—support vector machine, the outcomes of the tests provide that the bigram attributes' set yielded the highest overall accuracy (79%) when used with the bigram feature. The findings of these comparisons will be useful as baseline for further work on text categorization systems. Machine learning includes applications like speech detection and text classification using natural language processing.

Reliable models for predicting and classifying decisions enhance behaviors and reduce reliance on human experience, while the voice recognition process is preserved as a distinct issue. The voice recognition solution suggests differentiating between text and speech, both of which are increasingly limited in their ability to carry out their intended tasks. The lecture provided by author [17] demonstrates how to make accurate predictions in a CNN-based framework. In this regard, the suggested enhancements to voice recognition using a training dataset and discriminative models are successful. For continuous voice recognition with large amounts of text, SSVM is a viable option [18]. It is feasible to incorporate convex optimization approach into the training process, allowing for the extraction of SSVM features that provide continuous speech recognition with optimum segmentation. Speech recognition datasets may be interpreted with the use of supervised learning and artificial neural networks. In this study, SSVM was used for voice recognition with successful protocol adherence and satisfactory results. This study also provides an improved method of using CNN for voice recognition, leading to an optimal performance calculation.

3 Methodology

3.1 Dataset

The Audio Set-Ontology has an ordered group of about 600 classes of audio sounds and the samples' counts to 297,144. The FSD-Free Sound Dataset constitutes different types of sounds of things, natural events, animals, human sound in its samples [19]. The annotations count to about 685,403. The process of annotation is manually done and is available open as FSD50K. It includes clippings of audio of

Table 1 Data distribution for seven classes in datasets

Classes	Audio samples
Walk and footsteps	580
Kids speaking	255
Filling with water	148
Bass drum	339
Scissors	165
Clock	349
Cough	385

about 51,197 under class count of 200 recorded 100 h. The FSD50K comprises two modes, The Development and the Evaluation. Furthermore, FSD50K also consists of the supplementary data known as metadata and is free to download. Table 1 displays the sample size distribution for seven classes of audio sample. Figure 1 shows the frequency distribution of the same seven classes. The waveform for Bass drum class is seen in Fig. 2.

The seven classes of the audio samples are listed as: Walk and footsteps (580), Kids speaking (255), Filling with water (148), Bass drum (339), Scissors (165), clock (349), cough (385). Data augmentation was used to double the number of files associated with each audio class.

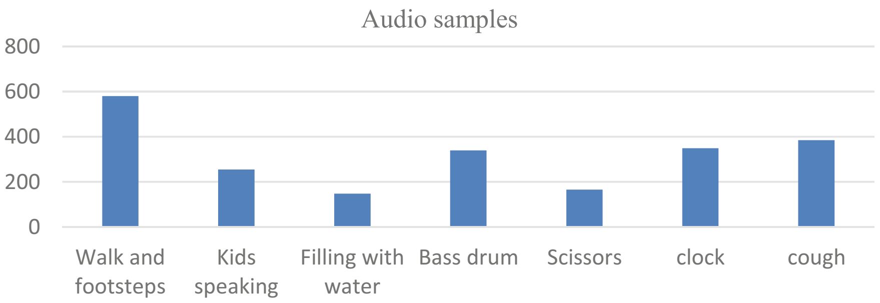


Fig. 1 Data distribution of audio samples in seven classes

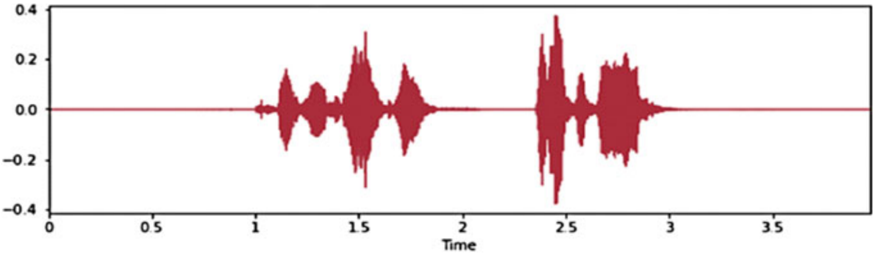


Fig. 2 Wave plot for audio for Bass drum class

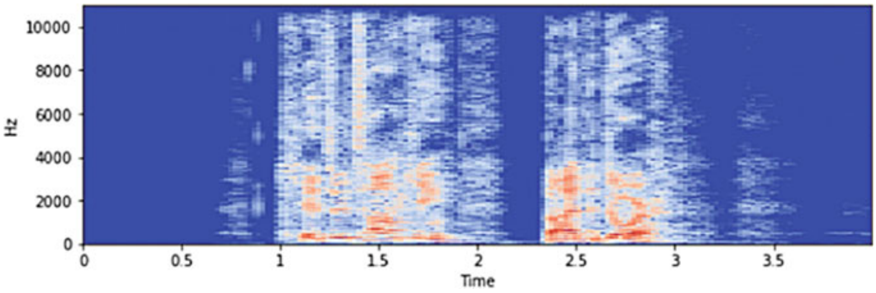


Fig. 3 Spectrogram for audio for Bass drum class

3.2 Data Augmentation

Using data augmentation, we may increase the amount of available data for training. Deep learning algorithms can only produce accurate forecasts when exposed to vast volumes of training data. So, we add extra details to the existing information to increase the model's adaptability. Figure 3 is a spectrogram of audio for Bass drum class. Image augmentation is a highly effective method for augmenting datasets with new information by creating fake variances in the source data. This generates fresh, original pictures from the data image collection, which itself offers a vast array of potential pictures. In order to achieve this effect, many transformation techniques are used, such as zooming in on the current picture, rotating the current image by a little amount, shearing, or cropping the current collection of photos.

3.3 Feature Extraction

Feature extraction plays a crucial role in analysis and discovering connections between variables. Since the algorithms are not capable of immediately comprehending the audio information presented, feature identification is utilized to transform the input into a form that can be comprehended. Time, amplitude, and frequency can all be represented along the audio signal's three-dimensional axes. To create smart audio mechanisms, we need audio features, which are descriptions of sounds or audio signals that may be input into statistical or ML models. The Amplitude Envelope of a signal is the sum of the largest amplitudes found in any one frame is worth of data. The volume level is approximately estimated by this function. However, it is quite susceptible to extreme values. Both onset detection and the categorization of musical genres make heavy use of this characteristic.

All samples in a frame contribute to the Root Mean Square Energy. It serves as a measure of volume since the greater the energy, the more audible the sound. However, unlike the Amplitude Envelope, it is less affected by extreme values. This quality has proven effective in applications such as audio segmentation and genre categorization. Mel-spectral coefficients are indicators of the frequency distribution of a signal. The degree to which a sound may be described as noise is measured by its spectral flatness. The extent of the spectrum may be measured by its spectral bandwidth. A spectral centroid is calculated by treating a magnitude spectrogram and then computing the mean of each frame individually. Each spectrogram frame is split into sub-bands so that the spectral contrast may be calculated. The energy contrast is computed for each sub-band by contrasting the mean energy of the top and bottom quantiles (peak and valley, respectively). Clear, narrow-band signals are often associated with high contrast numbers, while low-contrast numbers are associated with broadband noise. In a frequency response curve, the roll-off frequency indicates the approximate lower and upper bounds. Each sound has a unique frequency that contributes to its pitch, which in turn affects the heard sound's bass or treble. These frequencies determine the sound's pitch; as the frequency of the source increases, so does the perceived frequency of the sound, and vice versa. Frame-Relative Mean Squared (FRMS) is an RMS value that is determined for each frame.

Librosa was used to include white noise into the original voice signal. When compared to the loudest point of the spoken signal, the white-noise amplitude was just 3%. The average S/N ratio dropped by 5.73 dB when noise was introduced. Following Formula (1), the S/N ratio was determined.

$$\left(\frac{S}{N}\right)_{dB} = 10\log_{10}\left(\frac{PW_S}{PW_N}\right), \tag{1}$$

in which the power of the signal, PW_S, is related to the power of the noise, PW_N. Since the signal strength is unaffected by the noise, the pre-noise-to-noise power ratio is:

$$\frac{PW_N\text{with noise}}{PW_N\text{without noise}} = 10^{\frac{5.73}{10}} \approx 3.74. \tag{2}$$

Audio class analysis was transformed into analysis of picture recognition. Images representing each emotion's feature criteria were evaluated. A wav file's total number of picture components were calculated by multiplying its total number of attributes by its total number of frames. We arbitrarily choose 150 as the number of Mel-spectral factors and found that wav files, on average, contain 410 frames. To keep the total no. of frames consistent across all wav files, the frame shifting will adapt according to the file size. After reducing the challenge of class identification to that of picture recognition, a suitable model may be chosen for the task.

3.4 Deep Learning Model

Every one of the models in such a network has a layer performing convolution followed by pooling, making it a discriminative deep architecture. The layer

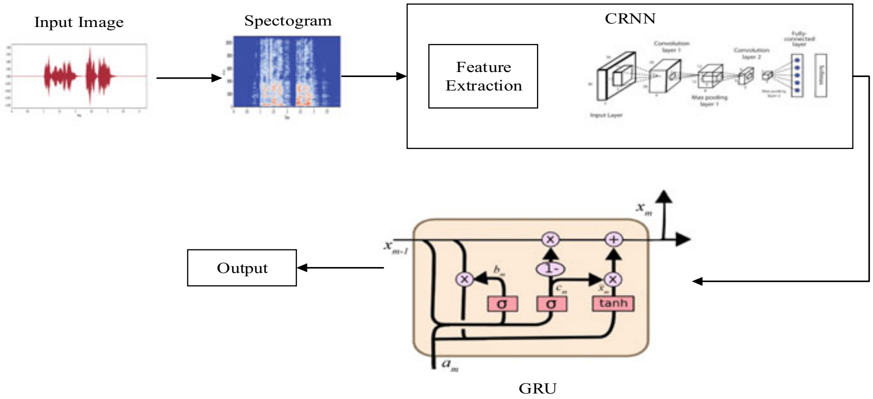


Fig. 4 Architecture of proposed model

performing convolution stocks numerous weights, but it is reduced in case of layer performing pooling and the data rate of the layer below is reduced by taking a subsample of the output from the layer of convolution. When the weights are distributed evenly and pooling algorithms are carefully selected, the CNN acquires invariance features. For complex pattern recognition tasks, some have suggested that CNN's level of invariance is insufficient. However, the CNNs have proven useful in computer vision and image identification applications. In addition, the CNN may be used for voice recognition with simple tweaks to the image analysis version of the network so that it contains speech features. Because of its use of parameter sharing, sparse interactions, and equivariant representation, the operator of convolution (which is really a specialized linear operator) is attractive for such applications. The structure of a deep learning model is seen in Fig. 4.

We conducted research with two different types that belongs to deep neural networks (CRNN and GRU). A deep neural network is, in the simplest words, a neural network that is more sophisticated than a standard neural network. In this situation, we may imagine audio sample class recognition in the same vein as picture recognition. Let us pretend that there are n frames of voice data in each wav file. Each of the image's n columns is represented by one of these n frames. The speech signal for a given time interval is represented by m feature parameters, one of which is included in each frame. There is one "pixel" for every feature parameter, or there are m elements across m rows in every picture column. The forward pass of a conventional CNN is rather simple: we feed an image into the network, the network extracts feature mappings, and we predict labels on the output. Except for the initial layer, whose input is the target picture, subsequent convolutional layers use the outcome of the preceding layer to generate a feature map, which is then fed into the model. If the CNN network is L-layered, then each layer will have L connections leading to the next layer.

Recurrent neural networks (RNNs) are created by repeatedly applying the same set of weights to a tree-like structure, with the tree being explored in topological

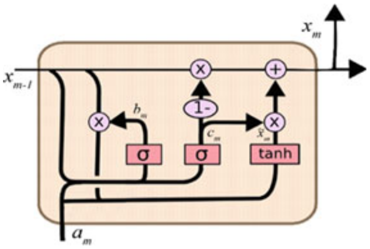


Fig. 5 Gated recurrent unit

order. The RNN is mostly employed for data sequence prediction by using past data examples. When it involves modeling sequence information like speech or text, the RNN is quite popular. However, until recently, these networks were not extensively employed because of how challenging it is to train them in a way that accurately represents dependencies across time. Recent developments in Hessian-free optimization, such as the use of approximations to second-order information or stochastic curvature estimations, have helped to overcome this difficulty.

Recent research has shown that RNNs optimized without using the Hessian are effective at producing text strings. Figure 5 depicts the GRU. To solve the vanishing gradient issue of a regular RNN, the GRU employs the gate, updates the gate, and resets the gate. These gates generate two vectors that are used to choose what data are sent to the results. These gates permit long-term information retention during training without the need to deal with diminishing gradients or throw away data that is no longer useful for forecasting. The function of update gate is to assist the storing mechanism of data from previous time steps for the purpose of prediction.

To fix the RNN's vanishing gradient issue, the GRU constantly adjusts the gate and resets it. In order to determine what data will be sent to the output, these gates generate two vectors. Training with persistent data retention is made possible by these gates, which prevent vanishing gradients and the need to discard data after it is no longer useful for forecasting. When deciding how much data from earlier time steps to keep in the model going ahead, the update gate is a useful tool. The layers employed by the CRNN and GRU are described in further depth below. The input picture for the first layer in the S1 variable set was 372 pixels wide by 128 pixels high. After convolution with a movable 3-by-3 filter and padding, the CRNN and GRU each have 64 feature maps measuring 372 by 128. The purpose of batch normalization is to establish a consistent distribution of activation values across training for each layer, which in turn results in a significant training time reduction. ELU accelerates the learning process in deep neural networks, resulting in improved classification accuracy. Max pooling decreases the amount of model parameters while simultaneously rendering feature identification scale- and orientation-insensitive. Overfitting in neural networks may be avoided using dropout.

4 Results and Discussions

The test is conducted on Intel Core i5 CPU, 16 GB—RAM, 2 TB for storage, and an NVIDIA Graphics Card. Python and Jupyter Notebook are used to run the code. Tenfold cross-validation was used on the data sets. Ten folds were culled down to just one for testing purposes. Other folds were used for validation, while the other eight were put to good use in the training process. Each of the eight folds was taken out in turn. When compared to alternative strategies, including a straightforward train/test split, cross-validation was selected because it produces a more accurate evaluation of the model's efficacy. Both CRNN and proposed models with 170 attributers took about 28.18 and 23.4 s to train for one epoch, on mean. For this reason, the proposed model's executional effectiveness was the quickest on average, while the CRNN's were the slowest. The average training time for single iteration for the GRU model with 148 parameters and no data augmentation was only 2.88 s. Figure 6 depits the input image and the same image after noise reduction, stretching and shifting is depicted in Figs. 7, 8 and 9 respectively.

Recognition accuracies for CRNN and suggested models are summarized in Table 2. According to Table 2, the proposed model with 148 attributes had the

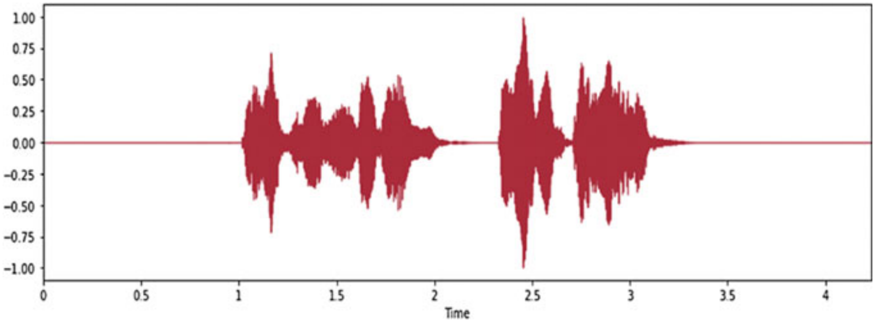


Fig. 6 Input image

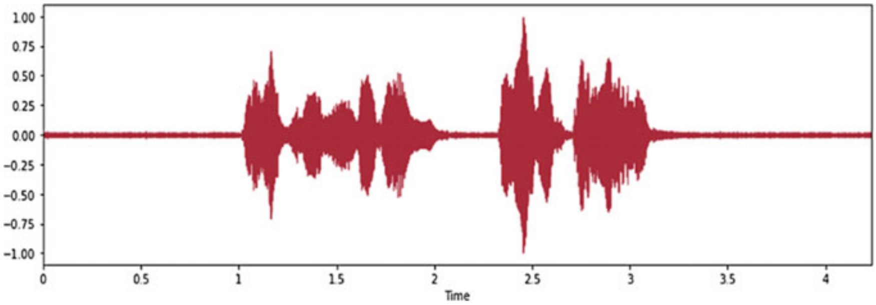


Fig. 7 After noise reduction

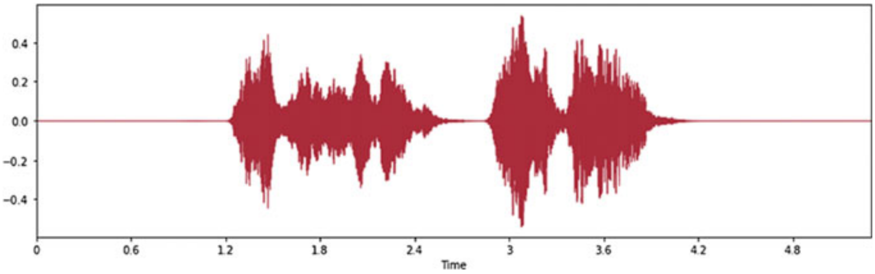


Fig. 8 After image stretching

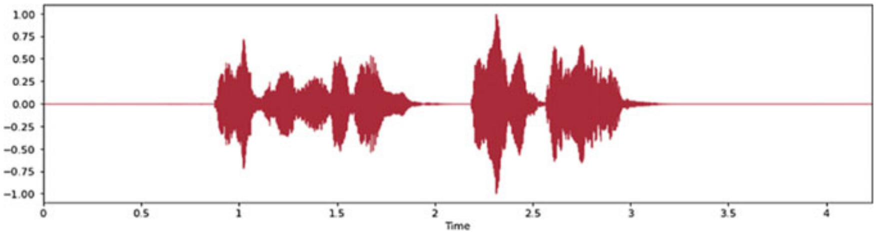


Fig. 9 Image shifting

maximum recognition accuracy of 93.25%. The accuracy rate of the CRNN and proposed models both improved as the number of parameters grew from 122 to 148; however, the average recognition accuracy of the CRNN model fell as the number of parameters grew. The proposed model can recall the past so that the process of predictions about the future becomes easier simultaneously eliminate the vanishing gradient which seems to be the best option here.

Tables 3 and 4 show the accuracy of CRNN and the proposed model with 148 parameters in terms of F1-score, recall, also area under the curve, respectively. Tables 3 and 4 show that the acquired values for recall, precision, and f1-score,

Table 2 Accuracy comparison between CRNN and proposed models

Fold	CRNN	Proposed model
0	91.26	92.78
1	91.47	93.98
2	93.75	93.52
3	92.78	94.32
4	90.85	92.89
5	90.83	91.65
6	91.72	93.65
7	91.26	92.78
8	91.47	93.96

Table 3 Performance metrics for CRNN model with 148 attributes

Types	Precision	Recall	F1-score
Walk and footsteps	89.12	89.45	89.76
Kids speaking	90.09	90.21	90.22
Filling with water	89.98	90.31	90.75
Bass drum	90.65	90.21	89.99
Scissors	91.11	91.01	90.92
Clock	90.41	91.33	91.01
Cough	91.43	91.06	91.82
Walk and footsteps	91.07	90.11	90.51

Table 4 Performance metrics for proposed model with 148 attributes

Types	Precision	Recall	F1-score
Walk and footsteps	92.15	92.57	92.81
Kids speaking	93.12	93.33	93.27
Filling with water	93.01	93.43	93.81
Bass drum	93.68	93.32	93.04
Scissors	94.13	94.13	93.95
Clock	93.43	94.45	94.06
Cough	94.46	94.18	94.87
Walk and footsteps	94.10	93.23	93.56

and also the AUC values, were all extremely near to 1. Tables 3 and 4 demonstrate that both the CRNN and the proposed models agree that the class "Cough" has the highest recall and f1-score, whereas the class "Walk & Footsteps" has the lowest recall and f1-score. The accuracy was best for the class "Bass drum" worst for the class "Scissors." Tables 3 and 4 portray these extremes.

Figure 10 displays some sample loss and precision distributions as a function of iteration for training, validation, and the confusion matrix for a single fold. Figure 10 demonstrates that the range of validation loss is consistent with the range of training loss. Effectiveness for both validation and training was high. Therefore, overfitting was not an issue. When it comes to representing the accuracy of classification models, a confusion matrix is by far the most often used tool. In Fig. 11, we see a test fold depicted by the confusion matrix.

There is a table that summarizes the results of a model for categorization called a confusion matrix. It may be used for both two- and three-way categorizations. Total TPs and TNs, or correct predictions, are displayed in the confusion matrix. Also displayed are the model's inaccuracies, such as FP—False Positives and FN—False Negatives or neglected instances. Classification quality measures like accuracy and recall may be computed using TP, TN, FP, and FN.

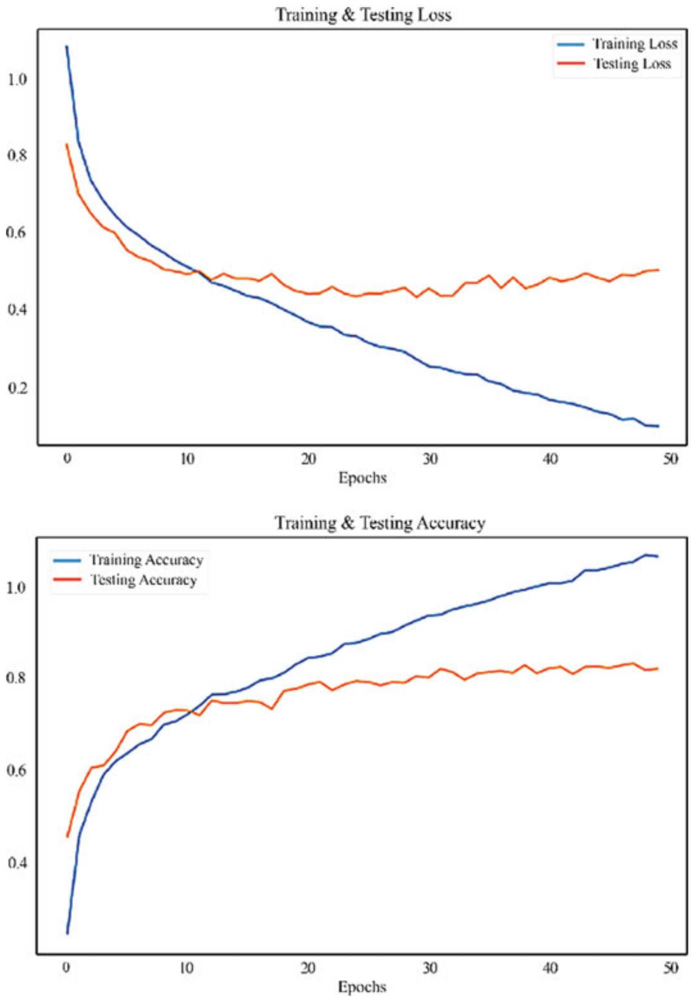
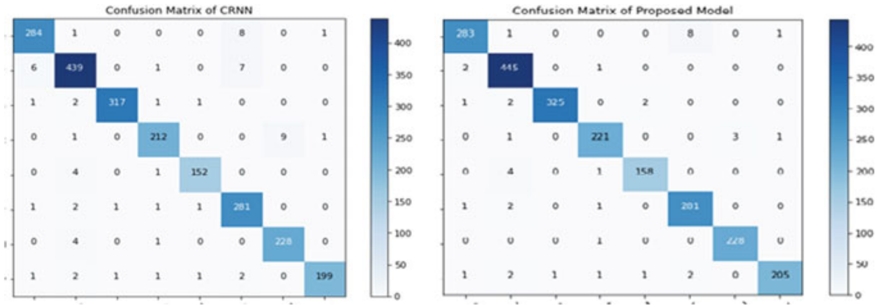


Fig. 10 Accuracy and loss variations for training and validation


Confusion Matrix of CRNN

284	1	0	0	0	8	0	1
6	439	0	1	0	7	0	0
1	2	317	1	1	0	0	0
0	1	0	212	0	0	9	1
0	4	0	1	152	0	0	0
1	2	1	1	1	281	0	0
0	4	0	1	0	0	228	0
1	2	1	1	1	2	0	199

Confusion Matrix of Proposed Model

283	1	0	0	0	8	0	1
2	445	0	1	0	0	0	0
1	2	325	0	2	0	0	0
0	1	0	221	0	0	3	1
0	4	0	1	158	0	0	0
1	2	0	1	0	201	0	0
0	0	0	1	0	0	228	0
1	2	1	1	1	2	0	205

Fig. 11 Confusion matrix

5 Conclusion

For reliable end-to-end ASR—Automatic Speech Recognition, the integrated training framework for voice augmentation and recognition approaches has achieved pretty good performances. The voice recognition component of these approaches is impacted by the issue of speech distortion since they only use the improved attribute as input. Two DNN—deep neural network models, a CRNN—Convolutional Recurrent Neural Network and a GRU—General Recurrent Unit, were utilized for audio sample class identification in our scenario, with the proposed model generally performing somewhat better than CRNN models. Data augmentation which includes modifying voice improved the recognition accuracy. The average rate of recognition was improved by using spectral aspects of the speech stream in addition to the Mel-spectral coefficients. Our study shows that the proposed approach is superior to the existing methods. In the future, we plan to investigate audio detection on data including a wider range of classes, in conjunction with speech synthesis.

References

1. Hu W, Qian Y, Soong FK, Wang Y (2015) Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. Speech Commun 67:154–166. https://doi.org/10.1016/j.specom.2014.12.008
2. Wang D, Wang X, Lv S (2019) An overview of end-to-end automatic speech recognition. Symmetry 11:1018. https://doi.org/10.3390/sym11081018
3. Khamparia A, Gupta D, Nguyen NG, Khanna A, Pandey B, Tiwari P (2019) Sound classification using convolutional neural network and tensor deep stacking network. IEEE Access 7:7717–7727
4. Yu C, Kang M, Chen Y, Wu J, Zhao X (2020) Acoustic modeling based on deep learning for low-resource speech recognition-an overview. IEEE Access 8:163829–163843
5. Zhang F, Xie X, Quan X (2022) Chinese dialect speech recognition based on end-to-end machine learning. In: 2022 international conference on machine learning, control, and robotics (MLCR), Suzhou, China, pp 14–18. https://doi.org/10.1109/MLCR57210.2022.00012
6. Phillips JM, Conrad JM (2022) Robotic system control using embedded machine learning and speech recognition. In: 2022 IEEE 19th international conference on smart communities: improving quality of life using ICT, IoT and AI (HONET), Marietta, GA, USA, pp 214–218. https://doi.org/10.1109/HONET56683.2022.10019106
7. Sheikhan M, Tebyani M, Lotfizad M (2023) Continuous speech recognition and syntactic processing in iranian farsi language. Inter J Speech Technol 1(2):135. https://doi.org/10.1007/BF02277194M
8. Xuesong Y, Kartik A, Andrew R, Samuel T, Bhuvana R, Mark H-J (2018) Joint modeling of accents and acoustics for multi-accent speech recognition. In: Proceedings of ICASSP, pp 1–5. https://doi.org/10.1109/ICASSP.2018.8462557
9. Ruiz B, Domingo P, Hernandez L (1999) A dual speech/speaker recognition using GMM in speaker identification and a HMM in keyword speech recognition. In: Proceedings IEEE 33rd annual 1999 international carnahan conference on security technology (Cat. No.99CH36303), Madrid, Spain, pp 251–254. https://doi.org/10.1109/CCST.1999.797922

10. Garud A, Bang A, Joshi S (2018) Development of HMM based automatic speech recognition system for Indian English. In: 2018 fourth international conference on computing communication control and automation (ICCUBEA), Pune, India, pp 1–6. https://doi.org/10.1109/ICCUBEA.2018.8697485
11. Tsenov GT, Mladenov VM (2010) Speech recognition using neural networks. In: 10th symposium on neural network applications in electrical engineering, Belgrade, Serbia, pp 181–186. https://doi.org/10.1109/NEUREL.2010.5644073
12. Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: a systematic review. IEEE Access 7:19143–19165. https://doi.org/10.1109/ACCESS.2019.2896880
13. Jamtsho T, Powdyel K, Powrel RK, Bhujel R, Muramatsu K (2021)OCR and speech recognition system using machine learning. In: 2021 innovations in power and advanced computing technologies (i-PACT), Kuala Lumpur, Malaysia, pp 1–5.https://doi.org/10.1109/i-PACT52855.2021.9697030
14. Zrar K. Abdul Abdulbasit KA-T (2022) Mel frequency cepstral coefficient and its applications: a review. IEEE Access 10:122136–122158
15. Passricha V, Aggarwal RK (2020) A Hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. J Intell Syst 29(1):1261–1274. https://doi.org/10.1515/jisys-2018-0372
16. Eray O, Tokat S, Iplikci S (2018)An application of speech recognition with support vector machines. In: 2018 6th international symposium on digital forensic and security (ISDFS), Antalya, Turkey, pp 1–6.https://doi.org/10.1109/ISDFS.2018.8355321
17. Chouhan K, Singh A, Shrivastava A, Agrawal A, Shukla BD, Tomar PS (2021) Structural support vector machine for speech recognition classification with CNN approach. In: 2021 9th international conference on cyber and IT service management (CITSM), Bengkulu, Indonesia, pp 1–7. https://doi.org/10.1109/CITSM52892.2021.9588918
18. Fauzi Z, Sarno R, Hidayati SC (2023) Recognition of real-time Bisindo sign language-to-speech using machine learning methods. In: 2023 international conference on computer science, information technology and engineering (ICCoSITE), Jakarta, Indonesia, pp 986–991. https://doi.org/10.1109/ICCoSITE57641.2023.10127743
19. Eduardo F, Xavier F, Jordi P, Frederic F, Xavier S (2022) FSD50K: an open dataset of human-labeled sound events. arXiv:2010.00475. https://doi.org/10.48550/arXiv.2010.00475

Matched Filter and Kirsch's Template Based Approach for Retinal Vessel Segmentation

Sonali Dash, Kanwarpreet Kaur, and Gaurav Bathla

Abstract The analysis of retinal images plays an instrumental role in diagnosing Diabetic Retinopathy. This progressive disease can cause blindness which can be inhibited with earlier detection. A robust approach is presented in this paper for blood vessel segmentation by integrating a matched filter with Kirsch's template with hysteresis thresholding. The proposed approach involved three steps: preprocessing, blood vessel extraction, and post-processing for vessel extraction. This approach achieved more accurate results than the original Kirsch's template for specificity, sensitivity, and accuracy on the DRIVE dataset.

Keywords Hysteresis thresholding · Kirsch's template · Matched filter · Retinal vessel segmentation

1 Introduction

In the realm of biomedical, automatic analysis of retinal images makes it simple for ophthalmologists to diagnose retina disorders, although conventional procedures like pupil dilation take time [1]. There exist several approaches for the automatic extraction of retinal blood vessels such as Kirsch's template and Matched Filter (MF). Edge image can be considered as a space gradient in Kirsch edge detection. Several suggested methods are available for blood vessel segmentation using Kirsch's template and its modified versions [2–6]. The conventional MF method is a representative one among the different retinal vessel extraction techniques, and it benefits from being simple and efficient [7]. Hoover et al. have recommended locating the blood vessels by using piecewise threshold probing of an MF response [8]. In [9], the response of the MF has been improved by changing parameters. Al-Rawi and

S. Dash (✉) · G. Bathla
CSED, Chandigarh University, Gharuan, Mohali, India
e-mail: sonali.isan@gmail.com

K. Kaur
ECED, Chandigarh University, Gharuan, Mohali, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_5

Karajeh have optimized the MF using a genetic algorithm for vessel segmentation [10]. Sreejini and Govindan have optimized MF using the Particle Swarm Optimization (PSO) method for vessel segmentation [11]. Oliveria et al. have combined three filters like Frangi filter, MF, and Gabor filter to improve retinal vessel segmentation [12]. Zolkifli et al. [13] presented a method based on Kirsch template for extraction of retinal blood vessels but it possesses the limitation of accurately detecting the fine blood vessels due to the presence of noise which is having approximately same size as that of fine blood vessels. In [14], the amalgamation of Fuzzy C-Means with Kirsch template is proposed for detecting the blood vessels in retinal images which obtained better performance than existing ones.

This work aims to improve the response of Kirsch's template in detecting retinal blood vessels. This work suggests an improved Kirsch template by integrating with MF followed by hysteresis thresholding for blood vessel segmentation. Initially, Kirsch's template followed by a thresholding is used for detecting the blood vessels. Afterward, MF integrated with Kirsch's template and hysteresis thresholding is utilized for vessel extraction. The presented technique is tested on the DRIVE dataset. The results obtained after experimentation confirm that the presented approach delivers better performance metrics than the original Kirsch template.

The paper is outlined as follows: Required materials and methods are explained in Sect. 2. Experimental results are discussed in Sect. 3 before providing the conclusions in Sect. 4.

2 Materials and Methods

2.1 *Materials*

Kirsch Templates. The edges detected in Kirsch edge detection are considered space gradients. The Kirsch operator is capable of automatic adjustment of threshold value owing to the characteristics of an image. It uses 8 masks for relevant directions which are applied on the provided image for edge detection. These distinct masks are made by rotating a basic 3 × 3 compass convolution filter as given below.

$$K_1 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \tag{1}$$

$$K_2 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \tag{2}$$

$$K_3 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \tag{3}$$

$$K_4 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \tag{4}$$

$$K_5 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \tag{5}$$

$$K_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \tag{6}$$

$$K_7 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \tag{7}$$

$$K_8 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \tag{8}$$

After the generation of blood vessel images by Kirsch's template matching, morphological closing is essential for closing the holes or vacant areas. The limitation of Kirsch's filter is that it leads to the thickening of blood vessels.

Matched Filter (MF). It is utilized for detecting retinal vessels [7]. It utilizes previous information that the Gaussian function can be used to approximate the cross-section of vessels. Thus, the vessels can be matched by utilizing the Gaussian-shaped filter for the purpose of detection. MF is given as:

$$\begin{gathered} f(x, y) = \frac{1}{\sqrt{2\pi s}}\exp\left(-\frac{x^2}{2s^2}\right) - m \\ \text{for } |x| \le ts,\ |y| \le L/2 \end{gathered} \tag{9}$$

where m is used for normalizing the mean of the filter to zero for the removal of a smooth background after filtering. It is defined as:

$$m = \left(\int_{-ts}^{ts} \frac{1}{\sqrt{2\pi s}}\exp\left(-\frac{x^2}{2s^2}\right)\mathrm{d}x\right)\Big/ (2ts) \tag{10}$$

such that s gives the filter scale. L denotes the length of the neighborhood along the y-axis for noise smoothing; criterion t is constant which is generally considered to be 3 as more than 99% of the area under the Gaussian curve lies in the range of $[-3s, 3s]$ such that L is selected on the basis of s. If s is small, then L is considered relatively small, and vice-versa. For implementation, vessels in different orientations are detected by rotating $f(x, y)$.

2.2 Methods

This paper recommends a vessel extraction method by combining MF and Kirsch templated with hysteresis thresholding. The presented approach for blood vessel extraction in retinal images consists of three phases which include preprocessing followed by segmentation and postprocessing. Due to the higher intensity than red and blue channels, the green channel is selected from the RGB retinal image. The steps for implementation of the proposed method are illustrated in Fig. 1.

The initial stage of preprocessing involves the extraction of the green channel of the color retinal image. Further, blood vessels are extracted using Kirsch's templates. It is one of the first-order discrete derivatives for detection and enhancement. This approach is used for detecting an edge of the blood vessel by using eight directions of template in which rotation is done by 45^0 followed by thresholding and results in an extracted blood vessel.

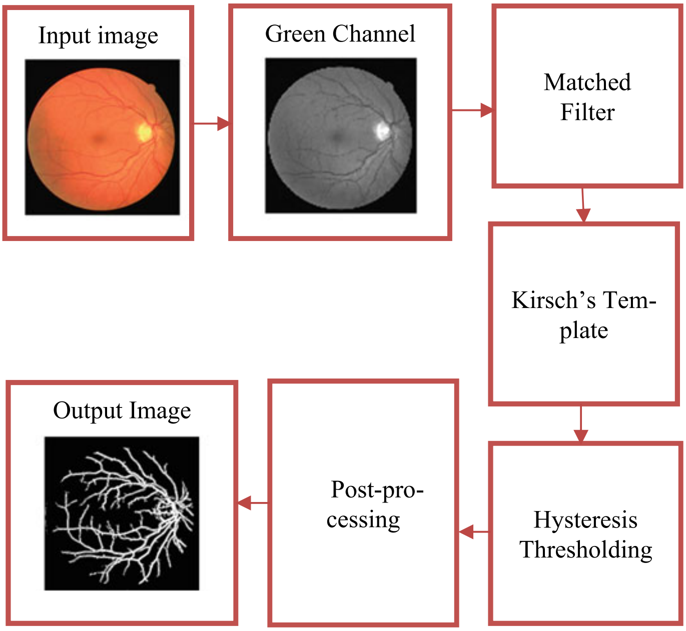


Fig. 1 Block diagram of presented technique

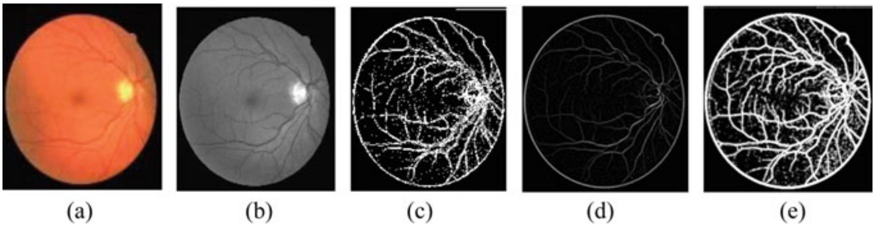


Fig. 2 Images for Rt2 of DRIVE dataset **a** original image; **b** green channel image of (**a**); **c** image obtained from Kirsch's template followed by threshold; **d** image obtained from MF; **e** image obtained from integrated MF and Kirsch template followed by threshold

The next step is to improve Kirsch's template performance by integrating with MF and vessel extraction through hysteresis thresholding. MF takes advantage of a symmetric Gaussian intensity profile at the cross-section of blood vessels. Thus, it is anticipated that when the retinal image is convolved with MF having a Gaussian profile acquires a low response to the background of almost constant intensity and a high response to the vessels. Figure 2a, b show the original as well as the green channel image of Rt2 from the DRIVE dataset while Fig. 2c–e illustrate the image obtained from MF, and an image obtained from an integrated MF and Kirsch template followed by threshold. Finally, hysteresis thresholding is applied on the integrated method of Kirsch template and MF for extracting the accurate blood vessels. The image is segmented using hysteresis thresholding by considering two threshold values T_{low} and T_{high}. Pixels having values less than T_{low} are set to 0 and more than T_{high} are set to 1, pixel values higher than T_{low}, and having at least one neighborhood pixel greater than T_{high} is set 1, others are set 0. Both these threshold values are selected experimentally, and these threshold values will be applied to segment every image.

It is perceived that the image obtained after thresholding consists of some unwanted pixels forming very thin lines and dots that emerged as noise, which can be misclassified as noise. Thus, a morphological opening operation removes these outgrowths in an efficient manner and provides a final segmented image in the postprocessing stage.

3 Experimental Results, Comparison, and Discussions

The presented approach for the enhancement of vessels is formulated by integrating the MF and Kirsch template with hysteresis thresholding for the detection of retinal vessels. DRIVE dataset is utilized for confirming the efficacy of the suggested approach which consists of 40 retinal images having size 768 × 584 pixels with 24 bits and 7 of them have several pathological cases [15]. Images present in the dataset are further classified into training and testing, each consisting of 20 images. The performance of the presented approach is evaluated on the testing images. The comparison of the segmented image of the presented technique is done with the ground truth. Further, the performance is analyzed by considering three different metrics, that is, sensitivity, specificity, and accuracy. Accuracy gives conventionality of segmentation. While Sensitivity shows the capability of the approach to identify the correct vessel pixel and specificity is considered to be the capability of identification of pixels in the background.

$$\text{Sensitivity} = TP/(FN + TP) \tag{11}$$

$$\text{Specificity} = TN/(TN + FP) \tag{12}$$

$$\text{Accuracy} = (TP + TN)/(TP + TN + FN + FP) \tag{13}$$

TP, FP, TN, and FN are defined as:

True Positive (TP) = pixel is correctly recognized as a vessel.
False Positive (FP) = pixel is incorrectly recognized as a vessel.
True Negative (TN) = pixel is correctly recognized as background.
False Negative (FN) = pixel is incorrectly recognized as background.

The evaluation of the presented technique is computed by using several performance metrics such as sensitivity, specificity, and accuracy. Table 1 provides the performance measures of the original Kirsch template.

Traditional Kirsch's template delivers sensitivity, accuracy, and specificity as 0.5507, 0.9089, and 0.9432 on the DRIVE dataset. Table 2 provides the performance metrics obtained for each image by applying an integrated MF and Kirsch's template followed by hysteresis thresholding.

The segmented images of the Rt2 and Rt4 from the DRIVE dataset have achieved better segmented result as represented in Fig. 3. From Table 2, it is perceived that the presented approach achieves higher values for accuracy, specificity, and sensitivity as compared to the traditional Kirsch template. The integrated approach achieved sensitivity, accuracy, and specificity as 0.6440, 0.9241, and 0.9528 on the DRIVE dataset. Table 3 provides the comparative analysis of proposed approach with the existing techniques.

It is evident from the above table that the proposed approach achieves better accuracy when compared to Zolkifli et al. [13] but almost similar accuracy is achieved by Mustafa et al. [14]. Thus, it is ascertained that the retinal vessel extraction can be achieved by utilizing the proposed approach.

4 Conclusions

This work is focused on improving the response of Kirsch's template for retinal blood vessel detection by integrating with MF. Hysteresis thresholding is used for vessel segmentation. The purpose of the recommended works is to increase the performance

Table 1 Evaluated performance measures on the original Kirsch's template followed by hysteresis thresholding

Image	Sensitivity	Specificity	Accuracy
Rt1	0.56926	0.93471	0.902103
Rt2	0.662622	0.927761	0.900609
Rt3	0.461588	0.961093	0.911298
Rt4	0.637675	0.939063	0.911338
Rt5	0.534711	0.963203	0.92306
Rt6	0.49287	0.961705	0.916072
Rt7	0.593659	0.938437	0.906931
Rt8	0.469548	0.949786	0.908468
Rt9	0.505852	0.967608	0.930185
Rt10	0.517712	0.954264	0.918336
Rt11	0.65835	0.915432	0.892417
Rt12	0.544753	0.940007	0.90588
Rt13	0.537803	0.946839	0.906849
Rt14	0.57469	0.936663	0.907398
Rt15	0.657322	0.920263	0.901446
Rt16	0.586049	0.941286	0.909213
Rt17	0.50359	0.946658	0.909259
Rt18	0.500535	0.929895	0.895875
Rt19	0.539659	0.937542	0.904537
Rt20	0.466309	0.953038	0.917245
Average	0.55072	0.94326	0.90892

Note Rt represents Retina

of Kirsch's template. The suggested technique is evaluated by using the DRIVE dataset, and the results are compared with the original Kirsch's template method. The presented integrated method achieves sensitivity, accuracy, and specificity of 0.6440, 0.9241, and 0.9528. It is observed that the recommended approach outperforms the traditional method in terms of metrics such as specificity, sensitivity, and accuracy on the DRIVE dataset.

Table 2 Evaluated performance measures on integrated MF and Kirsch's template followed by hysteresis thresholding

Image	Sensitivity	Specificity	Accuracy
Rt1	0.604925	0.949165	0.918451
Rt2	0.678159	0.953942	0.9257
Rt3	0.592193	0.968046	0.925176
Rt4	0.607992	0.972584	0.939044
Rt5	0.667249	0.957445	0.930258
Rt6	0.637159	0.958519	0.92724
Rt7	0.649907	0.951319	0.923776
Rt8	0.573356	0.953142	0.920466
Rt9	0.695711	0.946471	0.946149
Rt10	0.687104	0.952085	0.930276
Rt11	0.585125	0.95985	0.926303
Rt12	0.747034	0.956135	0.910671
Rt13	0.63697	0.952049	0.921245
Rt14	0.68351	0.94063	0.919842
Rt15	0.695138	0.959911	0.922394
Rt16	0.611292	0.949738	0.919181
Rt17	0.58836	0.951272	0.920639
Rt18	0.603351	0.946998	0.919769
Rt19	0.69201	0.933834	0.913774
Rt20	0.645333	0.944085	0.922115
Average	0.644093	0.95286	0.92412

Note Rt represents Retina

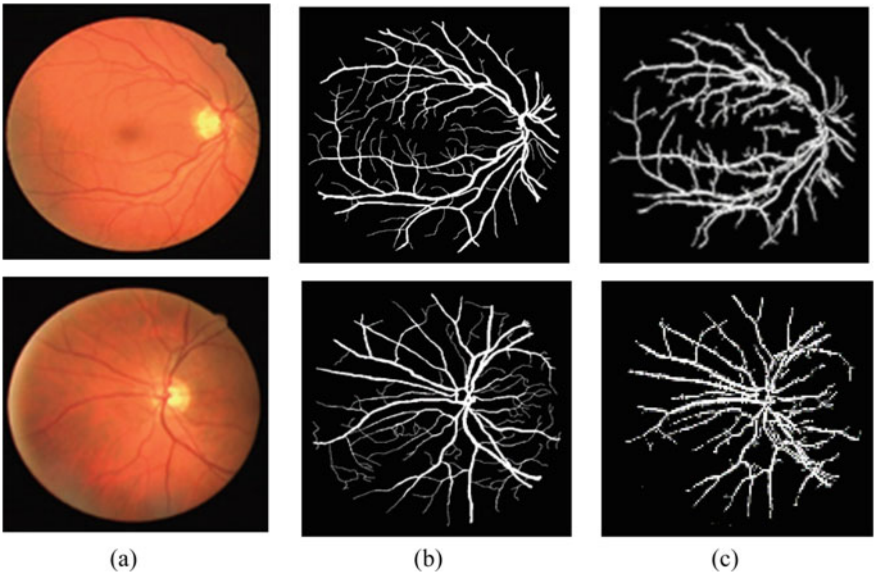


Fig. 3 Segmented images achieved by the suggested technique for Rt2 and Rt4 from DRIVE dataset **a** original image; **b** Ground truth for an image in (**a**); **c** segmented image obtained by MF integrated with Kirsch template followed by hysteresis thresholding

Table 3 Comparative analysis of proposed approach with existing ones

Sl. No.	Method	Accuracy (%)
1	Zolkifli et al. [13]	75.97
2	Mustafa et al. [14]	92.64
3	Proposed	92.41

References

1. Sopharak A, Uyyanonvara B, Barman S, Williamson TH (2008) Automatic detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods. Comput Med Imaging Graph 32(8):720–727
2. Gao P, Sun X, Wang W (2010) Moving object detection based on kirsch operator combined with optical flow. In: 2010 international conference on image analysis signal processing, Zhejiang, pp 620–624
3. Hussain A, Nazir S, Khan F, Nkenyereye L, Ullah A, Khan S, Verma S (2023) A resource efficient hybrid proxy mobile IPv6 extension for next generation IoT networks. IEEE Int Things J 10(3):2095–2103
4. Maitra IK, Nag S, Bandyopadhyay SK (2012) A novel edge detection algorithm for digital mammogram. Int J Inform Commun Technol Res 2(2):207–215
5. Majumdar J, Kundu D, Tewary S, Ghosh S, Chakraborty S, Gupta S (2015) An automated graphical user interface based system for the extraction of retinal blood vessels using kirsch's template. Int J Adv Comput Sci Appl 6(6):86–93

6. Li W, Chai Y, Khan F, Jan SR, Verma S, Menon VG, Kavita F, Li X (2021) A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system. Mobile Netw Appl 26:234–252
7. Chaudhuri S, Chatterjee S, Katz N, Nelson M, Goldbaum M (1989) Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans Med Imaging 8(3):263–269
8. Hoover AD, Kouznetsova V, Goldbaum M (2000) Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Trans Med Imaging 19(3):203–210
9. Al-Rawi M, Qutaishat M, Arrar M (2007) An improved matched filter for blood vessel detection of digital retinal images. Comput Biol Med 37(2):262–267
10. Singh AP, Pradhan NR, Luhach AK, Agnihotri S, Jhanjhi NZ, Verma S, Ghosh U, Roy DS (2020) A novel patient-centric architectural framework for blockchain-enabled healthcare applications. IEEE Trans Industr Inf 17(8):5779–5789
11. Gandam A, Sidhu JS, Verma S, Jhanjhi NZ, Nayyar A, Abouhawwash M, Nam Y (2021) An efficient post-processing adaptive filtering technique to rectifying the flickering effects. PLoS ONE 16(5):e0250959
12. Oliveira WS, Teixeira JV, Ren TI, Cavalcanti GD, Sijbers J (2016) Unsupervised retinal vessel segmentation using combined filters. PLoS ONE 11(2):e0149943
13. Zolkifli NS, Nazari A, Mustafa MM, Zakaria WNW, Suriani NS, Kairuddin WNHW (2020) Retina blood vessel extraction based on Kirsch's template method. Indonesian J Electric Eng Comput Sci (IJEECS) 18(1):318–325
14. Mustafa WA, Mahmud AS, Khairunizam W, Razlan ZM, Shahriman AB, Zunaidi I (2019) Blood vessel extraction using combination OF Kirsch's templates and fuzzy C-means (FCM) on Retinal Images. In: IOP conference series: materials science and engineering, vol 557, no 1, Indonesia, Art. No 012009
15. DRIVE: digital retinal images for vessel extraction [Online]. Retrieved from https://drive.grand-challenge.org/DRIVE/. Accessed on 12 June 2023

Prediction of Abnormality in Kidney Function Using Classification Techniques and Fuzzy Systems

Mynapati Lakshmi Prasudha, Sukhavasi Vidyullatha, and Yeluri Divya

Abstract Kidney diseases are life threatening. Its development is prevented by early detection and vigorous management. It is important to discover such disorders at an early stage in order to extend a patient's lifespan and to classify the abnormalities in kidney function based on pathological data. The primary goal is to identify the stages of the kidney disease and check the performance for various classifiers of the model. In this paper, classification algorithms are used to find out the accuracy of the supervised data. Not all machine learning classifiers predict the accurate results because of imprecision. So, fuzzy expert system (FES) is used to deal with imprecise data. To predict the disease at an early stage and also to identify the stages of the disease, FES is used. FES has shown promising results in identifying the stages of the patients. The accuracy of the pathological data is found by using machine learning algorithms. In addition, the probability of the occurrence of the disease is found by combining various parameters and identified the stages of the patient's disease.

Keywords Fuzzy expert system (FES) · Logistic regression (LR) · Support vector machine (SVM) · Decision tree (DT) · Random forest (RF) · Chronic kidney disease (CKD) · Acute kidney injury (AKI)

1 Introduction

Subsequent paragraphs, however, are indented. Kidneys are bean shaped organs in the human body located at the backside. Healthy kidneys are about 5 inches in size. The change in kidney size indicates an unhealthy kidney condition. Kidneys purify about 200 L of blood per day. The major function of the kidneys is to filter excess water, salts, and waste from the blood. The appropriate operation of this entire process is

M. Lakshmi Prasudha · S. Vidyullatha (✉) · Y. Divya
BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad, Telangana, India
e-mail: vidyullatha.1988@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_6

required to maintain electrolytes in a healthy level. Figures 1 and 2 show the healthy and unhealthy kidneys.

Ailments related to kidney are becoming more prevalent. Kidney damage happens slowly among many people over many years, generally as a result of diabetes mellitus or blood pressure, and it can be termed as CKD, whereas AKI happens when a person's renal function changes suddenly due to illness, accident, or by the use of

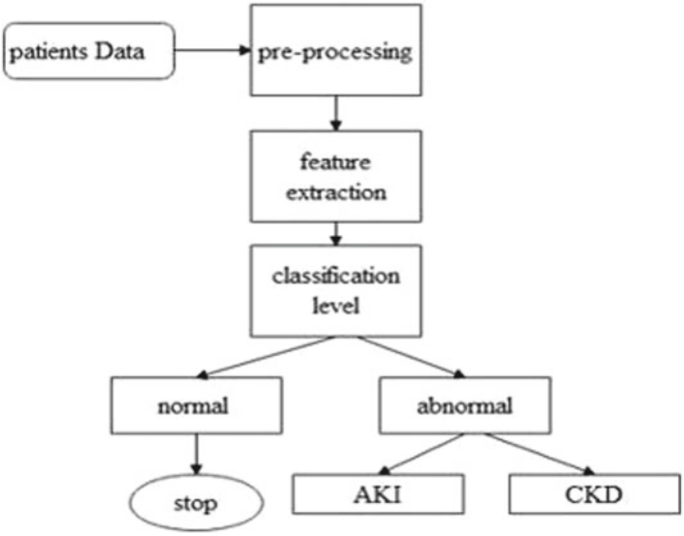


Fig. 1 Proposed model for kidney disease prediction [29]

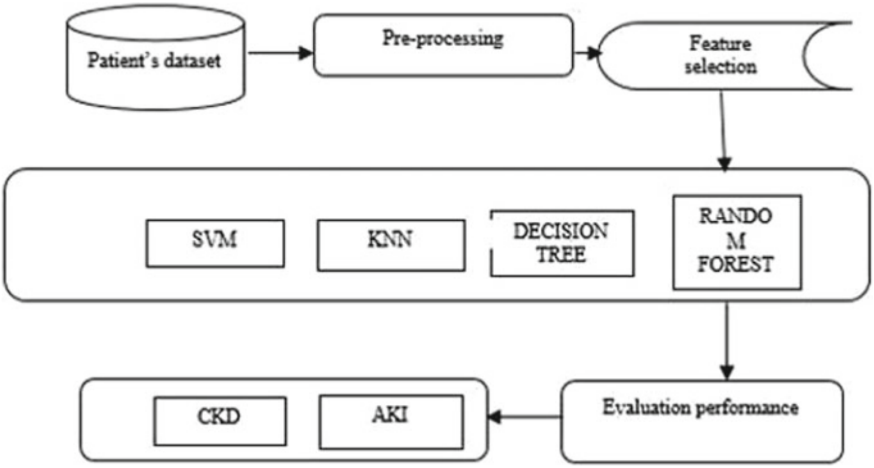


Fig. 2 Proposed framework for CKD and AKI prediction

certain drugs. This can effect the healthy people whose healthy kidneys or problems have related to kidneys. Chronic kidney disease (CKD) is usually dangerous condition if not identified at an early stage. Its progression is prevented by early detection and effective management [1–6]. It is vital to discover such disorders at an early stage in order to extend a patient's lifespan. Kidney disease is a quiet and serious disease that affects people all over the world. It is harmful since the symptoms do not appear until the kidney's functions have deteriorated by 85–90%. According to Global Burden of Diseases (GBDs), over 1.2 million individuals died from kidney disease in some form. Since 2005, the proportion has raised by 32%, implying that the death rate of renal patients has increased by 32% over a ten-year period. According to the findings of the study, around 5–10 million individuals die each year as a result of kidney failure.

2 Literature Survey

To predict renal disorders, SVM and ANN were used [7, 8]. The study examined the accuracy and execution time of the two methods mentioned above. To develop a set of features that can predict kidney damage, effectively feature selection algorithms are employed. The reduced feature set reduces costs, improves efficiency, and eliminates ambiguity [9]. To predict at an early stage, the combination of machine learning algorithms and predictive modeling is proposed [10]. ANN models were assessed for predicting patient's lifespan, especially while suffering with CKD [11, 12]. K-means algorithm was used to extract information about how CKD markers interact with patient's mortality and analyzed clustering methods to predict dialysis patient's lifespan. By using Hadoop environment, different machine learning algorithms are used, and KNN and SVM with an AUROC 0.83 is achieved [13]. Gradient boosting algorithms and clinical information from EHR to present a one-year prediction model for CKD [14] among diabetic patients [15]. Convolutional auto-encoder is used to encode the temporal features, which exceeded baseline models by using EHR data containing sequences of lab test results to predict the risk of progressing from the first to the second stage of diabetic nephropathy [16]. The prediction model in kidney disease patients is proposed, especially for hypertension individuals using textual and numeric data from EHR. A neural network, based on bidirectional long- and short-term memories and auto-encoders, were used to encode both textual, numerical data. Under-sampling is used to balance the data and is able to get an accuracy of 89.7% using tenfold cross-validation [17]. Dataset containing missing values are dealt since it results in reduction of the model's accuracy and prediction outputs. They discovered a solution to this by performing a recompilation process on CKD stages, which resulted in unknown values. They recalculated missing data to fill up the gaps [18]. Using several machine learning classifications techniques, the authors worked on reducing diagnosis time and increased accuracy for the same. The classification of different stages of CKD based on severity is proposed. Using algorithms such as the RBF, RF, and BPNN, the results shown that RBF algorithm performs better than other classifiers, with an accuracy of 85.3% [19, 20].

3 Proposed Work

3.1 *Kidney Disease Classification Using Machine Learning Based on Pathological Data*

Analyzing the medical data is a very sensible matter and that must be done correctly for disease prediction, detection, and analysis. This results in developing accurate tools and usage of such effective machine learning algorithms [21] which accurately detect or for diagnosing the disease. Appropriate and effective analysis of medical data has ushered in a revolution in machine learning field, especially for the widespread usage computationally demanding algorithms in recent years. However, existing number of clinical issues, namely accuracy, dependability, and rapid decision models, must be solved in order to guide physicians while diagnosing disease effectively [22, 23]. The classifiers' performance for disease prediction is determined by the obtained quality medical data and the classifier models used for the classification process. As a result, it is critical to employ various classifiers to correctly and accurately assess sensitive medical data in order to anticipate and detect diseases. In machine learning, classification [19, 20, 24–28] is a crucial challenge to extract knowledge from various real-time issues. Hence, a well-developed model shown in Fig. 1 is required to accurately predict the target class using collected data at multiple categorization levels.

3.2 *Early-Stage CKD Prediction Using Fuzzy Systems*

The major goal of developing this fuzzy expert system [30] is to assist doctors in detecting CKD in patients. This medical expert system can detect a disease and assist specialists in providing proper and appropriate treatment. Here, a patient's data is taken as input and classifying the stage of the patients is expected output. Inference and defuzzification are used to process the output as shown in Fig. 2 [7, 31–33]. The fuzzy system contains various methods like fuzzification, inference, and defuzzification for processing the output. A fuzzy expert system [34] is a conceptual framework used to diagnose and also to manage chronic kidney disease. The rule-based model receives its membership value through the defuzzification technique. This method converts output (linguistic values) into crisp values [35, 36].

4 Methodology

4.1 Dataset

We used a publicly available chronic kidney disease dataset from UCI repository. It contains 400 instances and 25 attributes.

4.2 ML Classifiers

Algorithm for proposed model

Input. Patient's dataset

Output. Correct classification of patient's dataset under various classification algorithms.

Step 1. Load dataset.

Step 2. Pre-processing the data.

- Row-elimination technique to deal with missing values.
- Convert the categorical values into numerical values

Step 3. Construct the classifier model (LR, DT, RF, and SVM) for preprocessed dataset.

Step 4. Performance analysis of constructed classifier models in step 3.

4.3 Classification Accuracy

Equation (1) is used to calculate the accuracy of given models:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

where TP, TN, FP, and FN are observations and prediction values which are given in terms of true, false, positive, and negative.

4.4 Fuzzification

Medical diagnosis frequently requires a thorough examination of a patient in order to determine whether the patient is suffering from a suspected condition. If we consider sugar level, it may be high for one patient and low for another patient or no sugar for others. So here are combined features and its strengths to obtain an accurate diagnostic conclusion. Here, physicians' experience is used in the current study to create a database of various fuzzy rules. Based on fuzzy decisions [37], a computer software can be developed to automatically evaluate if a patient with specific symptoms is suffering from one or other kind of a diseases.

The profile table can be determined as [r pij, rij, v].

$$\sigma i = \sum_{j=1}^{j-ki} (Wij\delta ij) \Bigg/ \sum_{j=1}^{j=ki} Wij\cdot \tag{2}$$

Equation (2) is used to take a diagnosis decision by adding the impact of Ki relevant features by adding weighing factor wij. In this case, all the features will have equal weighted factor.

$$\sigma i = \frac{1}{ki} \sum_{j=1}^{j=ki} \delta ij. \tag{3}$$

Equation (3) is to obtain precise crisp numbers which indicates the probability of each disease in the set *S*.

For the given data, the first step is to perform normality test. The main risk factors for CKD are SCR, blood sugar, blood pressure, age, GFR, and smoke. Here, normality tests are performed for GFR and SCR because these are main factors for CKD prediction.

Normality. Normality check is very important while considering any pathological or numerical data because the obtained data contains lots of imprecision. To deal with imprecise data, normality check is must.

Confidence Indicator. CIs can be used to determine the ranges that will function as the fuzzy [38] sets in the outputs and input variables of a given model.

After normality test is done, to measure uncertainty in variables, a confidence indicator is used. Equation (4) for confidence interval is:

$$CI = \overline{X} \pm \mathrm{Z}\frac{s}{\sqrt{n}}. \tag{4}$$

CI = confidence interval.
Z = confidence level value.
s = sample standard deviation.
n = sample size.

IF–THEN-RULES (knowledge base). The fuzzy variables for output categorization are linked with set of rules in this step. Mandani fuzzy rule-based model is used to store fuzzy rules [39]. Different membership functions are selected and analyzed for certain results, such as parameter as normal, moderate, or critical, using the MATLAB-FIS editor. Finally, the condition of a patient is established using the prepared rule bases, taking into account the status of individual parameters [40]. GFR = low (0–15), moderate (15–60), high (60 and above).

From the obtained data, to classify the stages of abnormality in kidney disease, six variables are considered. From these six variables, we get 324 rules.

Confidence indicator is used to estimate the performance using Eq. (5).

$$CI = \frac{\text{Success number}}{\text{total tests}} * 100. \tag{5}$$

5 Result Analysis

While calculating CI, we got 92% accuracy when fuzzy expert system is used. So, we can say that fuzzy system [41] can help many physicians for diagnosing CKD. Various metrics such as accuracy, precision, specificity, and sensitivity can be used for performance evaluation. In order to evaluate the performance metrics, Table 1 and Table 2 show the performance of classifiers and the confusion matrix must be reduced to 2 × 2 matrix and is shown in Table 3.

Table 1 Performance of various classifiers

Algorithms	Total predictions	Right predictions	Wrong predictions	Accuracy	precision	Recall	F1-score	Support
LR	210	193	17	94.56	0.98	0.91	0.95	162
SVM	210	168	42	88.52	0.79	1.00	0.89	162
DT	210	205	5	98.45	0.99	0.98	0.98	162
RF	210	206	4	98.75	0.99	0.98	0.99	162

Table 2 Confusion matrix for various classifiers

Classifier	TP	FP	FN	TN
LR	148	14	3	45
SVM	162	0	42	6
DT	159	3	2	46
RF	159	3	1	47

Table 3 2 × 2 matrix

Yes	No	Class
136	15	Yes
11	162	No

6 Conclusion

To predict the kidney disease at an early stage, the given input data was first classified into two levels, i.e., AKI and CKD using binary classification. A model was built using LR, SVM, DT, and RF. The random forest performed better while comparing to other classifiers, with an accuracy of 98.7%. In order to identify various stages of the disease, the attributes are combined to predict the outcome. By using fuzzy system, the inference rules were built and trained a model on these rules and achieved 96% accuracy when 200 rules were used for training using fuzzy expert system. So, it can be concluded that this can be helpful for many physicians in taking decisions to predict the severity of the disease at an early stage.

References

1. Lakshmi Prasudha M, Kasumolla R, Sukheja D (2021) Research reviews: towards identification and classification kidney disease using computational technology. In: 2021 5th international conference on computing methodologies and communication (ICCMC), pp 1387–1391. https://doi.org/10.1109/ICCMC51019.2021.9418454
2. Prasudha ML et al (2021) Comprehensive analysis of state-of-the-art CAD tools and techniques for chronic kidney disease (CKD). IJBDAH 6(2):1–12. https://doi.org/10.4018/IJBDAH.287605
3. Sivasankar E, Pradeep R, Sinandham S (2019) Identification of important biomarkers for detection of chronic kidney disease using feature selection and classification algorithms. Int J Med Eng Inform 11(4)
4. Aditya K, Babita P (2020) "A novel integrated principal component analysis and support vector machines-based diagnostic system for detection of chronic kidney disease. Int J Data Anal Tech Strateg (IJDATS) 12(2)
5. Pramila A, Eswaran P (2021) An efficient oppositional crow search optimization-based deep neural network classifier for chronic kidney disease identification. Int J Innov Comput Appl (IJICA) 12(4)
6. Khaled MA (2021) Prediction of chronic kidney disease using different classification algorithms. Inform Med Unlock 24:100631, ISSN2352-9148. https://doi.org/10.1016/j.imu.2021.100631
7. Fazel Zarandi MH, Abdolkarimzadeh M (2022) Fuzzy rule based expert system to diagnose chronic kidney disease. In: Springer NAFIPS 2017 annual conference, vol 648, pp 323–328
8. Panwong P, Iam-On N (2021) Predicting transitional interval of kidney disease stages 3 to 5 using data mining method. In: 2016 second Asian conference on defence technology (ACDT), Chiang Mai, pp 145–150
9. Vijayarani S, Dhayanand S (2021) Kidney disease prediction using SVM and ANN algorithms. Int J Comput Bus Res (IJCBR) 6(2)
10. Aljaaf J et al (2022)Early prediction of chronic kidney disease using machine learning supported by predictive analytics. In: 2018 IEEE congress on evolutionary computation (CEC), Rio de Janeiro, pp 1–9
11. Zhang H, Hung C, Chu WC, Chiu P, Tang CY (2021) Chronic kidney disease survival prediction with artificial neural networks. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM), Madrid, Spain, pp 1351–1356
12. Tazin N, Sabab SA, Chowdhury MT (2022) Diagnosis of Chronic Kidney disease using effective classification and feature selection technique. In: 2016 international conference on medical engineering, health informatics and technology (MediTec), Dhaka, pp 1–6

13. Kaur G, Sharma A (2022) Predict chronic kidney disease using data mining algorithms in Hadoop. In: 2017 international conference on inventive computing and informatics (ICICI), Coimbatore, pp 973–979
14. Al-Hayari AYA, Al-Taee AM, Al-Taee MA (2021) Clinical decision supprot system for diagnosis and management of chronic renal failure. In: IEEE Jordan conference on applied electrical engineering and computing technologies, pp 1–6
15. Johansson M, Buijs JOD, Song X, Waitman LR, Yu AS, Robbins DC, Hu Y, Liu M (2020) Longitudinal risk prediction of chronic kidney disease in diabetic patients using a temporal-enhanced gradient boosting machine: retrospective cohort study. JMIR Med Inform 8:e15510 [CrossRef]
16. Katsuki T, Ono M, Koseki A, Kudo M, Haida K, Kuroda J, Makino M, Yanagiya R, Suzuki A (2018) Risk prediction of diabetic nephropathy via interpretable feature extraction from EHR using convolutional autoencoder. Stud Health Technol Inform 247:106–110
17. Ren Y, Fei H, Liang X, Ji D, Cheng M (2021) A hybrid neural network model for predicting kidney disease in hypertension patients based on electronic health records. BMC Med Inform Decis Mak 19:131–138. [CrossRef] [PubMed]
18. Dilli Arasu S, Thirumalaiselvi R (2021) Review of chronic kidney disease based on data mining techniques. Int J Appl Eng Res ISSN 0973–4562 12(23):13498–13505
19. Ramya S, Radha N (2022) Diagnosis of chronic kidney disease using machine learning algorithms. Proc Int J Innov Res Comput Commun Eng 4(1)
20. Polat H, Mehr HD, Cetin A (2021) Diagnosis of chronic kidney disease based on support vector machine by feature selection method. Springer 41(4):1–11
21. Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine learning. Neural Statistic Class 12(12)
22. Sebasky M, Kukla A, Leister E, Guo H, Akkina SK, El-Shahawy Y, Matas AJ, Ibrahim HN (2009) Appraisal of GFR-estimating equations following kidney donation. Am J Kidney Dis 53(6):1050–1058
23. Jahantigh FF (2015) Kidney diseases diagnosis by using fuzzy logic
24. Sahani R, Rout C, Badajena JC, Jena AK, Das H (2021) Classification of intrusion detection using data mining techniques. In: Progress in computing, analytics and networking, Springer, Singapore, pp 753–764 CrossRefView Record in Scopus Google Scholar[4]
25. Das H, Naik H, Behera HS (2020) Classification of diabetes mellitus disease (DMD): a data mining (DM) approach Progress in computing, analytics and networking, Springer, Singapore, pp 539–549 CrossRefScopus Google Scholar
26. Dey N, Ashour A (2016) Classification and clustering in biomedical signal processing, IGI global Hershey, Google Scholar
27. Kamparia A, Saini G, Pandey B, Tiwari S, Gupta D, Kahnna A (2021) KDSAE: Chronic kid ney classification with multimedia data learning using deep stacked autoecnoder network. Springer, pp 1–6
28. Hua C, Wu R, Kei C, An Wang S (2019) A cloud based fuzzy expert system for the risk assessment of chronic kidney disease. Indrescience 9(4)
29. Fig 1 and fig 2 (google images)
30. Ahmed S, Tanzir Kabir M, Mehmood NT, Rehman RM (2022) Diagnosis of kidney disease using fuzzy expert system. In: IEEE The 8th international conference on software, Dhaka, pp 1–8, April, 2022.
31. Ramesh R (2022) Chronic kidney disease prediction using machine learning models. 9:6364. https://doi.org/10.35940/ijeat.A2213.109119
32. Al-Hayari AYA, Al-Taee AA, Al-Taee MA (2018) Clinical decision support system for diagnosis and management of chronic renal failure. In: IEEE Jordan conference on applied electrical engineering and computing technologies, pp 1–6
33. Ahmed S, Tanzir Kabir M, Mehmood NT, Rehman RM (2021) Diagnosis of kidney disease using fuzzy expert system. In: IEEE the 8th international conference on software, Dhaka, pp 1–8
34. Fazel Zarandi MH, Abdolkarimzadeh M (2021) Fuzzy rule based expert system to diagnose chronic kidney diseas. In: Springer NAFIPS 2017 annual conference, vol 648, pp 323–328, September, 2021.

35. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353 Article Download PDF Scopus Google
36. Himansu D, Bighnaraj NHS, Behera C (2020) Medical disease analysis using neuro-fuzzy with feature extraction model for classification. Inf Med Unlocked 18(1–12):100288. Inform Med Unlock 18:100299
37. Hua C, Kei Chiu R, An Wang S (2015) A cloud based fuzzy expert system for the risk assessment of chronic kidney disease. Indrescience 9(4)
38. Clinical decision support system to predict chronic kidney disease: a fuzzy expert system approach https://doi.org/10.1016/j.ijmedinf.2020.104134
39. Norouzi J, Yadollahpour A, Mirbagheri SA, Mazdeh MM, Hosseini SA (2022) Predicting renal failure progression in chronic kidney disease integrated fuzzy expert system. Hindawi Comput Mathematic Methods Med 2016:1–9
40. Shubhajit RC, Dipankar C, Hiranmay S (2008) Development of an FPGA based smart diagnostic system for spirometric data processing applications. Int J Smart Sens Intell Syst 1(4)
41. Zarandi MF, Abdolkarimzadeh M (2017) Fuzzy rule based expert system to diagnose chronic kidney disease North American fuzzy information processing society annual conference, Springer, pp 323–328

Implementation of Parallel Applications on the Hypercube Topology by Using Multistage Network

Qusay S. Alsaffar and Leila Ben Ayed

Abstract The recent computational systems include multicomputer formations. Multiple computers allow many tasks to be processed faster and concurrently and the possibility of implementing the same functions on various processors simultaneously. The problem is that companies are required to pay thousands of dollars to construct data centers, locations, servers, technicians, and hardware maintenance costs and over time need to update and upgrade. This paper simulates a virtual machine and uses the hypercube topology to implement a Multistage Network, a single computer is divided into eight computers as clients and eight computers as servers, the interconnection between servers is represented by mesh topology, and parallel processing is represented by multiplying two-dimensional matrices A and B. Cloud computing should virtualize the management of resources (such as memories, CPUs, storage) to users at reasonable costs. The goal of using this method is to construct a model that not only works on hypercube topology but also works on various topologies. The results showed that the outputs are computed through parallel virtual servers by using the threading technique. We produced a sample approaching a cloud computing system.

Keywords Parallel processing · Multistage hypercube topology · Mesh topology · Java · Threading · Socket server · Cloud system

Q. S. Alsaffar (✉)
University of Sfax, National School of Electronics and Telecommunications of Sfax, Sakiet Ezzit, Tunisia
e-mail: qusay_saffar@mohesr.gov.iq

L. B. Ayed
National School of Computer Science, Sakiet Ezzit, Tunisia
e-mail: leila.benayed@ensi-uma.tn

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_7

1 Introduction

The efficiency of the multiprocessor system can be raised by using Multistage Interconnection Networks (MINs). MINs are a vital element of these systems that allow communication between memory and many processors and between the processors themselves. Efficiency and fast with reasonable communication costs are obtained by providing MINs [1].

Parallel processing consists of a collection of processors running at the same time and sharing the same memory. The parallel system is tightly interconnected, meaning that the processors have access to the clock and memory shared in the system and are under the control of the operating system. Parallel systems are used by parallel computing to manipulate computational issues [2].

Hypercube structures execute an operation that depends on inclusive communication through T tasks, and these T tasks are implemented by using log T steps. There are a variety of parallel algorithms, such as (vector reduction, sorting, barrier synchronization, and matrix multiplication). These parallel algorithms use inclusive communication so that hypercube structures have a high benefit in parallel computations [3]. The parallel prefix [4], machine learning [5], linear algebra computations [6], and Boolean algebraic [7] are considered examples of hypercube structures in parallel processing. Designing data centers, hardware, software, provision of technicians, and cooling systems are expensive, and over time, this equipment needs to be upgraded and become outdated.

In this paper, we propose a new solution for the design of a model that simulates a network using Java and some techniques such as (threading, socket, socket server, and port number). This network model divides a single computer into eight clients and eight servers to execute an application (multiplication of matrices A and B), we make a connection between the client and servers, then the application is divided into threads, and these threads are sent from the selected client to eight servers; then, the interconnection between servers computes the results via hypercube structures, and communication is done using the TCP protocol.

This model provides interconnection between servers through a Multistage Network, the parallel computers work synchronously through mesh topology, and it would be considered as a kernel of various topologies and a small sample of the cloud computing system.

The paper is structured as follows: Sect. 2 provides a short overview of the relevant works. Section 3 defines parallel processing. Section 4 presents Multistage Interconnection Networks. The hypercube topology is described in Sect. 5. Section 6 explains mesh topology. Section 7 introduces the cloud system. Section 8 explains the proposed system. Section 9 discusses the experimental results. The last two sections are conclusions and recommendations.

2 Related Works

Distributed system performance (CORBA, RMI, and Socket) was evaluated by R. Eggen and M. Eggen. There are 2, 4, and 8 nodes that were used to implement parallel sorting application, and these nodes have analyzed the performance of CORBA, RMI, and Socket. The results of RMI and Socket have approximately the same performance. However, the result of CORBA showed that the performance was lower than RMI and Socket [8]. [The paper is only a comparison of some of the tools used in a distributed system].

H. Inoue, T. Moriyama, H. Komatsu, and T. Nakatani use multicores' SIMD processors to apply their parallel sorting algorithm named Aligned Access Sort. The characteristics of the third level of parallelism and SIMS instructions are used by the parallel sort algorithm. The out-core and in-core sorting algorithms are two methods included in the system. Data categorized as unordered is divided into blocks of appropriate size with memory. Every block was arranged with an in-core algorithm, and the out-core algorithm is merged with the sorted blocks to arrange blocks in parallel by using the even–odd merge sort [9]. [This work describes the application on the parallel system and does not introduce the type of topology used].

Merge sort, quick sort, and bubble sort algorithms were compared with their performance by El-Nashar [10], and these algorithms were executed using a message-passing interface on a Windows platform with dual core. The comparison is based on the number of operations working in parallel with the number of cores being used. When the number of processes is increased, the execution time of bubble sort is significantly reduced; although the number of physical cores is less than the number of processes, the execution time of quick sort and merge sort is increased because the number of cores is less than the number of processes. Because of the communication processes, quick sort and merge sort have a bigger communication payload than bubble sort which has a less communication payload. [The paper compares the execution time, the number of processes, and the number of physical cores for executing three applications. [The paper does not provide solutions to a specific problem].

Saleem et al. [11], Intel Cilk Plus article was used to develop two programs. C/C++ language is used to implement two sequential sorting algorithms by transforming them into multicore programs in the Intel Cilk Plus structure to obtain better performance and parallel processing. After the Cilk Plus transforms program, the CILK tools are used to inspect for the different events, and then, the sequential program with a single core compares performance and speed achieved. [Java programming is a more suitable language with a multicore system, and it is not used in this paper].

Damrudi and Aval [12] use hypercube topology to propose a parallel search algorithm. The number of nodes and data elements are equal and every node possesses only one data element in the hypercube. The comparison process is implemented by the node between the key value and the data, and the output of the comparison is transmitted to the adjacent node. Better performance was gained when applying this algorithm. [Also, this research is applied to MATLAB environment and focuses on getting better performance without implementing threading in Java language].

LAN Chat System was proposed by Abba et al. [13], and this system provides for saving conversation, voice call, and sending files. The Java socket was used to achieve the system. The system used TCP protocol to provide unlimited message transmission in the network. Data is delivered reliably when using the TCP protocol. [The proposed method is similar to social media that is used today to communicate between people].

The bubble sort algorithm was studied by Panigrahi et al. [14], in serial and parallel techniques. They selected eight elements from the data in descending order. There are two, four, and eight processors that were used to perform and compare sequential bubble sort and parallel bubble sort. They concluded that using eight processors for bubble sorting (the number of elements and processors being equal) gives better performance and lower execution time, while processors 1, 2, 4 give more execution time and more usage of processors. [The research does not address the structure or topology used].

Rinku and Asha Rani [15], two matrices of size 1000 × 1000 are divided into threads to perform multiplication. The process is executed on a single processor and multiprocessor, and this formula shows the benefit of using multithreading on multiprocessors. Execution time is reduced by 50% in multiprocessing compared to a single processor. [The research does not use virtualization to apply on physical multicores (additional cost) and they did not use Java which is convenient in distributed systems].

3 Parallel Processing

Parallel computers are a group of processing machines that work synchronously and fast to provide solutions to big problems, Fig. 1 [16] illustrates parallel processing, and a parallel system implies that different computers or processors are interconnected to enhance activity [17].

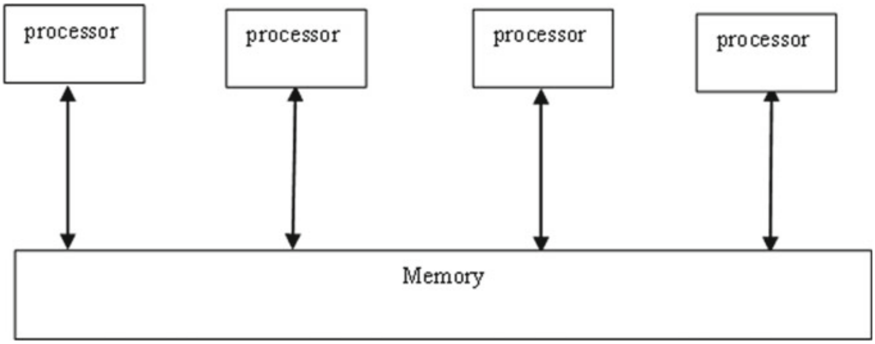


Fig. 1 Parallel system

In this paper, the elements of two matrices were distributed over eight interconnected processors and shared memory to compute the results using Java threading technology.

4 Multistage Interconnection Networks (MINs)

MINs mean that the network architecture includes more than one stage; these stages are small elements interconnected together by links [16].

In this paper, two-stage interconnection networks were designed to execute the matrices multiplication, where the elements of two matrices are sent to eight processors as threads.

4.1 Connection of Source and Destinations

This mechanism simulates the transmission between the source and destination. The switching is used to implement the mechanism in which the connection between the source and destination nodes, the mechanism is regarded as simulation to transmission. The XOR operation is exploited to determine the path between the source and destination nodes. Basically, 2 × 2 switching determines the straight or exchange in direction, and there are four cases as illustrated in (Fig. 2) [17].

In this figure, there is (path from source 0 to destination 0 (Fig. 2a)), (path from source 0 to destination 1 (Fig. 2b)), (path from source 1 to destination 0 (Fig. 2c)) and (path from source 1 to destination 1 (Fig. 2d)). This mechanism is the kernel of

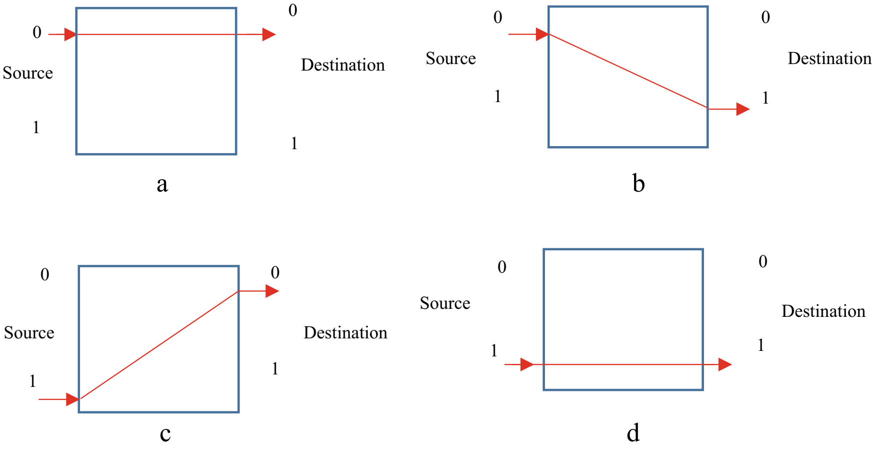


Fig. 2 Connection between source and destination

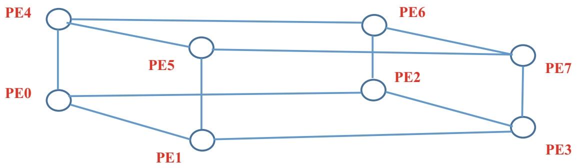


Fig. 3 Three-dimension hypercube network

the connection between client and server and it can be applied on multistage through the next equation [18]:

$$n = \log_2 P, \tag{1}$$

where n represents the number of stages and P is the number of destination nodes.

5 Hypercube Topology

In this paper, the 3D hypercube topology was exploited to apply matrix multiplication, and all threads are sent between neighboring nodes simultaneously to calculate results.

In the 3D hypercube, any node is straightly connected to three nodes. (Fig. 3 [16]) shows that the node PE7 is straightly connected to nodes PE3, PE5, and PE6 because the difference of IDs of these nodes is in only one bit. The hypercube topology can be used in many applications such as matrix transposition and matrix multiplication [19, 20].

6 Mesh Topology

In this paper, a 3D mesh topology with eight nodes was selected (Fig. 4), which illustrates the organization of nodes in the mesh topology and how these nodes implement 2D matrices. Every node takes two numbers, for example, node 1 (row 0, column 0) represents the first number of the first matrix and the first number of the second matrix [19].

7 Cloud System

Cloud computing is a paradigm to facilitate obtaining when needing to shared computing resources (application, network, storage, service) that are easy to prepare with minimum effort of management [21].

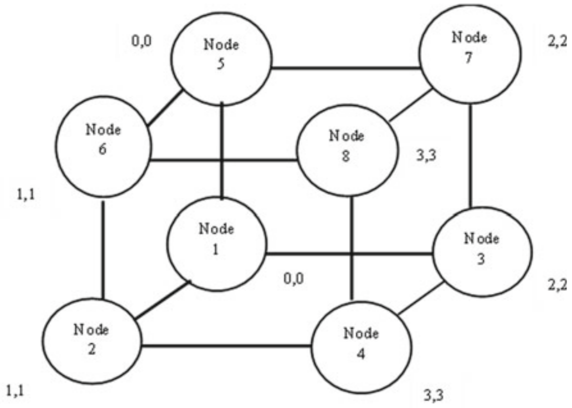


Fig. 4 Three-dimensional mesh topology with 2D matrices

There are three types of services that can be provided by cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [22]. This paper can represent a small sample of cloud computing systems (software as a service (SaaS)).

8 The Proposed System for Designing and Simulating Parallel Applications on a Hypercube Topology

The proposed system simulates the execution of the parallel application (multiplying 2D matrices A and B). A single computer is divided into eight computers as clients and eight computers as servers, the selected client communicates with eight servers using the TCP protocol, and depending on the Java threads, socket, socket server, port number, and local host, also the communication and interconnection between servers are depending on the threading of Java, socket server, port number, and local host. The following steps illustrate the algorithm:

Begin.

Step1: Input matrix $\mathbf{A} = \begin{bmatrix} i0 \; i1 \\ i2 \; i3 \end{bmatrix} and \mathbf{B} = \begin{bmatrix} j0 \; j1 \\ j2 \; j3 \end{bmatrix}$

Step2: Input client number.

Step3: Input servers from (0, 1, 2, 3, 4, 5, 6, 7).

Step4: The output is $\mathbf{C} = \begin{bmatrix} k0 \; k1 \\ k2 \; k3 \end{bmatrix}$

Step5: Connect the selected client with selected servers according to the mechanism that was illustrated in subsection (4.1).

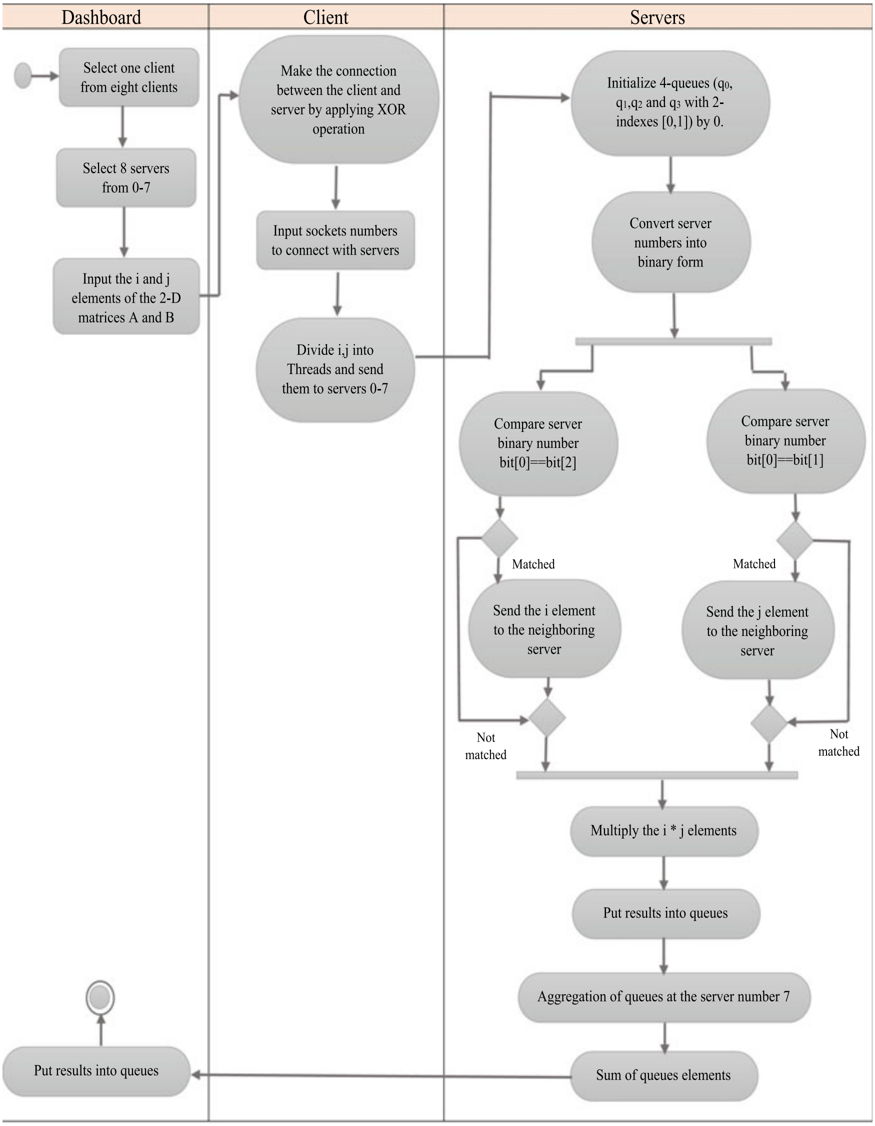
Step6: Split the matrices A and B into threads(i,j), then send them to 8 servers.

Step7: Input the queues q_0, q_1, $q_{2,}$ and q_3 with 2 indexes [0,1], then send the queues to 8 servers.

// The next steps represent the communication between servers to achieve parallel processing.

```
Step8: Convert the number of servers (0,1,2,3,4,5,6 and 7) from decimal form to
3-bits binary form, the server number will be (000, 001, 010, 011, 100, 101,110 and
111).
Step9: For each server in the network do
    If (bit_index[0] = = bit_index [2])
    If (binary bit differs by only one bit with the neighboring server)
        The communication will be enabled with the neighboring server
      The (i) thread is sent to the neighboring server.
        end if.
      end if.
    end for.
Step10: For each server in the network do.
    If (bit_index[0] = = bit_index [1]).
    If (binary bit differs by only one bit with the neighboring server)
        The communication will be enabled with the neighboring server
        The (j) thread is sent to the neighboring server
        end if
      end if.
    end for.
Step11: For each server (0..3) do
          Multiply $i_{no.} \times j_{no.}=a_{no.}$, then put $a_{no.}$ in $q_{no.}[0]$.
      End for.
Step12: For each server (4..7) do
         Multiply $i_{no.} \times j_{no.} = a_{no.}$, then put $a_{no.}$ in $q_{no}$ [1].
       End for.
Step13: Accumulate $q_0$, $q_1$, $q_{2,}$ and $q_3$ into server number (111) to make a
summation process as follows:
$q_0[0] + q_{0[1]} \rightarrow a_0 + b_0 = k_0$
$q_1[0] + q_{1[1]} \rightarrow a_1 + b_1 = k_1$
$q_2[0] + q_{2[1]} \rightarrow a_2 + b_2 = k_2$
$q_3[0] + q_{3[1]} \rightarrow a_3 + b_3 = k_3$
End.
```

The following activity diagram describes the behavior model:


Dashboard
Client
Servers
Select one client from eight clients
Select 8 servers from 0-7
Input the i and j elements of the 2-D matrices A and B
Make the connection between the client and server by applying XOR operation
Input sockets numbers to connect with servers
Divide i,j into Threads and send them to servers 0-7
Initialize 4-queues (q0, q1,q2 and q3 with 2-indexes [0,1]) by 0.
Convert server numbers into binary form
Compare server binary number bit[0]==bit[2]
Compare server binary number bit[0]==bit[1]
Matched
Matched
Send the i element to the neighboring server
Send the j element to the neighboring server
Not matched
Not matched
Multiply the i * j elements
Put results into queues
Aggregation of queues at the server number 7
Sum of queues elements
Put results into queues

9 Illustration and Experimental Results

1. Input the elements of matrices $\mathbf{A} = \begin{bmatrix} 45 & 28 \\ 33 & 12 \end{bmatrix} and \mathbf{B} = \begin{bmatrix} 56 & 60 \\ 72 & 64 \end{bmatrix}$.

2. The result is $\mathbf{C} = \begin{bmatrix} 4536 & 4492 \\ 2712 & 2748 \end{bmatrix}$.
3. Split the matrices elements A and B into threads(45, 28, 33, 12, 56, 60, 72 and 64).
4. Threads are sent from the selected client to the servers as follows:
   - Thread (45) is sent to servers (000 and 100).
   - Thread (28) is sent to servers (001 and 101).
   - Thread (33) is sent to servers (010 and 110).
   - Thread (12) is sent to servers (011 and 111).
   - Thread (56) is sent to servers (000 and 100).
   - Thread (60) is sent to servers (001 and 101).
   - Thread (72) is sent to servers (010 and 110).
   - Thread (64) is sent to servers (011 and 111).
5. To process and compute the output, the servers are communicated and threads are sent between them simultaneously according to the comparison of the matching of the bit-index (as explained in the algorithm section) as follows:
   - Server (000) sends thread (45) to the server (001).
   - Server (010) sends thread (28) to the server (011).
   - Server (101) sends thread (33) to the server (100).
   - Server (111) sends thread (12) to the server (110).
   - Server (000) sends thread (56) to the server (010).
   - Server (001) sends thread (60) to the server (011).
   - Server (110) sends thread (72) to the server (100).
   - Server (111) sends thread (64) to the server (101).
6. For every server, the multiplication operation is done and put the result in the queue as follows:
   - Server (000) multiply ($45 \times 56 = 2520$), then put 2520 into $q_0[0]$.
   - Server (001) multiply ($45 \times 60 = 2700$), then put 2700 into $q_1[0]$.
   - Server (010) multiply ($33 \times 56 = 1848$), then put 1848 into $q_2[0]$.
   - Server (011) multiply ($33 \times 60 = 1980$), then put 1980 into $q_3[0]$.
   - Server (100) multiply ($28 \times 72 = 2016$), then put 2016 into $q_0[1]$.
   - Server (101) multiply ($28 \times 64 = 1792$), then put 1792 into $q_1[1]$.
   - Server (110) multiply ($12 \times 72 = 864$), then put 864 into $q_2[1]$.
   - Server (111) multiply ($12 \times 64 = 768$), then put 768 into $q_3[1]$.
7. Queues are accumulated on the server (111) to compute the sum operation and get the output:
   - $q_0[0] + q_0[1] \rightarrow 2520 + 2016 = 4536$.
   - $q_1[0] + q_1[1] \rightarrow 2700 + 1792 = 4492$.
   - $q_2[0] + q_2[1] \rightarrow 1848 + 864 = 2712$.
   - $q_3[0] + q_3[1] \rightarrow 1980 + 768 = 2748$.

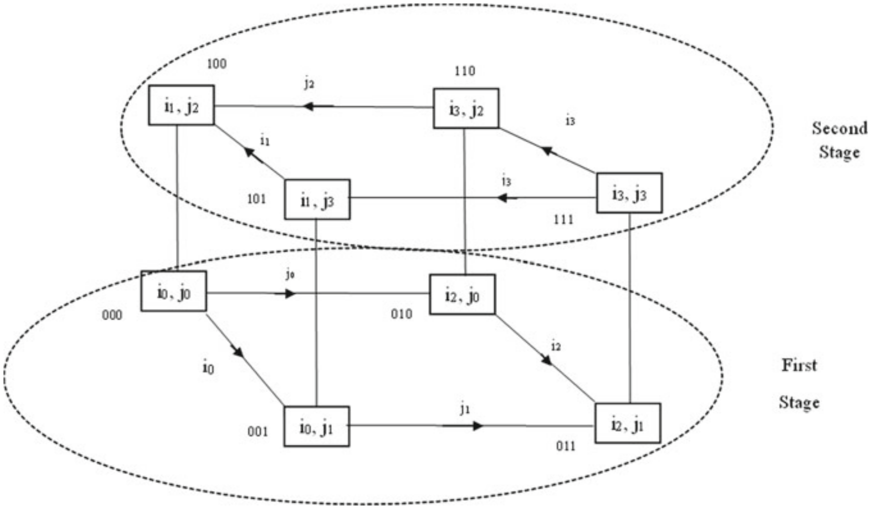


Fig. 5 Distributed matrices' elements on servers

The proposed system was achieved on Intel Core i7, 16 GB DDR4 RAM, and Windows 10 operating system. The Java programming language has been used to simulate parallel processing by dividing the elements of the matrices into threads and distributing them on the two-stage mesh topology in the form of the hypercube network. Figure 5 explains the distributed servers in two stages to implement the parallel system of matrices multiplication.

Figure 6 explains the communication between servers and how the matrices' elements are sent between neighboring servers; then calculating queues' summation to compute the final results, in these two figures, server (000) sends thread (i_0 is 45) to server (001) and thread (j_0 is 56) to server (010), server (001) sends thread (j_1 is 60) to server (011), server (010) sends thread (i_2 is 33)to server (011), server (101) sends thread (i_1 is 28) to server (100), and server (111) sends thread (i_3 is 12) to server (110) and sends thread (j_3 is 64) to server (101); then, the multiplication process is done in each server and put the results in (q_0, q_1, $q_{2,}$ and q_3), the sum of indexes $q_0[0,1]$ is calculated by the server (000 and 100), the sum of indexes $q_1[0,1]$ is calculated by the server (001 and 101), the sum of indexes $q_2[0,1]$ is calculated by the server (010 and 110), and the sum of indexes $q_3[0,1]$ is calculated by the server (011 and 111).

10 Conclusions

In this paper, we provided a simulation of a virtual network as a real network by using the Java programming language with its technologies (threads, sockets, and socket server). We applied the hypercube mesh network topology to perform the

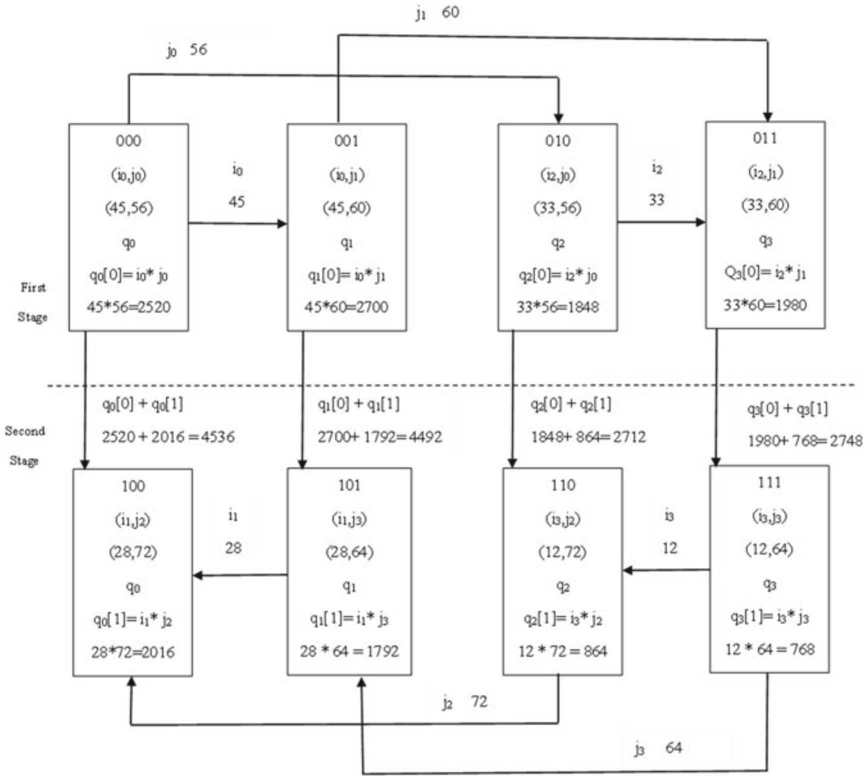


Fig. 6 Communication between servers

communication and interconnection between the servers by sending threads i and j to obtain the results. The communication between servers is achieved by positioning bits' matching conditions for each server in the binary number. We used the TCP protocol to enable communication between the client and the servers and between the servers themselves. This model is scalable, it provides a sample of the two-stage network and two-dimensional matrices, and it can be extended to (3 × 3, 4 × 4, etc.) matrices and (three dimensions, four dimensions, etc.) stages. The proposed work gives a small sample of the cloud system by providing the virtualization of parallel processing, where the virtualization can provide the ability to create multi-virtual machines and each one has its own OS and software on a single physical machine. This paper presented a multithreading model that uses the same memory, which means providing system memory and thus boosting system performance, by comparing with other models that implement the traditional process. In addition, threads provide optimal resource utilization, because each thread performs different tasks concurrently.

References

1. Soni S, Dhaliwal AS, Jalota A (2014) Behavior analysis of omega network using multi-layer multi-stage interconnection network. J Eng Res Appl 4(4): 127–130, ISSN: 2248–9622
2. Eijkhout V (2022) Parallel programming IN MPI and OpenMP. 2nd ed
3. Karthik K, Jena S, Venu Gopal T (2020) Evaluation and comparison of hypercube interconnection networks performance. Int J Adv Sci Technol 29(03):6954–6962
4. Reddy V (2021) An efficient exchanged hypercube for parallel and distributed network. Int J Recent Technol Eng 8(2S10):821–829. https://doi.org/10.35940/ijrte.B1150.0982S1019
5. Gupta N, Vaisla KS, Kumar R (2021) Design of a structured hypercube network chip topology model for energy efficiency in wireless sensor network using machine learning. SN Comput Sci 2(5):376. https://doi.org/10.1007/s42979-021-00766-7
6. Aksoy SG, Bruillard P, Young SJ, Raugas M (2021) Ramanujan graphs and the spectral gap of supercomputing topologies. J Supercomput 77(2):1177–1213. https://doi.org/10.48550/arXiv.1909.11694
7. Terry-Jack M, O'Keefe S (2023) Classifying 1D elementary cellular automata with the 0–1 test for chaos. Physica D: Nonlin Phenom 453:ISSN 0167–2789, https://doi.org/10.1016/j.physd.2023.133786
8. Eggen R, Eggen M (2021) Efficiency of distributed parallel processing using java RMI, sockets, and CORBA. In: Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA), pp 888–893. Retrieved from https://api.semanticscholar.org/CorpusID:14065828. Accessed on 21 Apr 2023
9. Inoue H, Moriyama T, Komatsu H, Nakatani T (2023) AA-sort: a new parallel sorting algorithm for multi-core SIMD processors. In: 16th international conference on parallel architecture and compilation techniques (PACT 2007). IEEE, pp 189–198. https://doi.org/10.1109/PACT.2007.4336211 (PACT 2007). Accessed on 23 Apr 2023.
10. Elnashar A (20211) Parallel performance of MPI sorting algorithms on dual–core processor windows-based systems. Int J Distrib Parallel Syst (IJDPS), 2(3):1–14. https://doi.org/10.5121/ijdps.2011.2301
11. Saleem S, Lali MI, Nawaz MS (2014) Multi-core program optimization: parallel sorting algorithms in Intel Cilk Plus. Int J Hybrid Inform Technol 7(2):151–164. https://doi.org/10.14257/ijhit.2014.7.2.15
12. Damrudi M, Aval KJ (2012) A parallel search on hypercube interconnection network. J Comput Sci Comput Math 2(1):1–4
13. Abba IM, Aziz NAB, Eaganathan U, Gabriel J (2013) LAN Chat Messenger (LCM) using java programming with VOIP. In: Proceeding of the 3rd international conference on research and innovation in information systems, pp 428–433, doi: https://doi.org/10.1109/ICRIIS.2013.6716748. Accessed on 25 Apr 2023
14. Panigrahi SK, Chakraborty S, Mishra J (2012) Statistical bound of bubble sort algorithm in serial and parallel computations. Int J Comput Sci Netw (IJCSN) 1(1):2277–5420
15. Rinku DR, Asha Rani M (2017) Analysis of multi-threading time metric on single and multi-core CPUs with matrix multiplication. In: Third international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB), Chennai, 2017, pp 152–155. https://doi.org/10.1109/AEEICB.2017.7972402. Accessed on 27 Apr 2023
16. Amodu OA, Othman M, Yunus NAM, Hanapi ZM (2021) A primer on design aspects and recent advances in shuffle exchange multistage interconnection networks. Symmetry 13(3):378. https://doi.org/10.3390/sym13030378
17. Jain T (2020) Nonblocking on-chip interconnection networks. Doctor of Engineering dissertation, Technical University of Kaiserslautern, Germany, (2020).
18. Sharma V, Ansari AQ, Mishra R (2021) A novel design layout of three disjoint paths multistage interconnection network & its reliability analysis. IJPCC 17(4):390–403. https://doi.org/10.1108/IJPCC-04-2021-0094

19. Liu A, Wang S, Yuan J, Li J (2017) On g-extra conditional diagnosability of hypercubes and folded hypercubes. Theoretic Comput Sci 704:62–73, ISSN 0304–3975, https://doi.org/10.1016/j.tcs.2017.09.030
20. Khudhair MM, Rabee F, Al-Rammahi A (2023) A new fractal topologies based on hypercube interconnection network. Al-Salam J Eng Technol (AJEST) 2(2):128–139. https://doi.org/10.55145/ajest.2023.02.02.016
21. Alsahly AM (2023) Feasibility for cloud computing: when to move your business to the cloud. Retrieved from https://www.researchgate.net/publication/352311954, Accessed on 6 may 2023
22. Banimfreg BH (2023) A comprehensive review and conceptual framework for cloud computing adoption in bioinformatics. Healthcare Anal 3. https://doi.org/10.1016/j.health.2023.100190

Integrating Artificial Intelligence for Adaptive Decision-Making in Complex System

Ajay Verma and Nisha Singhal

Abstract The integration of AI techniques in systems engineering can revolutionize decision-making in complex systems. This research investigates AI's role in enhancing adaptive decision-making and addresses integration challenges. AI technologies, like machine learning and cognitive computing, handle large data volumes, identify patterns, and make accurate predictions, enabling decision-makers to gain valuable insights into system behavior, risks, and performance optimization. The research develops intelligent algorithms for real-time data analysis, pattern recognition, and anomaly detection, facilitating proactive decision-making. Intelligent decision support systems integrate AI technologies, providing real-time insights and recommendations to optimize performance and enhance system resilience. The research evaluates AI-based approaches through case studies, assessing their performance and effectiveness. It also addresses challenges such as data privacy, transparency, and system reliability, offering practical guidelines for successful AI integration. In conclusion, this research explores AI integration for adaptive decision-making in complex systems, advancing systems' engineering and providing insights for practitioners and researchers implementing AI for effective decision-making in dynamic environments.

Keywords AI integration · Adaptive decision-making · Complex systems · Intelligent algorithms · System engineering

A. Verma (✉)
Department of Mechanical Engineering, Maulana Azad National Institute of Technology Bhopal, Bhopal, India
e-mail: avmanit@gmail.com

N. Singhal
Department of Mathematics, Indian Institute of Information Technology Bhopal, Bhopal, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_8

1 Introduction

In recent years, the field of artificial intelligence (AI) has witnessed significant advancements, revolutionizing various industries and domains [1]. One area where AI has shown tremendous potential is in enhancing adaptive decision-making processes in complex systems As the complexity of systems continues to grow, traditional decision-making approaches often fall short in handling the intricate dynamics and uncertainties associated with such systems. However, by integrating AI techniques into systems engineering, decision-makers can harness the power of intelligent algorithms and frameworks to gain valuable insights, optimize performance, and enhance overall system resilience.

This study aims to investigate how artificial intelligence can be integrated to support adaptive decision-making in complex systems. We will look into the ways that artificial intelligence (AI) technologies, such as machine learning, expert systems, and cognitive computing, might enhance decision-making and deal with integration-related issues [1]. In addition, we will concentrate on creating sophisticated frameworks and algorithms that can efficiently evaluate and comprehend complicated system data, giving decision-makers timely insights all the way through the system lifetime [2].

2 Literature Review

In complex systems, decision-making is often hindered by the sheer volume and complexity of available data. Traditional approaches struggle to analyze vast information, leading to suboptimal decisions and increased risks. AI techniques, including machine learning algorithms, are highly effective in handling large data volumes, identifying patterns, and making accurate predictions [2]. Leveraging these AI capabilities helps decision-makers to gain a deeper understanding of system behavior, risks, and performance optimization strategies. AI enables adaptive decision-making in dynamic, uncertain environments, improving performance by analyzing real-time data [3].

Developing intelligent algorithms is crucial for AI integration. They analyze complex system data, extract insights, and enable real-time trend identification. Anomaly detection algorithms spot potential failures or risks, aiding proactive mitigation [4]. These algorithms empower decision-makers to optimize system performance and enhance resilience. Integrating AI requires intelligent decision support systems providing real-time insights and recommendations based on data and AI-driven algorithms [4]. They process vast data, guide decision-makers, and consider constraints, objectives, and risks. Case studies and comparisons with traditional methods evaluate AI's impact in complex system environments [5].

Case studies demonstrate AI's effectiveness in adaptive decision-making, for example, optimizing energy consumption in a smart grid system through AI-driven

Table 1 Artificial intelligence and intelligence systems in complex decision-making

No.	Authors and year	Main concept
1	Russell and Norvig [1]	Advancements in the field of AI
2	Ahmad et al. [2]	Role of AI in improving decision-making
3	Johnson et al. [3]	Machine learning for handling large data volumes
4	Bengio et al. [8]	Adaptive decision-making in dynamic and uncertain environments
5	Holland [9]	Anomaly detection algorithms in complex systems
6	Jackson [10]	Performance evaluation of AI-driven approaches
7	Kim et al. [11]	Comparison of rule-based and AI-driven decision-making
8	Rajaraman and Ullman [12]	Addressing bias in AI algorithms

analysis [3]. Comparisons assess factors like energy efficiency, cost-effectiveness, and system stability, offering empirical evidence of AI benefits. Additionally, comparisons with traditional methods evaluate AI techniques' strengths and limitations [6]. Considerations must address potential bias in AI algorithms, which can lead to discriminatory outcomes [7]. Ensuring ethical AI development and deployment is essential to avoid unintended consequences and ensure fairness in decision-making processes. Table 1 is highlights applications of various AI techniques in decision-making reported in literature.

In conclusion, integrating artificial intelligence into adaptive decision-making processes in complex systems offers significant potential for enhancing performance and resilience. AI technologies, including machine learning, expert systems, and cognitive computing, enable the analysis of large volumes of data, adaptive decision-making in uncertain environments, and the development of intelligent algorithms and decision support systems.

3 Overview of Artificial Intelligence Techniques for Decision-Making

Artificial intelligence (AI) is a potent instrument for decision-making across a range of fields, including complex systems [1]. AI uses sophisticated models and algorithms to evaluate data and come to well-informed conclusions. Artificial intelligence (AI) improves decision-making's precision, effectiveness, and flexibility in complex and dynamic situations [3]. An overview of well-known AI decision-making approaches is given in this section.

Automated Learning Within artificial intelligence, machine learning creates models and algorithms that learn from data to make predictions or judgments without the need for explicit programming. It comprises three types of learning: unsupervised learning, which finds patterns in unlabeled data, reinforcement learning, which learns by interacting with an environment, and supervised learning, which detects patterns and makes predictions from labeled datasets. Effective use of machine learning in decision-making is demonstrated by its application to customer behavior, fraud detection, and predictive maintenance [13]. They excel in decision-making scenarios requiring domain-specific knowledge and have been applied in healthcare diagnosis, financial analysis, and quality control [14].

Natural Language Processing (NLP): NLP enables machines to understand and process human language, involving speech recognition, natural language understanding, and generation [15]. NLP supports decision-making by analyzing text data, such as customer feedback analysis, sentiment analysis, and text summarization [16].

Neural Networks: Neural networks, inspired by the human brain, recognize complex patterns and make predictions by processing information through interconnected nodes [3]. Deep learning, a subset, involves training deep architectures to learn hierarchical data representations [17]. Neural networks have been applied effectively in decision-making tasks like image and speech recognition, anomaly detection, and financial forecasting [9].

Genetic Algorithms: Genetic algorithms optimize solutions iteratively based on fitness criteria, inspired by natural selection and evolution [9]; they are valuable in decision-making with multiple objectives and challenging optimal solution search, applied in resource allocation, scheduling, and portfolio optimization [6].

Fuzzy Logic: Fuzzy logic deals with uncertainty and allows reasoning in situations with degrees of truth. It is useful in decision-making tasks involving subjective or uncertain information, applied in areas like control systems, risk assessment, and decision support systems [18]. By employing these AI techniques, decision-makers harness data analysis, pattern recognition, and intelligent reasoning for effective decisions in complex systems (see Fig. 1 for AI integration components in complex decision-making).

4 Advantages, Barriers, and Uses of AI in System Complexity

Use of AI in decision-making in systems includes many advantages along with some barriers. This section highlights some of the benefits, challenges, and uses of AI in system decisions.

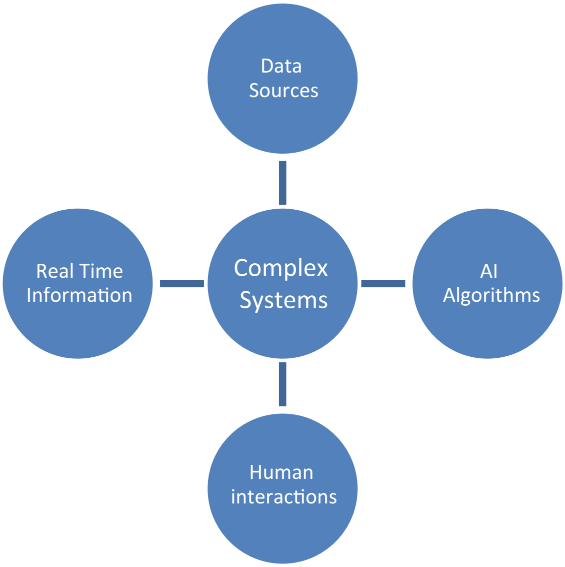


Fig. 1 Various components of AI integration in complex decision-making

4.1 Advantages

Enhanced decision-making, efficiency, automation, real-time analysis, learning, adaptability, and optimization are some of the benefits of using AI in decision-making for systems.

4.2 Barriers

- Data Quality and Availability: AI relies heavily on high-quality and relevant data for training and decision-making. Obtaining sufficient and reliable data can be a challenge in complex systems due to data fragmentation, quality issues, and privacy concerns [19]. Ensuring data accessibility, accuracy, and reliability is crucial for the success of AI integration.
- Interpretability and Explainability: AI models often operate as black boxes, making it difficult to understand their decision-making processes. In complex systems, where decisions may have significant consequences, interpretability and explainability of AI algorithms become crucial [20]. Decision-makers need to trust and understand the reasoning behind AI-generated insights and recommendations.
- Ethical and Legal Considerations: Integrating AI into complex systems raises ethical and legal concerns [21]. These include issues related to data privacy, algorithmic bias, accountability, and responsibility. Addressing these concerns requires the establishment of ethical frameworks, guidelines, and regulatory

measures to ensure that AI integration is aligned with societal values and legal requirements.

- System Complexity and Scalability: Complex systems often involve numerous interconnected components, making it challenging to integrate AI seamlessly. Scaling AI solutions to large-scale and distributed systems requires careful design, infrastructure support, and considerations for system complexity [15]. The integration process must account for the scalability and compatibility of AI algorithms within the existing system architecture [22].
- Human–Machine Collaboration: Effective integration of AI in complex systems requires collaboration between humans and machines [23]. This collaboration entails understanding the strengths and limitations of AI systems, establishing clear roles and responsibilities, and facilitating human oversight and intervention when necessary. Ensuring a smooth human–machine interaction is essential to harnessing the full potential of AI in decision-making processes [24].

5 Research Contributions of the AI Approaches

A brief description of research contributions of the approaches described above is presented below.

Machine Learning

Contribution: Machine learning techniques contribute to decision-making by automating the process of learning from data. This allows for data-driven predictions and classifications.

Application: Machine learning has been successfully applied in various decision-making tasks, including predictive maintenance, fraud detection, and customer behavior analysis.

Expert Systems

Contribution: Expert systems emulate human expertise in specific domains, contributing to decision-making by providing structured, rule-based reasoning.

Application: Expert systems are valuable in decision-making scenarios that require domain-specific knowledge, such as healthcare diagnosis, financial analysis, and quality control.

Natural Language Processing (NLP)

Contribution: NLP techniques contribute to decision-making by enabling the analysis of large volumes of textual data, providing insights into customer opinions, market trends, and business intelligence.

Neural Networks

Contribution: NN may identify patterns which are complex in predicting data and various tasks.

Genetic Algorithms

Contribution: Genetic algorithms contribute to decision-making by optimizing solutions based on fitness criteria, particularly in scenarios with multiple objectives. Application: Genetic algorithms have been applied to decision-making problems like resource allocation, scheduling, and portfolio optimization.

Fuzzy Logic

Contribution: Fuzzy logic contributes to decision-making by handling uncertainty and imprecision, providing a framework for reasoning in situations with degrees of truth.

Application: Fuzzy logic is applied in decision-making tasks involving subjective or uncertain information, such as control systems, risk assessment, and decision support systems.

These AI techniques collectively contribute to enhancing decision-making within complex systems by leveraging data analysis, pattern recognition, and intelligent reasoning. They provide various tools and methodologies to address different aspects of decision-making, from data processing and interpretation to optimization and handling uncertainty.

6 Choice and Recommendation of Any One Approach

The choice of the best AI approach among the ones mentioned (machine learning, expert systems, natural language processing, neural networks, genetic algorithms, and fuzzy logic) depends entirely on the specific problem you are trying to solve and the context in which you are working. Each of these approaches has its strengths and weaknesses, making them suitable for different types of tasks. Here are some considerations for selecting the best approach for a given problem:

Machine Learning

When to Use: Use machine learning when you have a large dataset and want the algorithm to learn patterns and make predictions or classifications.

Examples Predictive maintenance, image recognition, recommendation systems. **Expert Systems**: **When to Use**: Use expert systems when you have well-defined rules and domain-specific knowledge to make decisions.

Examples Healthcare diagnosis, financial analysis, quality control.

Natural Language Processing (NLP)
When to Use: Use NLP when you need to analyze and derive insights from textual data.

Examples Sentiment analysis, catboats, text summarization.

Neural Networks
When to Use: Use neural networks, especially deep learning, for tasks involving complex pattern recognition, such as images, speech, and sequential data.

Examples Image recognition, speech recognition, natural language translation.

Genetic Algorithms
When to Use: Use genetic algorithms when you need to optimize solutions based on multiple criteria and traditional optimization methods are not suitable.

Examples Resource allocation, scheduling, portfolio optimization.

Fuzzy Logic
When to Use: Use fuzzy logic when dealing with problems that involve uncertainty and imprecision, where traditional binary logic is inadequate.

Examples Control systems, risk assessment, decision support systems.

To determine the best approach for your specific problem, consider the following factors:

Nature of Data: What type of data are you working with? Is it structured or unstructured? Does it contain text, images, or other forms of data?

Complexity of the Problem: Is the problem relatively straightforward or highly complex? Some problems require more advanced techniques like deep learning.

Availability of Domain Knowledge: Do you have access to domain-specific expertise and well-defined rules, or do you need the algorithm to learn from data?

Data Size: Do you have a large dataset, or is the dataset relatively small? Machine learning methods may require more data to generalize effectively.

Interpretability: Do you need to explain the reasoning behind decisions? Some methods, like expert systems, provide more transparent decision-making.

Objective: What is your primary goal? Are you trying to classify, predict, optimize, or perform some other specific task?

Resource Constraints: Consider the computational resources available, as some techniques, like deep learning, can be computationally intensive.

In practice, it is often beneficial to experiment with multiple approaches and compare their performance on your specific problem to determine which one works best. Additionally, the availability of data and expertise within your organization can influence your choice of approach. Ultimately, the "best" approach is the one that aligns most closely with your problem's requirements and constraints.

7 Future Perspectives

Integrating artificial intelligence (AI) into complex decision-making processes holds immense future potential as AI technologies evolve and advance. These advancements offer new avenues for enhancing performance, resilience, and efficiency across various industries and domains. In this exploration of future perspectives on AI integration in complex decision-making, several key areas stand out. Advanced machine learning techniques, which have already demonstrated their prowess in handling vast data and accurate predictions, are expected to evolve further to handle even more complex and diverse datasets.

This development will empower decision-makers to extract deeper insights from intricate systems and make real-time, informed choices. Additionally, the demand for transparency and interpretability in AI models, driven by their growing complexity, will lead to advancements in explainable AI. This will enable decision-makers to better comprehend the underlying reasoning behind AI-based decisions, fostering greater trust in the system. Collaborative decision-making is set to play a pivotal role, with AI acting as a supportive tool for human decision-makers, rather than a replacement, thus improving the efficiency of complex systems. The integration of AI with the Internet of Things (IoT) will create a powerful synergy, allowing real-time data analysis from IoT devices to inform adaptive decision-making.

Ethical and responsible AI practices will be paramount, requiring the development of frameworks and guidelines to address bias, fairness, and transparency in AI algorithms, backed by regulatory measures. Lastly, AI systems that continuously learn and adapt over time will become increasingly significant, enabling decision-makers to leverage AI that evolves alongside the complex systems it serves. In conclusion, AI's integration into complex decision-making processes is poised to bring transformative benefits as it evolves, facilitating data-driven choices, optimized performance, and enhanced system resilience in an era of increasing complexity and uncertainty.

8 Conclusions

This review highlights the enormous potential of integrating artificial intelligence (AI) into complex decision-making within intricate systems. Advancements in AI technologies, including machine learning, expert systems, natural language processing, neural networks, genetic algorithms, and fuzzy logic, have transformed decision-making by offering valuable insights, automation, real-time analysis, and optimization. AI's benefits in complex systems are multifaceted, ranging from enhanced decision-making through data analysis to automating repetitive tasks, thus boosting efficiency and freeing human resources for more strategic endeavors. Real-time insights from AI support proactive decision-making and early anomaly detection. AI techniques, such as reinforcement learning and neural networks, bolster adaptability and learning, enabling complex systems to navigate dynamic, uncertain environments. Additionally, AI algorithms can optimize systems by identifying patterns and dependencies, enhancing overall efficiency.

Nevertheless, integrating AI into complex systems presents unique challenges, including data quality, interpretability, ethical concerns, system complexity, and the necessity for effective human–machine collaboration. Notwithstanding these difficulties, there is hope for the future of AI in complex systems adaptive decision-making, as new developments in the field are expected to overcome these barriers. The growing influence of AI will profoundly alter how decisions are made in a variety of industries, including manufacturing, transportation, healthcare, and finance. Organizations can improve system resilience in the face of growing complexity and uncertainty, optimize performance, and make data-driven decisions by utilizing AI technologies. In conclusion, by utilizing the potential of intelligent algorithms and frameworks, integrating AI into complex systems offers a powerful remedy to the drawbacks of conventional decision-making techniques. Successful integration requires close consideration of interpretability, ethics, system complexity, data quality, and human–machine cooperation. The future is incredibly promising for the advancement and use of AI in adaptive decision-making in complex systems.

References

1. Russell SJ, Norvig P (2016) Artificial intelligence: a modern approach. First Edition, Pearson
2. Ahmad MO, Beg MS, Ahmad I (2020) Role of artificial intelligence in decision-making processes. In: Handbook of research on emerging business models and managerial strategies in the nonprofit sector. IGI Global, pp 107–128
3. Johnson L, Brown M, Lee J (2020) Adaptive decision-making in dynamic environments using AI techniques. In: Proceedings of the international conference on artificial intelligence, pp 245–252
4. Chen L, Wang H, Zhang Y (2018) Anomaly detection algorithms for proactive risk mitigation in complex systems. IEEE Trans Syst Man Cybern 48(5):769–783
5. Gandomi A, Alavi AH, Yun GJ, Azar AT (2021) Artificial intelligence in complex system decision-making: a comprehensive review. Complexity 1–23
6. Deb K (2001) Multi-objective optimization using evolutionary algorithms. John Wiley & Sons
7. Caliskan A, Bryson JJ, Narayanan A (2017) Semantics derived automatically from language corpora contain human-like biases. Science 356(6334):183–186
8. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
9. Holland JH (1992) Adaptation in natural and artificial systems. MIT Press
10. Jackson P (1999) Introduction to expert systems. Addison-Wesley
11. Kim S, Park J, Lee C (2022) A comparison of rule-based and AI-driven decision-making systems for traffic management in smart cities. Int J Urban Sci 26(1):63–80
12. Rajaraman A, Ullman JD (2011) Mining of massive datasets. Cambridge University Press
13. Floridi L, Cowls J, Beltrametti M, Chatila R, Chazerand P, Dignum V, Tamburrini G (2018) AI4People—an ethical framework for a good AI society: opportunities, risks, principles, and recommendations. Minds Mach 28(4):689–707
14. Gandomi A, Alavi AH, Yun GJ, Azar AT (2021) Artificial intelligence in complex system decision-making: a comprehensive review. Complexity 2021:1–23

15. Jurafsky D, Martin JH (2019) Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Pearson
16. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
17. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
18. Mendel JM (1995) Fuzzy logic systems for engineering: a tutorial. Proc IEEE 83(3):345–377
19. Duda RO, Hart PE, Nilsson NJ (2000) Pattern classification. John Wiley & Sons
20. Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215
21. Floridi L, Cowls J, Beltrametti M, Chatila R, Chazerand P, Dignum V, Tamburrini G (2018) AI4People—an ethical framework for a good AI society: opportunities, risks, principles, and recommendations. Mind Mach 28(4):689–707
22. Brynjolfsson E, McAfee A (2014) The second machine age: work, progress, and prosperity in a time of brilliant technologies. W. W. Norton & Company
23. Chen H, Chiang RH, Storey VC (2012) Business intelligence and analytics: from big data to big impact. MIS Q 36(4):1165–1188
24. Kusiak A (2018) Artificial intelligence for smart and sustainable manufacturing. CRC Press

Qualitative Research Reasoning on Dementia Forecast Using Machine Learning Techniques

Tanvi Kapdi and Apurva Shah

Abstract The rise in mental health issues and the demand for high-quality medical care have prompted researchers to investigate how machine learning might be used to treat mental health issues. Dementia is a disease that causes loss of cognitive skills in a way that interferes with a person's day-to-day activities. It causes a breakdown of brain function, comprehension, recognition, reasoning, and behavioral abilities to the point where a person experiences difficulties in day-to-day activities. Dementia gradually kills the brain cells and causes people to lose their reading and thinking capabilities. According to the Lancet report, the incidences of dementia cases in India are predicted to nearly triple by 2050. According to the survey, the number of cases is roughly predicted to quadruple to 153 million by 2050. This research presents the analysis and findings related to forecasting dementia using machine learning techniques. The study has been conducted using the Open Access Series of Imaging Studies (OASIS) dataset. This dataset has been explored by using various machine learning algorithms such as support vector machine, random forest, decision tree, logistic regression, AdaBoost, and XGBoost. The conclusion has been drawn regarding the evaluation metrics in accuracy. It has been found that XGBoost gave the best result with 93.02% accuracy. With XGBoost, it is simple to determine the ideal number of boosting iterations in a single run.

Keywords Machine learning · OASIS · Dementia · Classification

T. Kapdi (✉) · A. Shah
The Maharaja Sayajirao University Baroda, Vadodara, India
e-mail: tanvi.kapdi-cse@msubaroda.ac.in

A. Shah
e-mail: apurva.shah-cse@msubaroda.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_9

1 Introduction

With 1.97 billion people, India makes up 18% of the world's total population and has already overtaken China in 2023. By 2050, it is anticipated that the percentage of Indians aged 60 and over will have increased to about 20%. Life expectancy in India has progressively grown from 43.9 years in 1960 to 72.4 years in 2023 [1]. Dementia is one of the several neurodegenerative disorders that leads to 10% of global mortality [2]. Few dementia patients are unable to manage their emotions, and personalities can alter. Dementia ranges in intensity from a milder stage to severe. Apart from therapy, there is no cure. Focusing on the early stages, prompt care, and delaying the disease is essential [3]. A comprehensive medical history supplied by patients and their families, a neurological examination, and cognitive testing are used to make a clinical diagnosis of dementia [4]. To rule out further dementia causes, additional tests such as blood tests, CT-Scan MRI should be carried out [5]. Although there have been several clinical tests available for the early identification of dementia, still the creation of improved diagnostic tools is crucial [6]. While automated dementia detection using machine learning is becoming popular, its clinical use has not yet been fully utilized. Hence, the need to build reliable and generalizable models that produce decisions and accurate predictions is crucial [7].

The goal of machine learning is to create a system that can learn from experience by using sophisticated statistical methods [8]. Many commonly used machine learning algorithms like support vector machine, logistic regression, decision tree, etc., are used to forecast and categorize future events [9]. It is helping many researchers to gather crucial information from the data, offer personalized experiences, and create automated intelligent systems. Most research, studies, and experiments using machine learning use supervised methods in particular to predict the disease [10]. In supervised learning, all data instances should represent the words, characteristics, and values [11]. More specifically, supervised learning is a technique for classifying information using structured training data [12]. Unsupervised learning however does not require supervision to make predictions. Its primary objective is to handle data without supervision [13].

2 Related Work

Several approaches are discussed by the researchers in identifying the onset of dementia at an early stage. Deep convolution encoders are used by Martinez-Murcia et al. [14] to investigate data analysis of dementia. We can extract MRI characteristics from MRI pictures that describe a person's cognitive symptoms as well as the underlying neurodegenerative process using data-driven deconstruction of MRI images [15]. The distribution of the collected feature is then examined using regression and classification analysis, and the effect of each coordinate of the auto-encoder manifold on the brain is estimated. With imaging-derived indicators and MMSE or ADAS score, an AD diagnosis may be predicted with 75% accuracy [16]. The three primary types of dementia intervention, according to the National Academy of Medicine [17, 18], are cognitive training, hypertension treatment, and increased physical activity. Alzheimer's disease is the kind of disease that affects people the most [19]. Vascular Alzheimer's VAD is the second most persistent form of dementia, followed by Lewy bodies. Other forms are linked to alcohol misuse, infections, and brain trauma. Tatiq and Barber [20] hypothesized that Alzheimer's can be avoided by focusing on modifiable vascular risk factors. Williams et al. [21] used four alternative methods to derive the estimates of cognitive performance based on neuropsychological demographic data: decision tree, SVM, Naive Bayes, and Nearest neighbor. The accuracy of Naive Bayes was the greatest in this situation because average values were used to fill in the gaps left by the missing data [22]. Tenfold cross-validation is applied to the ADNI trial, and the results suggest a strong correlation between the neuropsychological outcomes and imaging [23]. In a machine learning model, an algorithm is designed to recognize particular patterns which means that it analyzes the data and identifies the hidden structures [24]. Feature extraction determines the input function and applies it to the dataset, following which the algorithm of the model utilizes a set of training values and creates a method to forecast the result along with saving the process for later use [25]. The most common supervised machine learning model called the support vector machine employs a swift and well-grounded classification technique that outperforms when small-scale data is given to training [26]. Classification complications are also settled through the logistic regression method. It is a measure of forward-looking analysis that is discovered on the presumption of probability. The probability of a categorical dependent variable is forecasted using the logistic regression method [27]. In logistic regression, the dependent variable is a binary variable with data coded as 1 or 0. Determining an association between features and the probability of a certain output is the goal of logistic regression [28]. The term itself inferred that it displays the predictions from a series of attribute-based splits using a flowchart that resembles a tree [29]. In simple words, decision trees are nothing more than a collection of if-else expressions that determines if the condition is true, and if it is, it moves on to the next node associated with that choice [30]. Random forest, on the other hand, uses diverse specimens, establishes decision trees, and uses their average for classification and majority vote for regression [31]. The potential of random forests to handle samples with continuous variables as in regression and categorical values is its most crucial quality [32]. AdaBoost is used as an ensemble machine learning method. The most popular estimator used with AdaBoost is a decision tree with one level which is also known as decision stump [33]. This algorithm creates a model while assigning each data variable an equal weight. Then, it gives points that were incorrectly categorized as larger weights. It will continue to train the model until a smaller error is seen [34]. Several weak model predictions are combined using this ensemble learning technique to get a stronger prediction [35, 36]. It can also handle larger datasets and achieve state-of-the-art performance handling of missing values enabling it to handle real-world data without the need for extensive preprocessing [37].

3 Materials and Methods

3.1 Preprocessing

The major aim of the system is to forecast dementia in subjects using the OASIS dataset which has 373 rows X 15 columns [38]. The 15 columns suggest the various features collected as a part of the dataset. And 373 rows are the entries of the subjects. Table 1 shows the attributes of the dataset which is considered for the research. The study consists of MRI data from participants ranging in the age group of 60–98 [39]. The dataset is thoroughly analyzed to identify the importance of every attribute.

To start with, the dataset has been examined for any categorical values, and it is found that a few unmitigated qualities are present in the dataset [40]. The gender and group attribute columns are among them and are changed into binary values 0 and 1. The relationship among the credits has been checked by utilizing the "correlation framework" work given gathering ascribes and marked to encompass them better [41]. Later, the data is evaluated for any invalid or null qualities, and the middle worth is utilized to fill in those missing qualities for the two elements [42]. Then, in order for the model to predict, the elements were assigned to create the expectation, and the target worth was set [43]. The split has been prepared using stratified sampling. The distribution of item categories as a consequence is balanced. Besides a few scatterplot, representation has been done in WEKA to understand the test cases better [44]. The ski-kit learn library has been used to implement each model.

Table 1. OASIS dataset

Sr. No.	Attribute	Description
1	ID	Identification
2	M/F	Male/female
3	Age	Age in years
4	Hand	Left/right hand
5	EDUC	Education
6	SES	Socio economic status
7	MMSE	Mini-mental state exam
8	CDR	Clinical dementia rating
9	eTIV	Estimated total intracranial volume
10	nWBV	Normalized whole brain volume
11	ASF	Atlas scaling factor
12	Delay	Delay

3.2 Constructing the Diagnostic Model

To create the diagnostic model, different classification algorithms like Support Vector Machine, logistic regression, decision tree, random forest, AdaBoost, and XGBoost have been undertaken for training as shown in Fig. 1. Additionally, in order to assess the diagnostic model using the test set, performance evaluations were compared [45]. First, there has been no fine-tuning of the Support Vector Machine model implementation. Here it takes regularization parameter C as 1 when not tuned, and it employs the radial basis function for the kernel. The model has then been adjusted using grid search.

Several regularization parameters such as distinct types of kernels- sigmoid, linear, poly, the RBF and gamma values, C have been chosen for the parameter combinations. Moreover, fivefold cross-validation has been used to assess each potential combination. When the model was trained once more, the results were significantly better. This version has been used to calculate the confusion matrix. The logistic regression model has been used the same as in SVM. The independent and dependent variables are established. For determining decision limits and probability forecasts, it employs

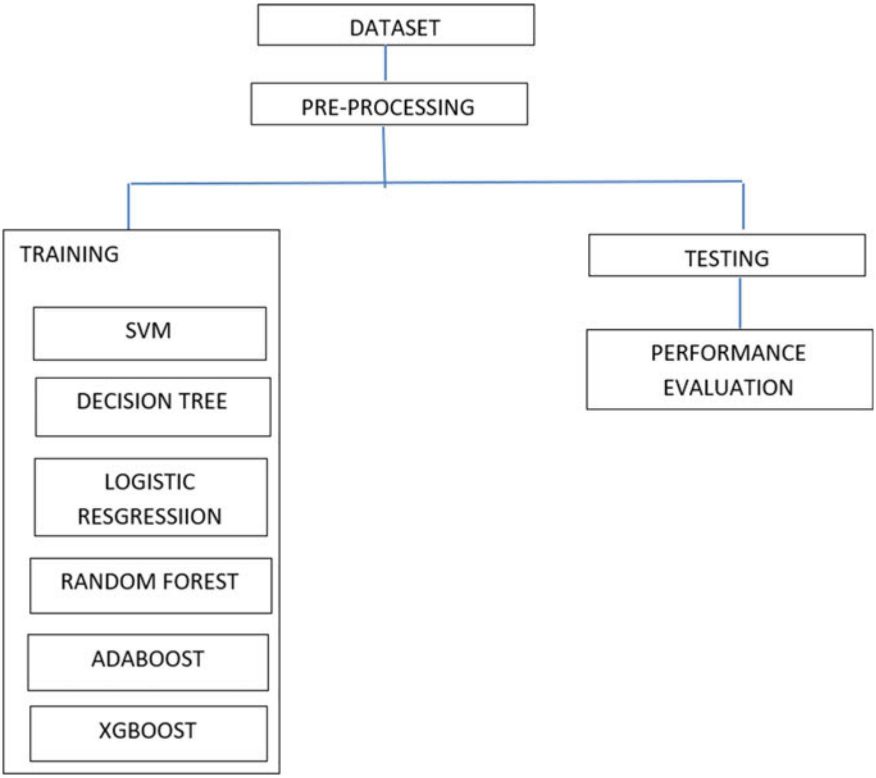


Fig. 1 Proposed model framework

the sigmoid function. The l2 penalty and various regularization parameters' value C have been employed for the fine-tuning, which is the only distinction [46]. The model was earlier instructed in the absence of any fine-tuning, and later, grid search was utilized to identify the real parameters. In this case, the Gini criteria have been used as a constant gain to assess the tree's depth. Every node is chosen, and it goes deeper. It forecasts the outcomes for the optimal solution after examining all the node's outputs. Random forest consists of several decision trees. The data is preprocessed and some random samples are chosen from the dataset for training [47]. The random forest was first implemented without fine-tuning just like the previous models and the grid search was applied with fivefold CV and a variety of feature combinations, including the n estimators, the feature to utilize the number of parameters exhibited on each split, the level in the tree, and the process for choosing samples for training each tree [48]. The Gini criteria have been applied to assess the tree's quality.

Finding the ideal parameter is crucial since the AdaBoost algorithm's prediction output is tightly tied to the number of base regression loops, learning rate, base regression, and loss function, [49]. The grid search approach organizes all potential parameter values after collecting them all. The mean square error for each forecast model is compared using the cross-validation grid search method, within the bounds of the set of optimum model parameters to determine the best prediction model that avoids the issue of poor accuracy [45]. Like extreme gradient boost, XGBoost employs the "residual error" learning strategy which involves fitting regression trees to the residuals [50].

4 Results and Discussion

The dataset has been examined for any clear-cut entries, and it is found that a few unmitigated qualities are present in the dataset. The relationship among the credits has been checked by utilizing the "correlation framework" work given gathering ascribes and marked to encompass them better.

$$\rho_{x,y} = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y}.$$

Orientation, socio-economic status, and atlas scaling factor showed a nearer connection with the gathering trait. Later, the data is evaluated for any invalid or null qualities. Socio-economic status and mini-mental state exam sections have 18 and 3 missing qualities, individually. As referenced before, socio-economic status highlights a lost connection with the objective trait. Thus, the elements have been allotted to make the expectation, and the objective worth has been set so that the model can foresee. Irregular examining has been utilized for the split, yet this creates an irregularity in preparing and validating the split. In this way, delineated testing has been anticipated with a preparing training split of 70% and a testing phase of 30%. From that point forward, normalization has been enforced to mark the scaling of the highlights. Besides a few histogram representation has been done in WEKA to understand the test cases better as shown in Fig. 2 which presents that the age of most subjects is in the range of 60–90 along with a high bar for educational and socio-economic status. Atlas scaling factor, normalized whole brain volume, and intracranial total estimated volume are also relatively high.

The model has been tested using 30% of the test data collected from the OASIS dataset and trained the model using the remaining 70% of the data. Table 2 displays the optimal parameter selected for the diagnostic model based on grid search selection. Hence, the default algorithm parameters are not shown.

The precision, recall, f-score, and accuracy metrics are evaluated for the model as shown in Table 3 XGBoost performed the best with an accuracy of 93.02, recall

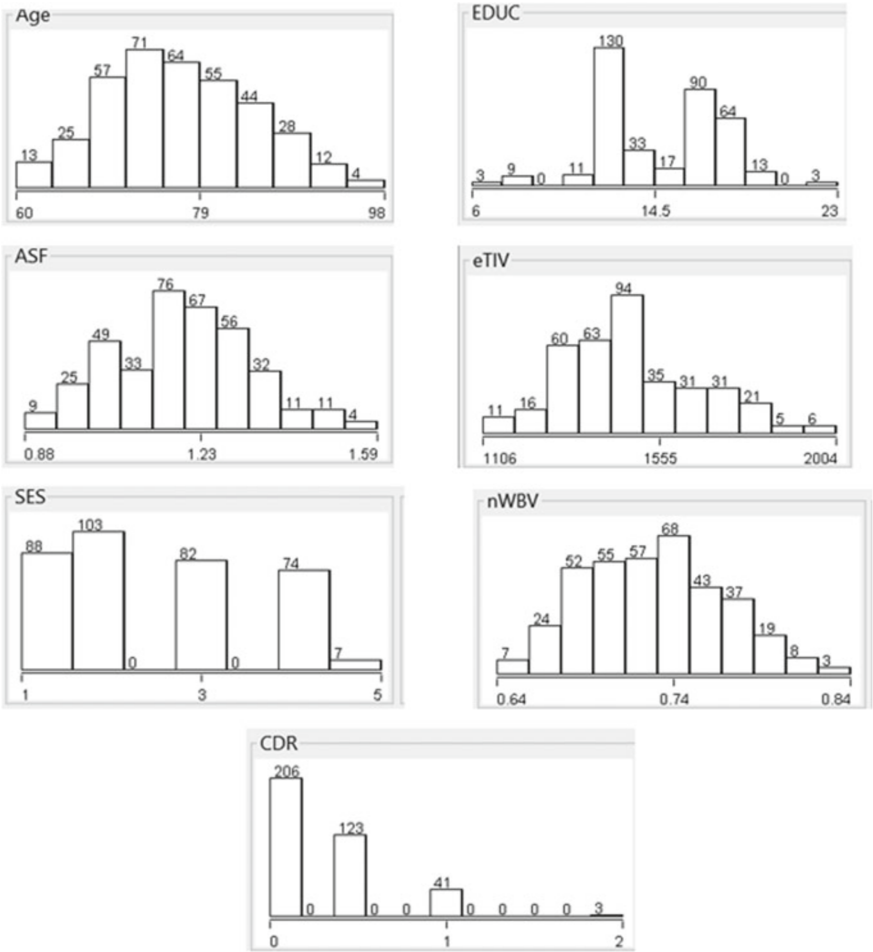


Fig. 2 Histogram of OASIS data

Table 2 The optimal model parameter

Sr. No.	Model	Parameter
1	SVM	Kernel = linear,rbf; c = 1; CV = 5
2	Decision tree	Criterion = gini,splitter = best, max_depth = none,max_leaf_node = none, min_sample_leaf = 1
3	Logistic regression	Penalty = L2,class_weight = none, random state = none
4	Random forest	N_estimator = 100, max_depth = none,min_sample_split = 2, min_sample_leaf = 1, max_leaf_nodes = none
5	AdaBoost	N_estimator = 100, cv = 3
6	XGBoost	Estimator = estimator, param_grid = paramters, scoring = roc_auc, n_jobs, CV = 100,

Table 3 Comparative analysis of distinct machine learning algorithms

Sr. No.	Model	Accuracy	Recall	Precision	F1-score
1	SVM	83.03	86.04	82.3	84.3
2	Decision tree	84.82	81.4	88.9	85.0
3	Logistic regression	69.64	74.6	69.8	72.1
4	Random forest	86.60	88.1	86.7	87.4
5	AdaBoost	91.57	89.28	78.43	72.1
6	XGBoost	93.02	89.28	91.57	81.25

of 89.28, precision of 91.57, and f1-score of 81.25, followed by AdaBoost with an accuracy of 91.57, recall of 89.28, precision of 78.43, and f1-score of 721.1. However, in the case of logistic regression, it creates more linear boundaries, and hence, classification does not yield appropriate results.

5 Conclusion and Future Scope

Since the fundamental focal point of the exploration so far has been on separating the dementia disarray, the capacity to analyze different sorts of illness has been lagging [45]. Additionally, it requires noteworthy improvements. The system's primary goal is to provide qualitative reasoning on dementia forecast. The dataset was made available on the Open Access Series of Imaging Studies project and has been used to predict dementia in older adults [50]. The missing values have been added and some extraneous characteristics involved were removed in preprocessing. For the values to be readily put in the machine learning models, standardization was done. Later, the machine learning models were trained on the dataset. Accuracy, recall, precision, and F1-score have been employed as assessment measures. XGBoost has produced the best result for deployment out of all the models. Future system models might be enhanced by using more ensemble methods and utilizing a larger dataset. It will improve the system's performance and dependability. By simply entering MRI data, a machine learning system can assist in gaining insights into the likelihood of dementia in adult patients [1]. Perhaps, it would assist patients in receiving early dementia therapy and enhance their quality of life.

References

1. Kalaria RN, Maestre GE, Arizaga R et al (2018) Alzheimer's disease and vascular dementia in developing countries: prevalence, management and risk factors. Lancet Neurol 7(9):812–826
2. Mendez MF (2017) Early onset alzheimer disease. Neurologic Clin 35:263–281
3. Jellinger KA (2018) Dementia with Lewy bodies and Parkinson's disease-dementia: current trends and controversies. J Neural Transm 125(4):615–650
4. Niessen WJ (2016) Mr brain image analysis in dementia: from quantitative imaging biomarkers to aging brain models and Imaging Genetics. Med Image Anal 33:107–113
5. Henriksen OM, Marner L, Law I (2016) Clinical pet/mr imaging in dementia and neuro-oncology. PET Clinics 11(4):441–452
6. Arab A, Wojna-Pelczar A, Khairnar A, Szabo N, RudaKucerova J (2018) Principle of diffusion kurtosis imaging and its role in early diagnosis of neurodegenerative disorders. Brain Res Bull 139:91–98
7. Martinez Murcia FJ, Ortiz A, Gorriz JM, Ramirez J, Castillo BD (2020) Studying the manifold structure of Alzheimer's disease: a deep learning approach using convolution autoencoders. IEEE J Biomed Health Inform 24:17–26. https://doi.org/10.1109/JBHI.2019.2914970
8. Osama Khalaf I, Ghaida M, Abdul SD (2020) Energy efficient routing and reliable data transmission protocol in WSN. Int J Adv Soft Comput Appl 12:45–53
9. National Academies of Science, Engineering and Medicine (2018) Preventing cognitive decline and dementia: A way forward. London: The National Academies Press
10. Tariq S, Barber PA (2018) Dementia risk and prevention by targeting modifiable vascular risk factors. J Neurochemistr 144:565–581. https://doi.org/10.1111/jnc.14132
11. Williams Jennifer A, Weakly A, Cook MS, Edgecombe DJ (2018) Machine learning techniques for diagnostic differentiation of mild cognitive impairment and dementia. In: workshops at the Twenty-Seventh AAAI conference on artificial ıntelligence. 13:9277–82
12. Caddementia: A standardized evaluation framework for computer-aided diagnosis of dementia based on structural MRI. Retrieved from: https://caddementia.grand-challenge.org/Home/. Accessed on 2018–05–26
13. Huang W, Zeng S, Li J, Chen G (2016) A new image-based immersive tool for dementia diagnosis using pairwise ranking and learning. Multimedia Tools Appl 75(9):5359–5376
14. Ishii K Pet approaches for diagnosis of dementia. AJNR Am J Neuroradiol 35(11):2030–2038
15. Ramirez J, Gorriz J, Salas-Gonzalez D, Romero A, Lopez M, Alvarez I, Gomez-Rio M Computer-aided diagnosis of dementia combining support vector machines and discriminant set of features. Inform Sci 237:59–72
16. Bron EE, Smits M, Niessen WJ, Klein S (2015) Feature selection based on the SVM weight vector for classification of dementia. IEEE J Med Biomed Health Inform 19(5):1617–1626
17. Sorensen L, Nielsen M (2018) Ensemble support vector machine classification of dementia using structural MRI and mini-mental state examination. J Neurosci Methods 302:66–74
18. Nanni L, Lumini A, Zaffonato N (2018) Ensemble based on static classifier selection for automated diagnosis of mild cognitive impairment. J Neurosci Methods 302:42–46
19. Nambiar Jyothi R, Prakash G (2018) Predictive analysis for healthcare sector using big data technology. In: Second ınternational conference on green computing and ınternet of things (ICGCIoT), IEEE

20. Liang H, Mengzi L, Ruixue W, Peixin L, Wei L*, Long L* (2018) Big data in health care: applications and challenges. Data Inform Manage 2(3):ACM 175–197K
21. Raffaele C, Marta R (2020) Artificial Intelligence and Machine Learning applications in brilliant production: progress, trends and direction. Sustainability 12:492; Pearson.https://doi.org/10.3390/su12020492
22. Guest Editorial (2016) Mining big data in biomedicine and health care. J Biomed Inform 63:400–403. https://doi.org/10.1016/j.jbi.2016.09.014
23. Rashmeet T, Inderveer C Network analysis as a computational technique and ıts benefaction for predictive analysis of healthcare data: a systematic review. Archives of Computational Methods in Engineering, Springer https://doi.org/10.1007/s11831-020-09435-z
24. Natalia A, Gennady A (2018) Designing visual analytics methods for massive collections of movement data. Cartographica Int J Geographic Inform Geo Visual 42(2):117. Retrieved from http://openaccess.city.ac.uk/2842/
25. Sunil K, Ilyoung C (2019) Correlation analysis to ıdentify the effective data in machine learning: prediction of depressive disorder and emotion States. Int J Environ Res Public Health 3(10):114–124. https://doi.org/10.3390/ijerph15122907
26. Simione M, Yi Z, Nina Z (2020) Compressive big data analytics: an ensemble meta-algorithm for high-dimensional multisource datasets. Compressive Big data analytics v2.0. Plos One. https://doi.org/10.1371/journal.pone.0228520
27. Afsaneh D, Daniella KV, Prerna C, Janine D Identifying behavioral phenotypes of loneliness and social ısolation with passive sensing: statistical analysis, data mining and machine learning of smartphone and fitbit data. JMIR JHealth UHealth 7(7). https://doi.org/10.2196/13209
28. Brian S, Yasue M, Kuo-Ching L (2020) Speech quality feature analysis for classification of depression and dementia patients. Sensors 20:3599. https://doi.org/10.3390/s20123599
29. Srividya M, Mohanavalli S, Bhalaji N (2018) Behavioral modeling for mental health using machine learning algorithms. J Med Syst 88.
30. R. Bhatnagar and G. Gohain, "Prediction Analysis Using Decision Trees and Random Forest Machine Learning Algorithms on Data from Terra (EOS AM-1) & Aqua (EOS PM-1) Satellite Data", *Studies in Computational Intelligence*, pp. 107–124, 2019. Available: https://doi.org/10.1007/978-3-030-20212-5_6 [Accessed 4 December 2019].
31. Esteban AR, David EL, Fabio C (2021) A survey of computational methods for online mental state assessment on social media. ACM Transact Comput Health 17. https://doi.org/10.1145/3437259
32. Morshedul BA, Shafayet Jamil AHM, Maliha M, Monirujjaman Khan M, Aljahdali S, Kaur M, Singh P, Masud M (2021) Comparative analysis of machine learning algorithms to predict Alzheimer's data. J Healthcare Eng 9917919. https://doi.org/10.1155/2021/9917919
33. Goldstein O, Kachuee M, Karkkainen K, Sarrafzadeh M (2020) Target-focused feature selection using uncertainty measurements in healthcare data. ACM Transact Comput Health Care 15. https://doi.org/10.1145/3383685
34. Chen Y-T, Hou C-J, Derek N, Huang M-W (2021) FMRI investigation of semantic lexical processing in healthy control and Alzheimer's disease subjects using naming task: a preliminary study. Brain Sci 11(6):718. https://doi.org/10.3390/brainsci11060718
35. Rezaei S, Moturu A, Zhao S, Prkachin KM, Hadjistavropoulos T, Taati B (2021) Unobtrusive pain monitoring in older adults with dementia using pairwise and contrastive training. IEEE J Biomed Health Inform 25(5)
36. Meng Y, Speier W, Ong M, Arnold CW (2021) HCET: hierarchical clinical embedding with topic modeling on electronic health records for predicting future depression. IEEE J Biomed Health Inform 25(4)
37. Eke CS, Jammeh E, Li X, Carroll C, Pearson S, Emmanuel teacher (2021) Early detection of Alzheimer's disease with blood plasma proteins using support vector machines. IEEE J Biomed Health Inform25(1)
38. Khan T, Jacobs PG (2021) Prediction of mild cognitive impairment using movement complexity. IEEE J Biomed Health Inform 25(1)

39. Boser BE, Guyon IM, Vapnik VN A training algorithm for optimal marginal classifiers. In: Proceedings of annual ACM workshop on computational learning theory, vol 5, pp 145–152
40. Dolph CV, Alam M, Shboul Z, Samad MS (2017) Deep learning of texture and structural features for multiclass Alzheimer's disease classification. Int Joint Conferen Neural Netw 2259–2266
41. Akhila JA, Markose C, Aneesh RP (2017) Feature extraction and classification of dementia with neural network. In: International conference on intelligent computing, instrumentation and control technologies, pp 1446–1450
42. Alam R, Anderson M, Bankole A, Lach J (2018) Inferring physical agitation in dementia using a smartwatch and sequential behavior models. In: IEEE EMBS international conference on biomedical informatics, pp 170–173
43. Ju R, Hu C, Zhou P, Li Q (2017) Early diagnosis of Alzheimer's disease based on resting state brain networks and deep learning. IEEE/ACM Transact Comput Biol Bioinform 1–1
44. Liu J, Shang S, Zheng K, Wen JR (2016) Multiview ensemble learning for dementia diagnosis from neuroimaging: an artificial neural network approach. Neurocomputing 195:112–116
45. Islam J, Zhang Y (2017) A novel deep learning based multi-class classification method for Alzheimer's disease detection using brain MRI data. In: International conference on brain ınformatics, pp 213–222
46. Nosakhare E, Picard R (2020) Toward assessing and recommending combinations of behaviors for improving health and well-being. ACM Transact Comput Healthcare 4. https://doi.org/10.1145/3368958
47. Jin Z, Cui S, Guo S, Gotz D, Sun J, Cao N (2020) CarePre: an ıntelligent clinical decision assistance system. ACM Transact Comput Healthcare 6. https://doi.org/10.1145/3344258
48. Al-Qazzazz NK, Ali SHBM, Ahmad SM, Chellappan K, Islam MS, Escudero J (2014) Role of the egg as a biomarker in the early detection and classification of dementia. Scientific World J 2014:9003068
49. Shi J, Zheng X, Li Y, Zhang Q, Ying Y (2018) Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of Alzheimer's disease. IEEE J Biomed Health Inform 22(1):173–183
50. Patel HH, Prajapati P (2018) Study and analysis of decision tree based classification algorithm. Int J Comput Sci Eng 6(10):74–78

Implementation of Vision Transformers on SPECT Heart Dataset: A Comparative Study

Poonam Verma, Vikas Tripathi, and Bhaskar Pant

Abstract Medical imaging has gone through many changes and one of the changes could be observed since the introduction of transformers and deep learning models. Transformers have complicated architecture that can help in carrying out complex tasks on medical images or data. This research paper presents a comprehensive study on the implementation of the Vision Transformers (ViTs) on the Single-Photon Emission Computed Tomography (SPECT) heart dataset. The basic objective is to evaluate the accuracy of the classification of abnormal cardiac conditions from SPECT images using ViT. Further, this paper tries to explore different aspects of the architecture of transformers and their performance in comparison to the traditional machine learning models in analyzing the SPECT heart dataset. It helps to contribute to the growing community of research carried out in medical images and data using new technologies.

Keywords ViT · SPECT heart dataset · Image classification · Deep learning

1 Introduction

SPECT is a medical imaging technique that makes use of gamma rays to develop 3D images of the internal organs and tissues in the body [1]. SPECT works basically by injecting a small dose of radioactive material, called a radiotracer, into the patient's bloodstream [2]. The radiotracer emits gamma rays as it moves across the body, and it is detected by the gamma camera positioned outside the body [3]. The data collected from the gamma camera is then used to create detailed images of the patient's internal organs and tissues [4]. SPECT is commonly used for the diagnosis and treatment of a variety of medical conditions, ranging from heart disease to cancer

P. Verma (✉) · V. Tripathi · B. Pant
Computer Science and Engineering, Graphic Era University Deemed, Dehradun, India
e-mail: poonamddn2020@gmail.com

P. Verma
School of Computing, Graphic Era Hill University, Dehradun, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_10

and neurological disorders. Some of the challenges that are witnessed with the SPECT images are that many times the image quality gets affected due to the factors such as patient motion, photon attenuation, and scatter [5]. Further, SPECT imaging will cause radiation exposure for the patients. It also suffers from lower spatial resolution than the other traditional methods such as magnetic resonance imaging (MRI) and computed tomography (CT), which can make it difficult to distinguish between the small details in the images, and there is also a probability of overlapping between normal and abnormal tissues [6]. This paper is proposed to carry out a comparative study of ViT on the SPECT heart dataset with the other baseline models of the machine learning and deep learning techniques. The remaining organization of the paper is prepared as follows. The survey of the literature related to the SPECT heart dataset and ViT is presented in Section II. In Section III, the proposed methodology is explained in detail. Section IV describes the performance metrics of the proposed methodology. Section V concludes the work and describes the future scope of the paper.

2 Literature Review

Different Image Processing techniques are used to enhance the quality of the SPECT image dataset with reduced noise. Preprocessing techniques are used to remove the noise from the data by implementing filtering, thresholding, and segmentation. Many popular Image Registration techniques such as landmark-based, spatial-based, and feature-based have been implemented on the SPECT images [7]. Segmentation techniques have been used to extract the features from SPECT images by partitioning the images into the Regions of Interest (ROI) [8]. More features are extracted to derive meaningful information by making use of popular techniques such as Fourier Transform and Wavelet Transform methods [9]. Once the information has been derived, it can be classified into different classes by implementing conventional machine learning techniques such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs). Once the classification has been carried out successfully, different visualization techniques are used to display the important features of the SPECT images [10]. Statistical techniques such as MANOVA have been used by some authors to identify the abnormal patterns from the given images and help to distinguish the different diseases and healthy tissues [11]. Some authors implemented an ensemble model of Principal Component Analysis (PCA) to reduce the dataset dimension and then to carry out hierarchical clustering to cluster a group of patients that suffer from cognitive impairment issues [12]. Few authors have also made use of a Bayesian-based hierarchical machine learning model to classify the patients suffering from epilepsy and create a customized treatment for them [13]. Different innovative imaging machine learning and visualization models are optimized using different simulation techniques on the images. SPECT images can be simulated using various techniques that can help to evaluate the performance of the image reconstruction algorithms, optimizing the imaging protocols, and also help in the development of the new radiotracers [14]. Popular Monte Carlo simulation can help to model different physical effects of radiotracers such as photon attenuation, scatter, and collimator response [15]. Researchers have utilized many analytical methods on SPECT to evaluate the performance of the imaging systems [16]. Phantoms are used to study the effect of imaging parameters such as collimator design, energy window selection, and reconstruction algorithm on the quality of the SPECT images. Hybrid methods combining Monte Carlo simulation with statistical analytical methods help to maintain the performance speed as well as the accuracy.

Some researchers were able to distinguish between benign and malignant breast tumors which were analyzing SPECT data using the machine learning algorithm of Support Vector Machine (SVM) [17]. Further, Alzheimer's cognitive issues were classified as mild cognitive impairment using SPECT data by implementing 3D CNN [18]. LSTM-based RNN was also used by Ren et al. (2020) to predict the probability of survival among hepatocellular carcinoma patients using SPECT data [19]. Implementation of deep autoencoder for detection of the myocardial infraction was also successful on SPECT data [20]. The above researches clearly describe that machine learning and deep learning techniques have been successfully implemented on the SPECT data for different disease diagnoses, treatment plannings, and outcome predictions [21].

The authors have developed an innovative YoloV5 pre-trained architecture for the purpose of extracting the Region of Interest (ROI) from images that address the constraints posed by previous architectures by introducing Vision Transformers for binary image classification, trained and validated on a public dataset known as UTA-RLDD achieving 96.2 and 97.4% during training and validation, respectively. Research proposes a novel multi-class prediction framework for skin lesions' classification, based on ViT and ViT-GAN, which leverages Vision Transformers-based Generative Adversarial Networks (GANs) to tackle the class imbalance [22].

3 Proposed Methodology

3.1 Description of the SPECT Heart Dataset

The Single--Photon Emission Computed Tomography (SPECT) heart dataset is a popular dataset comprising 267 SPECT images, where each image has 23 × 23 pixels and is labeled as either "normal" or "abnormal". For each patient, a pattern of 44 continuous features was created which was processed to generate a binary pattern comprising 22 features. CLIP3 algorithm was used to generate rules that were 84% accurate when compared with the cardiologist's diagnosis [23].

3.2 Architecture of Vision Transformers

ViT architecture has been recently very popular. This architecture makes use of long-range dependencies between the image patches and utilizes it for classification purposes. The ViT architecture was designed for images represented as 2D arrays of pixels. However, the present advancement has made it possible to reframe the architecture so that categorical data can be also analyzed using ViT. One of the easiest methods is to represent the complete categorical data as one-hot vectors. Now this data can be fed to the ViT architecture posing as if the categorical data is in the format of the image patches. The ViT architecture will extract the relevant features from this data and helps to carry out the classification of the categorical data. Hence, it can be assumed that ViT can be also used to classify categorical data of the SPECT heart dataset and help to identify abnormalities in the heart.

The ViT architecture has transformer layers. There are two sub-layers for each defined transformer layer comprising a self-attention mechanism and neural network of feedforward mode. The sub-layer of the self-attention is multi-head which permits the model to understand the input sequences provided. As described above, since the element in each vector has a unique category, it calculates an attention score for it in proper sequence. These attention scores are used like weights for the given input.

ViT architecture has a feedforward network that uses ReLU activation between the two linear layers. It operates independently on each position in the sequence, transforming the information learned by the self-attention mechanism. The architecture has sub-layers, residual connections, and layer normalizations. This can help to learn the features more easily by the models and also help to deal with the vanishing gradient problem. Based on the type of task to be completed, the number of layers required in the ViT architecture can be decided. Like for example, ViT-B/16 will have 12 layers.

To apply ViT to the SPECT heart dataset, the data was preprocessed and normalized to make them compatible with the model as shown in Fig. 1. We have used a pre-trained ViT model and it was first fine-tuned on the available SPECT heart dataset. During this process, some of the lower layers were frozen which was used to understand or extract the important features and these features were used by the higher layers for training. For SPECT heart dataset, the model was fine-tuned on the labeled images, and its weights are adjusted to improve the performance of the model on the specific task.

Since we have discussed about the Self-Attention Mechanism of ViT, so to understand the concept, let us delve into it's representation. In order to represent the mechanism, we will consider the Query as Q, Key as K, and Value as V.

$$Q = XWQ, K = XWK, V = XWV. \tag{1}$$

We have made use of the Attention Scores (A) and output (O) for the transformer layers that can be written as:

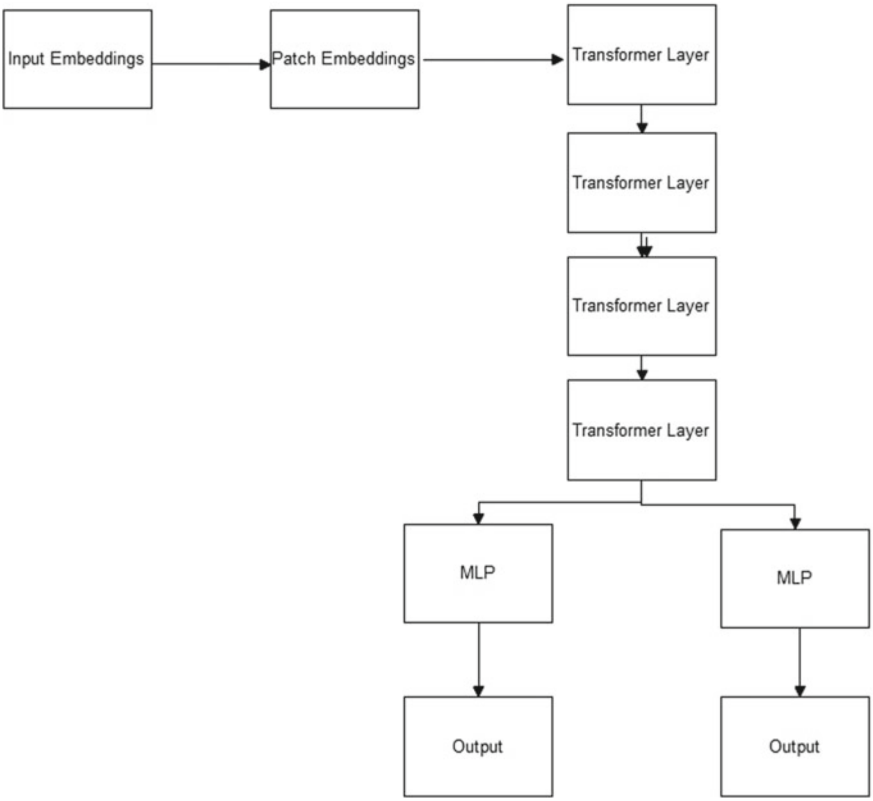


Fig. 1 Preprocessing, training setup, and hyperparameter selection

$$A = softmax(QKT \ / \ sqrt(d\_k)) \tag{2}$$

and

$$O = AV. \tag{3}$$

As shown in Fig. 1, the input given to the ViT is sent in the format of the image patches using patching process. Further, the patch *X* is split and expressed as P.

$$P = [p1, p2, ..., pn] = \text{Split}(X). \tag{4}$$

And embedding the patches is expressed as *E*.

$$E = [e1, e2, ..., en] = MLP(P). \tag{5}$$

Since the position-wise feedforward neural network is implemented, hence the positional matrix is calculated using Eq. (6) which can be added to the patch embedding E and represented as shown in Eq. (7).

Positional encoding matrix:

$$PE = [pe1, pe2, ..., pen]. \tag{6}$$

On addition of positional encoding to the patch embeddings, the positional encoding matrix can result into

$$PEE = E + PE. \tag{7}$$

Since the transformer has many layers, it can be described using the following equations from (8–13):

$$FFN = MLP(PEE), \tag{8}$$

$$LN1 = \text{Layer Norm}(PEE + FFN), \tag{9}$$

$$MHA = \text{Multi Head Attention}(LN1), \tag{10}$$

$$LN2 = \text{Layer Norm}(LN1 + MHA), \tag{11}$$

$$FFN2 = \text{MLP}(LN2), \tag{12}$$

$$LN3 = \text{Layer Norm}(LN2 + FFN2). \tag{13}$$

We finally achieve the output when the embedded token is processed through *MLP* and results into *Y* as shown from Eqs. 14–16:

$$C = MLP(class\_token), \tag{14}$$

$$CI = LN3 + C, \tag{15}$$

$$Y \ = \ softmax(MLP(CI)). \tag{16}$$

Since the SPECT heart dataset comprises categorical data, it is converted into one-hot encoders where each vector belongs to a unique category. Each category based on the self-attention mechanism has weights. As shown in Fig. 1, the SPECT heart dataset is fed into the ViT model as input features. The input data is represented as the 2D arrays of vectors which is transformed into the patch embeddings making use of a patching process by the layers. Finally, the output is also obtained when the data is passed through a Multi-layer Perceptron (MLP).

4 Description of Other Baseline Models for Comparison

As shown in Table 1, the performance analysis of ViT with conventional machine learning models has been carried out. Implementation of Naïve Bayes involves the analysis of the SPECT heart dataset, which assumes that each symptom's presence is independent of others. This assumption allows for a simplified calculation of probabilities. Naive Bayes is particularly useful in situations where feature independence is a close approximation of reality and also works well with limited training samples. Decision trees are also popularly used as the classification models that classify the given data using entropy or Information Gain. In our specific context, the SPECT heart dataset is analyzed by decision trees, which meticulously select the most informative symptoms to split the data into subsets. The algorithm evaluates each internal node to determine which feature produces the best separation of classes. The maximum depth considered is 4. These values ensure that the tree does not become too deep and avoids splitting on nodes with few samples. TensorFlow has enabled the successful implementation of our deep learning model. The neural network used has two hidden layers and each of the layer comprises 64 and 32 neurons, respectively. Since the SPECT heart dataset has two outputs, so in order to get a binary classification, we have made use of sigmoid activation function. Further, the model was optimized by making use of an Adam Optimizer and the loss function used in the layers was binary cross-entropy. We have considered a batch_size of 16 and the loop was carried out for 20 epochs. All the above-explained models were trained using 50 epochs. Further early stopping was also carried out on the validation loss.

Table 1 Comparison analysis of ViT with baseline models and state-of-the-art models

Model	Accuracy	Classification error	Precision	F-measure	Sensitivity	Specificity
Naïve Bayes	74.3	22.6	89.5	83.6	78.7	62
Decision tree	83	13.0	87.2	90.8	94.6	0
DL fine-tuned	86	12.1	90.1	90.3	91.7	35
A-EFO-CRNN [24]	75	20	–	–	–	–
RF hybrid method [25]	86.70	10.70	–	–	–	–
ViT	89.53	9.98	90.6	89.9	93.5	78

The performance of the baseline models was evaluated basically using accuracy metrics. We conducted five-fold cross-validation and reported the average performance across the folds.

Table 1 summarizes the results obtained from training and evaluating the baseline models on the SPECT heart dataset. The ViT model was able to outperform in comparison to other conventional machine learning models; thereby, it has proved the effectiveness in extracting relevant features from the SPECT heart dataset. Overall, the baseline models provided a foundation for performance comparison and demonstrated the potential of different machine learning techniques in analyzing the SPECT heart dataset.

5 Discussion

The basic advantage of Vision Transformers is to capture the global context by considering the image patches that permit to model the long-range dependencies and capture the high-level semantic information for image classification object detection. Further ViT can be used to process images of arbitrary sizes. ViTs offer to understand the model by using the attention mechanism. Further, the ViT mode has good transfer-learning capabilities. One of the limitations that ViT architecture has is, it requires a large amount of labeled database for training purposes. Moreover, it has a tendency to lose minute spatial information, as it processes the data in patches. The performance of ViTs can be sensitive to the choice of patch size and its positional encoding.

SPECT heart datasets may have limited sample sizes and annotations, posing challenges for training deep learning models. Incorporating ViTs into medical imaging and diagnosis workflows for SPECT heart images holds significant implications for improving accuracy, efficiency, interpretability, and personalized care. However, it becomes necessary that validation and integration with existing models used with medical data are also researched to understand the possible potential that transformers have in the field of cardiology.

References

1. Elangovan A, Jeyaseelan T (2016) Medical imaging modalities: a survey. In: 2016 International conference on emerging trends in engineering, technology and science (ICETETS). https://doi.org/10.1109/icetets.2016.7603066
2. Shaikh FA, Kurtys E, Kubassova O, Roettger D (2020) Reporter gene imaging and its role in imaging-based drug development. Drug Discov Today 25:582–592. https://doi.org/10.1016/j.drudis.2019.12.010
3. Spanoudaki VC, Ziegler SI (2008) Pet & SPECT instrumentation. Mol Imaging I:53–74.https://doi.org/10.1007/978-3-540-72718-7_3
4. Coura-Filho GB, Torres Silva de Oliveira M, Morais de Campos AL (2022) Basic principles of scintigraphy and SPECT (single-photon emission computed tomography). Nuclear Med Endocrine Disord 9–14. https://doi.org/10.1007/978-3-031-13224-7_2

5. Rahmim A, Zaidi H (2008) Pet versus SPECT: strengths, limitations and challenges. Nucl Med Commun 29:193–207. https://doi.org/10.1097/mnm.0b013e3282f3a515
6. Kumar V, Gu Y, Basu S et al (2012) Radiomics: the process and the challenges. Magn Reson Imaging 30:1234–1248. https://doi.org/10.1016/j.mri.2012.06.010
7. Oliveira FPM, Tavares JM (2012) Medical image registration: a Review. Comput Methods Biomech Biomed Eng 17:73–93. https://doi.org/10.1080/10255842.2012.670855
8. Rogowska J (2000) Overview and fundamentals of Medical Image segmentation. Handbook Med Imaging 69–85.https://doi.org/10.1016/b978-012077790-7/50009-6
9. Nabti M, Bouridane A (2008) An effective and fast iris recognition system based on a combined multiscale feature extraction technique. Pattern Recogn 41:868–879. https://doi.org/10.1016/j.patcog.2007.06.030
10. Fryback DG, Thornbury JR (1991) The efficacy of diagnostic imaging. Med Dec Mak 11:88–94. https://doi.org/10.1177/0272989x9101100203
11. Graff BJ, Harrison SL, Payne SJ, El-Bouri WK (2022) Regional cerebral blood flow changes in healthy ageing and alzheimer's disease: a narrative review. Cerebrovasc Dis 52:11–20. https://doi.org/10.1159/000524797
12. Khachnaoui H, Khlifa N, Mabrouk R (2022) Machine learning for early parkinson's disease identification within Swedd group using clinical and DaTSCAN SPECT imaging features. J Imag 8:97. https://doi.org/10.3390/jimaging8040097
13. Olaniyan OT, Adetunji CO, Dare A et al (2023) Cognitive therapy for brain diseases using artificial intelligence models. Artific Intell Neurolog Disord 185–207.https://doi.org/10.1016/b978-0-323-90277-9.00013-4
14. Llosá G, Rafecas M (2023) Hybrid PET/compton-camera imaging: An imager for the next generation. Euro Phys J Plus. https://doi.org/10.1140/epjp/s13360-023-03805-9
15. Saed M, Sadremomtaz A, Mahani H (2022) Design and optimization of a breast-dedicated SPECT scanner with multi-lofthole collimation. J Instrum. https://doi.org/10.1088/1748-0221/17/01/p01006
16. Auer B, Könik A, Fromme TJ et al (2023) Mesh modeling of system geometry and Anatomy Phantoms for realistic gate simulations and their inclusion in SPECT reconstruction. Phys Med Biol 68:075015. https://doi.org/10.1088/1361-6560/acbde2
17. Ma X, He Y, Lin Q, et al (2023) Fine-grained classification of bone scintigrams by using Radiomics features. In: 2023 3rd international conference on neural networks, information and communication engineering (NNICE). https://doi.org/10.1109/nnice58320.2023.10105690
18. Lien W-C, Yeh C-H, Chang C-Y et al (2023) Convolutional Neural Networks to classify alzheimer's disease severity based on SPECT images: a comparative study. J Clin Med 12:2218. https://doi.org/10.3390/jcm12062218
19. Hosseini MS, Ehteshami Bejnordi B, Trinh VQH, Hasan D, Li X, Ki T, Zhang H et al (2023) Computational pathology: a survey review and the way forward. arXiv:2304.05482
20. Kalou Y, Al-Khani AM, Haider KH (2023) Bone marrow mesenchymal stem cells for heart failure treatment: a systematic review and meta-analysis. Heart Lung Circ 32:870–880. https://doi.org/10.1016/j.hlc.2023.01.012
21. Krishna GS, Supriya K, Vardhan J (2022) Vision transformers and YoloV5-based driver drowsiness detection framework. arXiv preprint arXiv:2209.01401
22. Krishna GS, Supriya K, MalSorgile M (2023) LesionAid: vision transformers-based skin lesion generation and classification. arXiv:2302.01104
23. Spect heart. In: UCI machine learning repository. https://doi.org/10.24432/C5P304.
24. Chamundeshwari N, Biradar NU (2022) Hybrid pattern extraction with deep learning-based heart disease diagnosis using echocardiogram images. Int J Image Graph. https://doi.org/10.1142/s0219467823500249
25. Kishor A, Chakraborty C (2021) Artificial Intelligence and internet of things based healthcare 4.0 monitoring system. Wireless Pers Commun 127:1615–1631. https://doi.org/10.1007/s11277-021-08708-5

CSR U-Net: A Novel Approach for Enhanced Skin Cancer Lesion Image Segmentation

V. Chakkarapani and S. Poornapushpakala

Abstract Early detection is very critical step in skin cancer diagnosis and treatment. This paper introduces a novel deep learning approach for skin cancer lesion image segmentation model, CSR U-Net: Channel–Spatial Regularized U-Net. The proposed model focuses on both channel attention and spatial attention with additional optimized regularization methods to prevent model overfitting. The paper discusses the implementation of U-Net, Attention U-Net, Residual U-Net models, and CSR U-Net and also compares the results. The segmentation task often has challenges due to variations in skin tones, quality of the image, variations in the lesion, noise, class imbalance, and boundary delineation. This research aims to create a better high performing model CSR U-Net that over comes the above-said challenges.

Keywords Skin cancer · Image segmentation · Deep learning · Channel–Spatial Regularized U-Net (CSR U-Net) · ISIC-2018 dataset

1 Introduction

Skin cancer is one of the very common cancers worldwide. Early detection and diagnosis are very critical in improving patient outcomes. Image segmentation plays a crucial role for skin cancer detection. In majority of the skin cancer detection process, the skin lesion images usually undergo a segmentation process to gain better classification results. This is necessary for the given complexity nature of the skin lesion images by different variations in the skin tones and irregular patterns [1]. In particular, darker skin tone images add up still more challenges as this affects the visibility of the lesion.

V. Chakkarapani (✉) · S. Poornapushpakala
Sathyabama Institute of Science and Technology (Deemed to be University), Chennai, Tamil Nadu 600119, India
e-mail: chakkarapani.ai@gmail.com

S. Poornapushpakala
e-mail: poornapushpakala.enc@sathyabama.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_11

The quality of images used for the segmentation impacts the accuracy of the models. Skin lesions vary widely in terms of size, shape, color, texture which add complexity to the segmentation tasks [2]. Additionally, skin also contains noise, hairs, bubbles, and some marks that interfere with the accuracy of the model. Most clinical images are significantly imbalanced, and this again introduces a certain level of challenges of bias in the model toward the majority class. Identifying the exact boundaries from the skin lesions is not so easy, most of the lesions are irregular, and it is tough to exactly identify where it starts and where it ends [3]. Given the complex nature of the skin lesion images, many researchers focus on building diverse datasets (benchmark datasets) that represent different populations and store them by different modalities and made it available through the ISIC archive. This enables researcher to focus on more complex problems [4].

This paper introduces a novel deep learning model, Channel–Spatial Regularized U-Net (CSR U-Net) more specifically for skin cancer lesion image segmentation. The proposed model uses attention mechanism with the unique combination of channel-wise attention and spatial attention with regularization that prevents overfitting of the model. This approach aims to address all the challenges pertaining to image quality, lesion variations, noise, skin tone, class imbalance, and boundary irregularities. For evaluating the model, we utilized the ISIC-2018 dataset, a recognized benchmark dataset for skin lesion analysis.

The main objective is to provide an efficient method for skin cancer lesion segmentation that can help with improved early detections for patients with skin cancer.

2 Literature Review

In recent years, there are extensive research which have been carried out by applying deep learning models for medical image segmentation tasks to prove improved performance. These models have shown promising results in various biomedical image segmentation tasks, and this includes skin cancer lesion image segmentation [5, 6]. This section reviews some of the recent advancements in this area of research and discusses their relevance to the proposed Channel–Spatial Regularized U-Net (CSR U-Net) model.

The U-Net model, introduced by Ronneberger et al., proves its ability to capture all the relevant details by retaining the contextual information for biomedical images [7]. The model's architecture consists of a contracting path to capture context and a symmetric expanding path to enable precise localization. This has proven effective in various segmentation tasks. Subsequent research on the U-Net architecture incorporates attention mechanism with the U-Net architecture (Attention U-Net) that specifically focuses on specific regions of the image. This mechanism improves the model's performance by enabling it to focus on relevant features and ignore irrelevant ones. On the other hand, Residual U-Net model incorporates residual connections into the U-Net architecture. This is introduced to address the vanishing gradient problem, which can affect the training of deep neural networks [7]. These residual connections allow the model to learn more complex representations and improve its segmentation performance. Recent research has introduced several innovative deep learning models for image segmentation. Another model, the mobile anti-aliasing Attention U-Net model (MAAU), introduced by Phuong Thi Le et al. [8] features both encoder and decoder paths and has fewer parameters while outperforming various state-of-the-art segmentation methods. The Residual-Dense-Attention (RDA) U-Net model, introduced by Ming-Chan Lee et al. [9], integrates residual connections, dense connections, and attention mechanisms, achieving superior performance in bladder cancer segmentation. The study introduces an automatic segmentation method employing semantic segmentation and the U-Net framework for MS cerebral lesion detection. Using various MRI slice orientations during training, the model, trained with Dice loss, demonstrated a favorable balance between automated and manual lesion masks [10]. The traditional U-Net model in deep learning has limitations in image segmentation due to its reliance on stacked local operators. This study introduces a multi-attentional U-Net, proving that attention mechanism provides better performance for medical images. The models exhibited enhanced performance and efficiency with fewer parameters [11].

The proposed CSR U-Net model combines the strengths of these models and introduces a unique combination of channel-wise attention and spatial-wise attention with regularization to prevent overfitting. This innovative approach aims to address the current challenges in skin cancer lesion image segmentation, including issues related to image quality, lesion variability, noise and artifacts, class imbalance, and boundary delineation.

In conclusion, the literature reveals a trend toward the integration of attention mechanisms and residual connections in deep learning models for image segmentation. The proposed CSR U-Net model aligns with this trend and introduces novel features that aim to improve the accuracy and efficiency of skin cancer lesion image segmentation.

3 Methodology

This section details the methodology used in this research, focusing on the implementation of the U-Net, Attention U-Net, Residual U-Net, and the proposed Channel–Spatial Regularized U-Net (CSR U-Net) models for skin cancer lesion image segmentation.

3.1 Dataset

For this study, ISIC-2018 dataset [12] was used. This is one of the widely used benchmarks for skin lesion analysis toward melanoma detection. This dataset includes dermoscopic images and its corresponding masks of various skin lesions. This dataset is a diverse and challenging dataset that can be used by many researchers for evaluation of the models. This dataset contains 2594 in total out of which randomly selected 80% of the image and mask pairs are used for training and 20% is used for validation during the training phase.

3.2 Preprocessing

For this study, the images were preprocessed to standardize the input for the models. This preprocessing included resizing the images to a uniform size, normalizing the pixel values and noise removal techniques before taking it to the model training process.

3.3 Model Implementation

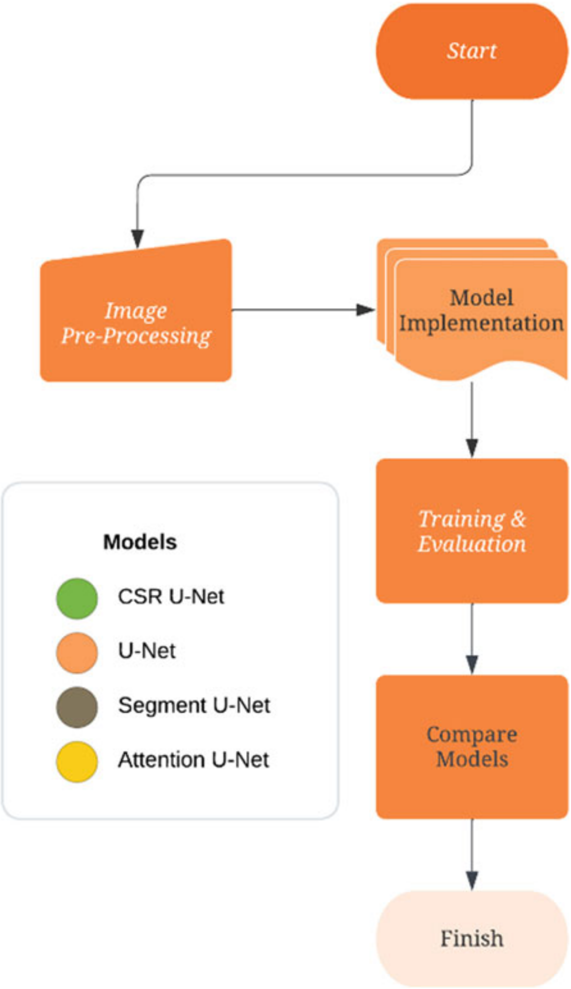
In this study, four models were implemented: U-Net, Attention U-Net, Residual U-Net, and the proposed CSR U-Net. Figure 1 shows the high-level approach for training and evaluating models. The U-Net model was implemented as per the original architecture proposed by Ronneberger et al. [7] with a contracting path to capture context and a symmetric expanding path for precise localization. The Attention U-Net model was implemented based on the U-Net architecture with the addition of an attention mechanism. The Residual U-Net model was implemented by incorporating residual connections into the U-Net architecture. In this novel approach CSR U-Net model, was implemented with channel-wise attention and spatial attention with optimized regularization techniques to prevent overfitting.

Figure 2 shows CSR Block introduced to the U-Net architecture that shows the channel attention and spatial attention blocks. Figure 3 shows how the proposed CSR Block is introduced in the U-Net architecture. In this novel architecture, these blocks are applied after each convolutional layer in the contracting path and concatenated with the corresponding layer in the expanding path. This allows the model to selectively emphasize relevant information and suppress noise. This addition of attention blocks helps to improve the model's segmentation performance by capturing more meaningful features.

4 Training and Evaluation

The models were trained using the training subset of the ISIC-2018 dataset, with the validation subset used for tuning the model parameters. The performance of the models was evaluated on the test subset using segmentation specific metrics.

Fig. 1 High-level approach for training and evaluating models



In this study, all deep learning models were trained and evaluated on a high-performance computing system equipped with a Tesla P100 GPU. The Tesla P100 is a full-sized data center GPU, based on the NVIDIA Pascal architecture, and is designed to deliver maximum throughput for deep learning and high-performance computing (HPC) workloads. It provides a peak performance of 4.7 TeraFLOPS for double-precision, 9.3 TeraFLOPS for single-precision, and 18.7 TeraFLOPS for half-precision computations. The GPU has a memory capacity of 16GB HBM2, which allows for the efficient handling of large datasets and complex computational tasks.

The use of the Tesla P100 GPU significantly accelerated the training and inference times of our models, enabling us to iterate quickly over different model architectures and hyperparameters. All models were implemented using Python programming language with the aid of deep learning libraries such as TensorFlow and Keras. The

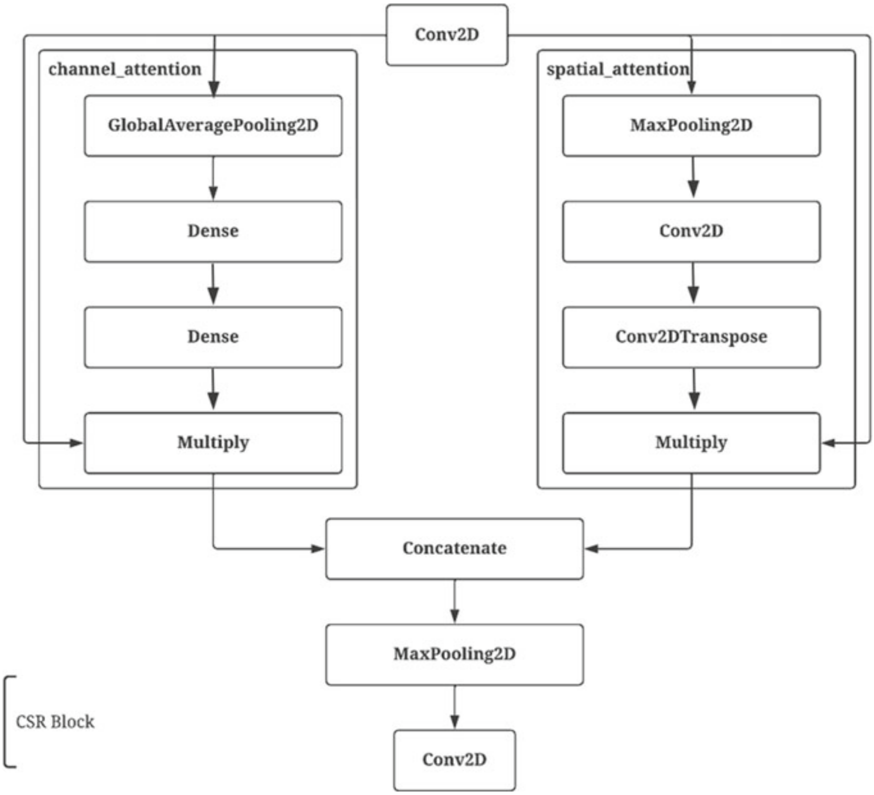


Fig. 2 CSR block with channel attention and spatial attention blocks

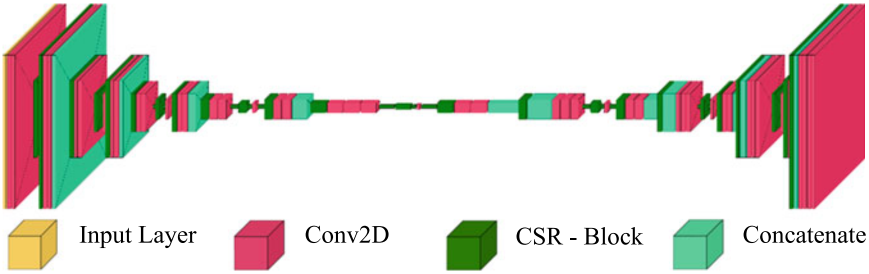


Fig. 3 Proposed CSR block in U-Net architecture

code was optimized to fully utilize the GPU's capabilities, ensuring efficient memory usage and parallel computation.

4.1 Evaluation Metrics

In image segmentation, the choice of evaluation metrics is crucial as it directly influences on the performance of the model. For this study, we have chosen: Accuracy, Intersection over Union (IoU), Dice Score, and Pixel Accuracy. Each of these metrics provides a unique perspective on the performance and, when considered together, they offer a comprehensive evaluation of the model's ability in medical image segmentation.

Accuracy

Accuracy in machine learning denotes the fraction of correct predictions among total cases. For image segmentation, it signifies the percentage of pixels correctly classified. However, its effectiveness diminishes in the face of class imbalances, common in medical image segmentation.

Intersection over Union (IoU)

Also termed the Jaccard index, IoU evaluates image segmentation by measuring the overlap between predicted and actual results. It is the ratio of their intersection to their union, emphasizing the quality of overlap in segmentation tasks. Higher the IoU score indicates better the accuracy of the segmentation model. This metric is particularly used in applications like object detection, medical imaging, autonomous vehicles, etc.

Dice Score

The Dice Score quantifies image segmentation performance by weighting the intersection area twice. It is computed as twice the intersection of predicted and actual regions divided by their combined area. Especially valuable for smaller regions of interest, it is prevalent in medical image analysis.

Pixel Accuracy

Pixel Accuracy assesses segmentation by the percentage of pixels accurately classified, focusing only on the positive class, such as lesions. It is a pertinent metric when the positive class, often in the minority, is of primary concern.

These metrics were chosen for their relevance to the task of image segmentation and their ability to provide a comprehensive evaluation of the models. By considering these metrics together, we can assess not only the quantity of correct predictions (as given by Accuracy and Pixel Accuracy) but also the quality of the segmentation (as given by IoU and Dice Score). This comprehensive evaluation allows us to accurately assess the performance of the models and identify the most effective model for skin cancer lesion image segmentation.

The results of the evaluation were then compared to assess the performance of the models and the effectiveness of the proposed CSR U-Net model in addressing the challenges in skin cancer lesion image segmentation.

4.2 Results and Discussion

In this study, we trained and evaluated four deep learning models—U-Net, Attention U-Net, Residual U-Net, and Channel–Spatial Regularized U-Net (CSR U-Net)—on the ISIC-2018 dataset, a comprehensive collection of dermoscopic images of skin lesions. Our goal was to assess the performance of these models in skin lesion segmentation tasks and to identify the model that delivers the most accurate and reliable results. The performance of the models was assessed using four key metrics: Accuracy, Intersection over Union (IoU), Dice Score, and Pixel Accuracy.

The U-Net model, as a baseline, demonstrated solid performance with an accuracy of 93.23%, an IoU of 97.96%, a Dice Score of 95.17%, and a Pixel Accuracy of 87.76%. These results reflect the robust architecture of U-Net and the effectiveness of its contracting and expanding paths in capturing context and enabling precise localization.

The Attention U-Net model, with its ability to focus on specific regions of the image, showed improved performance with an accuracy of 94.36%, an IoU of 98.96%, a Dice Score of 96.75%, and a Pixel Accuracy of 94.39%. These results highlight the model's ability to capture relevant features and ignore irrelevant ones.

The Residual U-Net model, with its residual connections, achieved an accuracy of 94.08%, an IoU of 98.97%, a Dice Score of 96.10%, and a Pixel Accuracy of 91.82%. These results indicate that the model was able to learn more complex representations and address the vanishing gradient problem, leading to improved segmentation performance.

The proposed CSR U-Net model demonstrated superior performance with an accuracy of 97.57%, an IoU of 99.96%, a Dice Score of 97.43%, and a Pixel Accuracy of 96.04%. The model's channel-wise attention allowed it to focus on the most informative features across the channels, while the spatial-wise attention enabled it to focus on the most relevant regions in the spatial domain. The regularization has helped to prevent overfitting, improving the model's generalization ability.

The CSR U-Net model's superior performance can be attributed to its ability to effectively capture both local and global contextual information in the images. The channel attention mechanism allows the model to focus on the most informative feature channels, while the spatial attention mechanism enables the model to concentrate on the most relevant spatial locations. This dual attention mechanism, combined with the U-Net's powerful segmentation capabilities, allows the CSR U-Net model to deliver highly accurate and precise segmentation results. Figures 4 and 5 show sample skin lesion images, its corresponding ground truth images, and the segmentation outputs of different models implemented as per this study.

Furthermore, the CSR U-Net model's robust performance across all four metrics—Accuracy, IoU, Dice Score, and Pixel Accuracy—demonstrates its consistency and reliability in skin lesion segmentation tasks. This consistency is crucial in medical image analysis, where reliable and repeatable results are of paramount importance. Figures 6 and 7 show the training loss, validation loss, training accuracy, and validation accuracy of all segmentation models implemented as per this study.

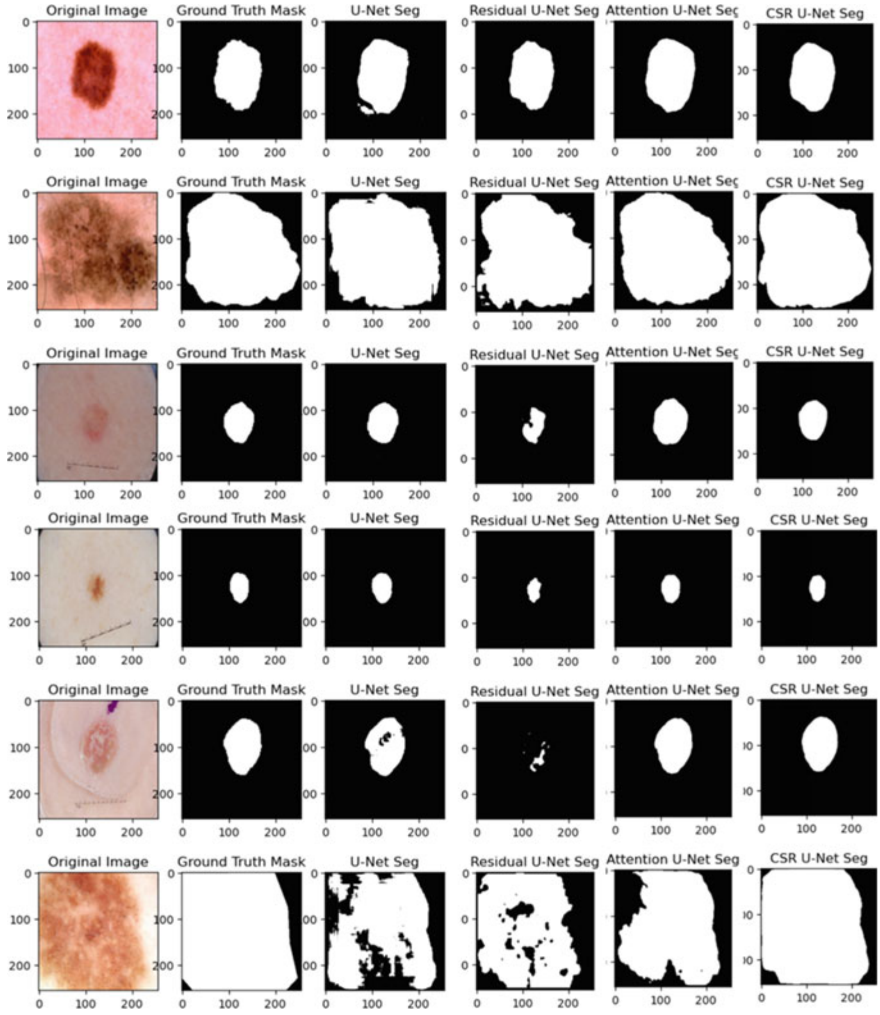


Fig. 4 Sample 1: Sample skin lesion images, its corresponding ground truth images and the segmentation outputs of different models U-Net, Residual U-Net, Attention U-Net, and CSR U-Net (proposed model architecture)

Table 1 and Fig. 8 show the comparison of the metrics Accuracy, IoU, Dice Score, and Pixel Accuracy of the segmentation models U-Net, Residual U-Net, Attention U-Net, and CSR U-Net (proposed model architecture). The results highlight the effectiveness of the proposed CSR U-Net model in addressing the challenges in skin cancer lesion image segmentation, including issues related to image quality, lesion variability, noise and artifacts, class imbalance, and boundary delineation.

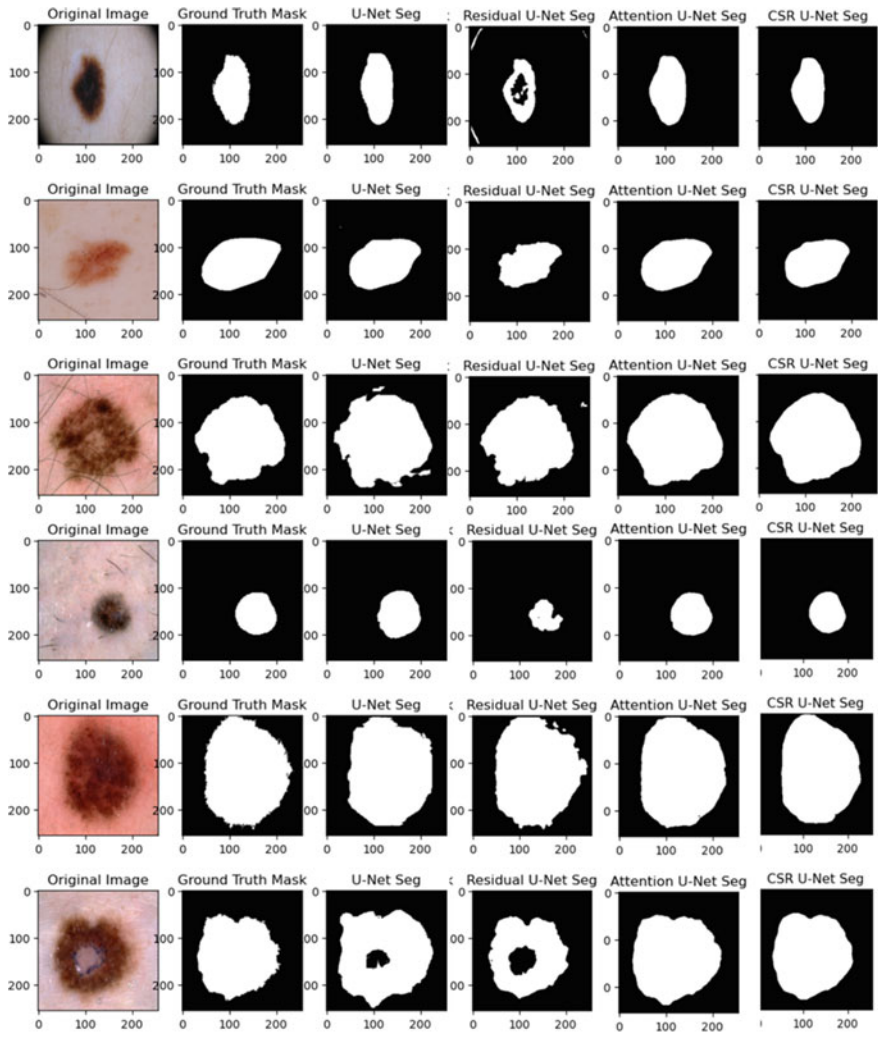


Fig. 5 Sample 2: Sample skin lesion images, its corresponding ground truth images, and the segmentation outputs of different models U-Net, Residual U-Net, Attention U-Net, and CSR U-Net (proposed model architecture)

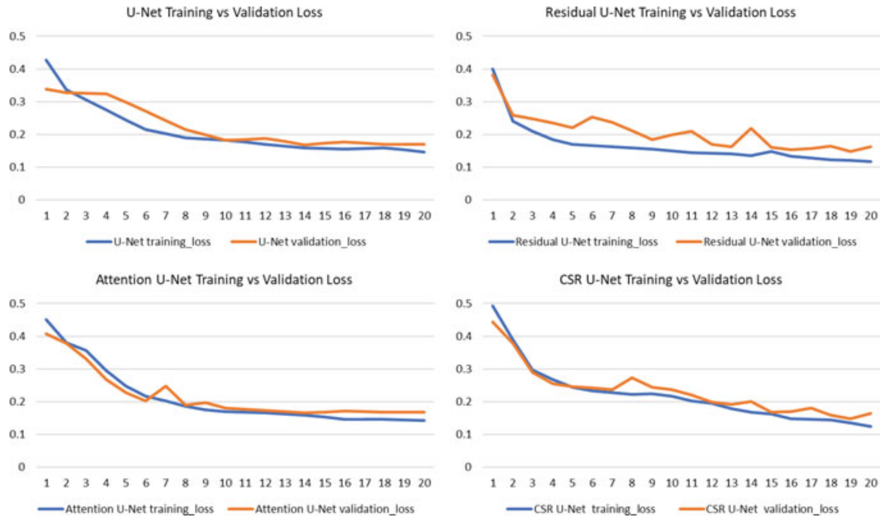


Fig. 6 The comparison of training and validation losses of the segmentation models U-Net, Residual U-Net, Attention U-Net, and CSR U-Net (proposed model architecture)

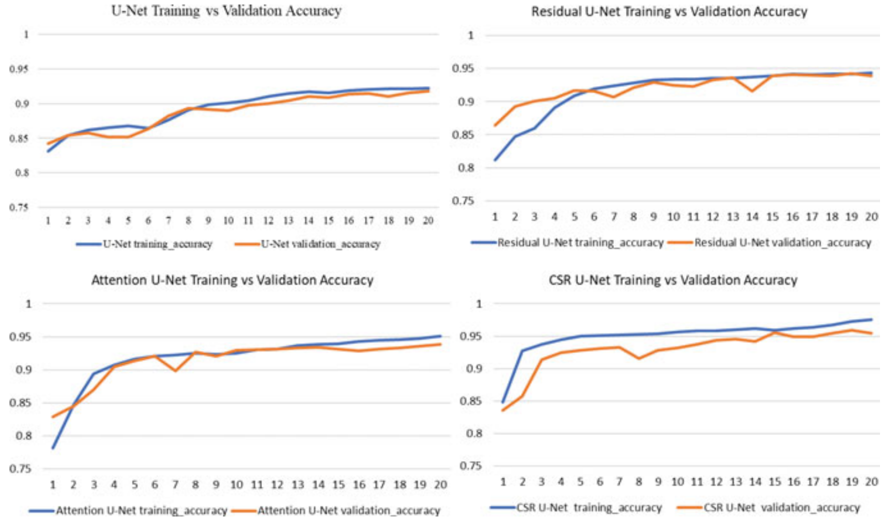


Fig. 7 The comparison of training and validation accuracies of the segmentation models U-Net, Residual U-Net, Attention U-Net, and CSR U-Net (proposed model architecture)

Table 1 Comparison of model metrics

Model	Accuracy	IoU	Dice score	Pixel accuracy
U-Net	0.93232	0.97958	0.95172	0.87756
Residual U-Net	0.94081	0.98966	0.96097	0.91820
Attention U-Net	0.94357	0.98959	0.96751	0.94390
CSR U-Net	0.97570	0.99964	0.97427	0.96040

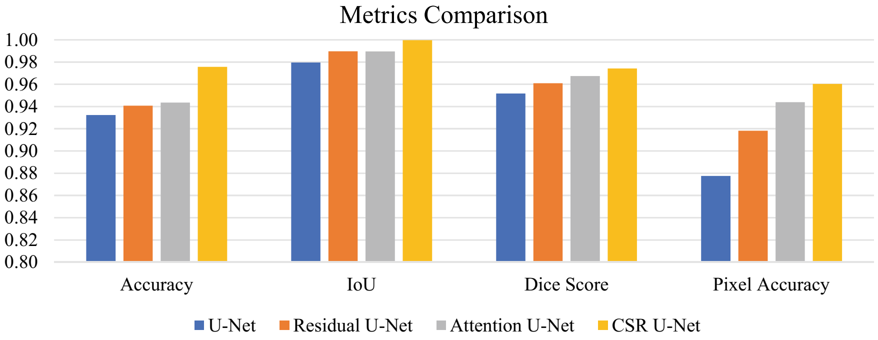


Fig. 8 Comparison of accuracy, IoU, Dice Score, and Pixel Accuracy of the segmentation models U-Net, Residual U-Net, Attention U-Net, and CSR U-Net (proposed model architecture)

5 Conclusion

This study introduced a novel deep learning model, the Channel–Spatial Regularized U-Net (CSR U-Net), for skin cancer lesion image segmentation. The proposed CSR U-Net model combines the strengths of the U-Net, Attention U-Net, and Residual U-Net models and introduces a unique combination of channel-wise attention and spatial-wise attention with regularization to prevent overfitting.

The results demonstrated the superior performance of the CSR U-Net model, with an accuracy of 0.9757, an IoU of 0.9996, a Dice Score of 0.9743, and a Pixel Accuracy of 0.9604. These results highlight its potential in addressing the current challenges in skin cancer lesion image segmentation. The study contributes to the ongoing efforts to improve the accuracy and efficiency of skin cancer lesion image segmentation, with the ultimate goal of improving early detection and treatment outcomes for patients with skin cancer. This work can set new standard in skin cancer lesion segmentation and CSR U-Net can offer as a foundation for any new research in this area. This could serve as a baseline for comparison.

Future work could explore further enhancements to the CSR U-Net model, such as the incorporation of additional attention mechanisms or the use of more sophisticated regularization techniques. Additionally, the model could be evaluated on other datasets and in other segmentation tasks to assess its generalizability.

References

1. SM J, P M, Aravindan C, Appavu R (2023) Classification of skin cancer from dermoscopic images using deep neural network architectures. Multimedia Tools Appl 82(10):15763–15778
2. Alenezi F, Armghan A, Polat K (2023) A novel multi-task learning network based on melanoma segmentation and classification with skin lesion images. Diagnostics 13(2):262
3. Dimililer K, Sekeroglu B (2023) Skin lesion classification using CNN-based transfer learning model. Gazi Univ J Sci 36(2):660–673
4. Tschandl P, Rosendahl C, Kittler H (2018) The HAM10000 dataset, a large collection of multisource dermatoscopic images of common pigmented skin lesions. Scientific Data 5:180161
5. Thaajwer MA, Ishanka UP (2020) Melanoma skin cancer detection using image processing and machine learning techniques. In: 2020 2nd international conference on advancements in computing (ICAC). IEEE, Malabe, pp 363–368
6. Thomas SM, Lefevre JG, Baxter G, Hamilton NA (2021) Interpretable deep learning systems for multi-class segmentation and classification of non-melanoma skin cancer. Med Image Anal 68:101915
7. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A (eds) MICCAI 2015, LNCS, vol 9351. Springer International Publishing, Cham, pp 234–241
8. Le PT, Tran DQ, Nguyen HQ (2022) Mobile antialiasing attention u-net for medical image segmentation. J Med Imaging Health Inform 12(2):1–9
9. Lee MC, Wang SY, Pan CT, Chien MY, Li WM, Xu JH, Luo CH, Shiue YL (2023) Development of deep learning with RDA U-Net network for bladder cancer segmentation. Cancers 15(4):1343
10. Ghodhbani G, Sahnoun M, Kallel F, Siarry P (2022) U-NET Architecture for automatic MS lesions segmentation using MR images. In: 2022 6th international conference on advanced technologies for signal and image processing (ATSIP). IEEE, Sfax, pp 1–5
11. Hong Z, Xi H, Hu W, Wang Q, Wang J, Luo L, Zhan X, Wang Y, Chen J, Chen L (2022) Multi-attentional u-net for medical image segmentation. In: 2022 2nd international symposium on artificial intelligence and its application on media (ISAIAM). IEEE, Xi'an, pp 135–139
12. Codella NC, Gutman D, Celebi ME, Helba B, Marchetti MA, Dusza SW, Kalloo A, aa K, Mishra N, Kittler H, Halpern A (2018) Skin lesion analysis toward melanoma detection. In: 15th international symposium on biomedical imaging (ISBI 2018). IEEE, pp 168–172

Automatic Detection and Classification System for Mesothelioma Cancer Using Deep Learning Models with HPO

Apeksha Koul, Rajesh K. Bawa, and Yogesh Kumar

Abstract Mesothelioma is a deadly cancer, but its early detection is important to save the life of a human. Hence, the paper focuses on the development of a novel method to detect Mesothelioma cancer using deep learning techniques like Gated Recurrent Unit, Multilayer Perceptron, and Long Short-Term Memory along with GridSearchCV(a hyper-parameter optimization technique). To evaluate the method, an experiment has been conducted on the dataset of 324 records, where 228 represent healthy individuals and 96 depict Mesothelioma patients. After analyzing and studying its pattern, feature selection technique such as Standard Scaler is applied to remove extraneous attributes. Besides this, SMOTE technique has been also used to address class imbalance and balance the binary classes in the data. During model training, all the applied models have been trained as well as examined for the parameters like precision, accuracy, loss, F1-score, recall, and AUC-ROC. In addition to this, for enhancing the performance of MLP model, GridSearchCV has been incorporated to fine-tune the hyper-parameters. During experimentation, the results show that the MLP model incorporated with GridSearchCV optimizer achieves the highest testing accuracy of 98.97%, precision and AUC-ROC of 1.00, while as F1-score and recall of 0.98. These findings indicate that our proposed approach obtained through GridSearchCV demonstrates improved performance and serves as a reliable tool for early Mesothelioma detection.

Keywords Mesothelioma · SMOTE · GridSearchCV HPO · Deep learning

A. Koul · R. K. Bawa
Punjabi University, Patiala, Punjab, India

Y. Kumar (✉)
Pandit Deendayal Energy University, Gandhinagar, Gujarat, India
e-mail: yogesh.arora10744@gmail.com; yogesh.kumar@sot.pdpu.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_12

1 Introduction

Machine learning and deep learning have reshaped the medical industry by playing an important role in numerous areas of health care. Due to these powerful algorithms, it has been possible to analyze massive amounts of medical data for improving diagnosis, treatment, and patient care. Besides this, these algorithms also work on genetic data, electronic health records, and other clinical data for the development of personalized medicine, which allows for more specific interventions and treatments [1].

In context of Mesothelioma, these models have also shown promising results to deal with it. Mesothelioma is a cancer that originates in the Mesothelium, a thin layer of tissue that covers the body's internal organs. This cancer most often affects the lung lining, but it can also damage the lining of the belly, heart, and other organs. Mesothelioma is typically caused by asbestos exposure, which is a naturally occurring mineral that is frequently employed in building and industrial applications. After being inhaled, asbestos fibers trap in the lungs as well as other organs which thereby causes scar and produces inflammation [2]. If not treated on time, it can also lead to cancer as shown in Fig. 1.

The symptoms of Mesothelioma patients are pain in the chest, exhausting, coughing, breath shortness, and reduced weight, but unfortunately the clinicians cannot rely on these symptoms as they do not turned up until or unless the cancer has reached to its final stage which ultimately makes the treatment difficult. Oncologist treats the Mesothelioma patients with chemotherapy, surgery, and radiation therapy. Additionally, the prognosis rate of Mesothelioma is terrible as patients only survive for few months to years after being diagnosed [3].

On the other side, recent developments in machine learning as well as deep learning models have shown tremendous promise in terms of enhancing the recognition and classification of Mesothelioma, as mentioned earlier. Early Mesothelioma signs can be identified using machine learning and deep learning algorithms, enabling earlier diagnosis and treatment. This improves patient outcomes and increases

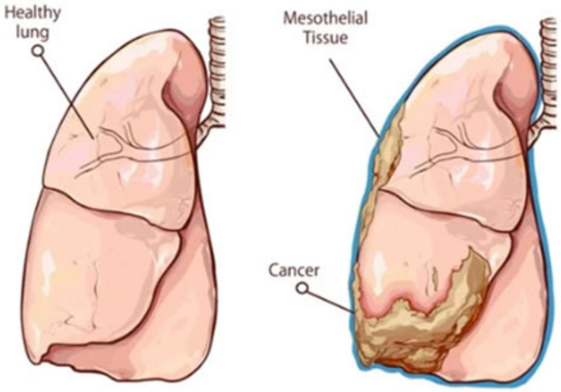


Fig. 1 Healthy lung versus mesothelioma lung [3]

survival rates [4]. In fact, it can be challenging to distinguish Mesothelioma from other conditions that affect the lungs or chest cavity, but it can be solved if AI-based learning algorithms will be trained on large datasets. This reduces misdiagnosis and enhances patient outcomes. Overall, machine learning and deep learning have the potential to improve Mesothelioma detection and classification, resulting in earlier as well as more accurate diagnosis along with the personalized treatment options [5].

Researchers have contributed a lot by applying multiple machine as well as deep learning models to predict and classify Mesothelioma disease. Courtiol et al. [6] had used a novel CNN method, known as MesoNet for predicting the survival rate of patients who suffered from Mesothelioma by analyzing whole slide digitized images. Similarly, Gupta et al. [7] worked on the 324 Malignant Mesothelioma data which had been collected from UCI machine learning repository. The researchers initially worked on the class imbalance issue of the dataset by using the techniques such as ADASYN and SMOTE. For feature selection, Principal component Analysis technique had been applied along with the artificial neural network which computed the highest accuracy of 96%. Alam et al. [8] investigated the risk factors for Malignant Mesothelioma (MM). To identify the symptoms of Mesothelioma, they analyzed data from both healthy individuals and those with the disease. However, the dataset had an imbalance in the number of healthy individuals compared to those with MM. To address this issue, they used a technique called synthetic minority oversampling. They further applied the Apriori algorithm, which involved removing duplicate and irrelevant attributes and converting numerical attributes into nominal ones. Zadsafar et al. [9] presented a machine learning approach for the early detection of Mesothelioma disease. Their study used an accessible database and applied various methods to the challenge of diagnosing Mesothelioma. Performance metrics such as accuracy and sensitivity were employed to assess the efficacy of the methods. The findings demonstrated the diagnostic effectiveness of different machine learning techniques and established a successful early detection system. Mukherjee et al. [10] used data mining techniques for classifying the data of the patient along with Multilayer Perceptron ensembles and support vector machine. While implementing the model for ten cross-validations, it was found that support vector machine obtained the highest accuracy rate of 99.87%, while as Multilayer Perceptron ensembles (MLPE) obtained 99.56%.

After analyzing the research thoroughly, it has been analyzed that the machine and deep learning models have done tremendous performance to detect and classify Mesothelioma disease.

Hence, in this paper, the aim is to develop an AI-based model to perform the binary classification of Mesothelioma disease which predicts whether the patient is normal or is suffering from Mesothelioma.

1.1 Contribution

The contribution that has been made to develop the AI-based model for the detection and classification of Mesothelioma is mentioned as under:

- Initially, the data is collected in the form of an xlsx dataset with 34 attributes.
- The collected dataset is then converted to .csv and cleaned to check for NaN or missing values before being visualized in the form of a bar graph and correlation matrix.
- Later, feature selection technique is used to choose the necessary features as well as delete the undesirable ones. Moreover, the data has been suffered from a class imbalance issue, i.e., minority and majority classes of Mesothelioma and healthy people, respectively. Hence to work on it, SMOTE technique is used to balance the class for the improvisation of the classification accuracy.
- In the next stage, a feedforward network, also known as Multilayer Perceptron (MLP) model, Long Short-Term Memory, and Gated Recurrent Model along with the hyper-parameter optimizer technique (HPO), i.e., GridSearchCV is proposed and trained on the specified featured dataset and is evaluated using a variety of evaluative criteria, which include accuracy, precision, loss, recall, AUC-ROC, and F1-score.

The structure of the paper goes like this: Sect. 1, which has already been covered as an introduction defines the brief background of Mesothelioma disease and the impact of AI learning models to detect and classify it along with little literature work in the same domain. Section 2 covers the main part of the research, i.e., the flow to conduct the work which includes the dataset, cleaning of the data, feature selection, data augmentation, training the models, and the metrics to evaluate them. The result part is covered in Sect. 3, and the whole paper is summarized in Sect. 4.

2 Methodology

This section describes the methodology (Fig. 2) that has been used to develop a proposed system for the detection and classification of Mesothelioma disease. Initially libraries have been imported as they are important in the field of model development because they provide developers with pre-built functions, tools, and classes that make their implementation more effective and efficient. In this study, we have used various libraries such as *Pandas, Numpy, Scikit-learn, Imbalanced-learn, Tensorflow, and Matplotlib* to preprocess and develop the model.

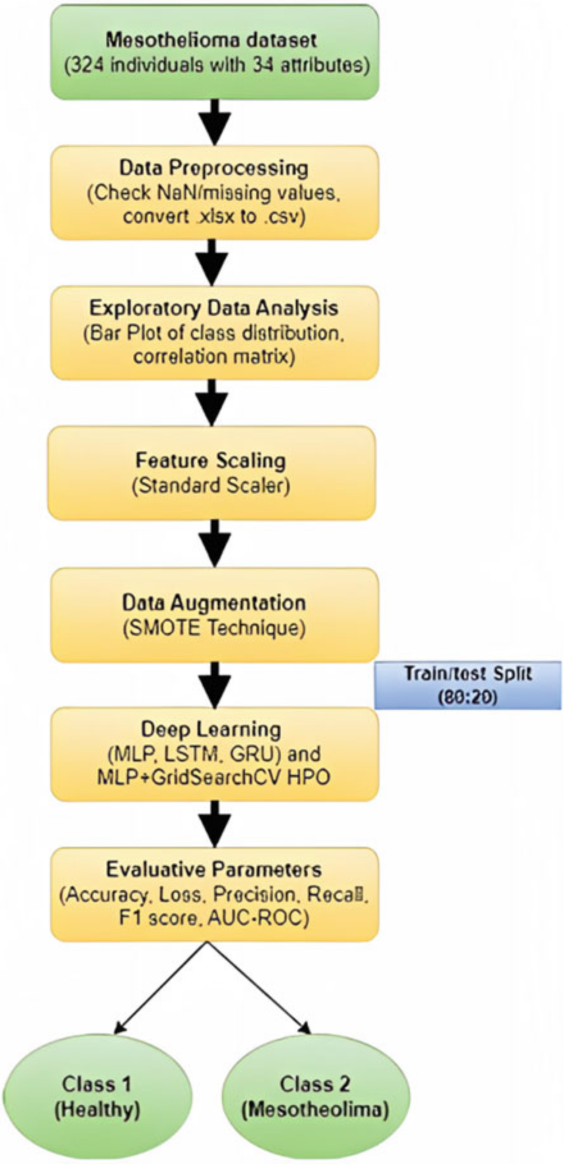

Mesothelioma dataset
(324 individuals with 34 attributes)
Data Preprocessing
(Check NaN/missing values, convert .xlsx to .csv)
Exploratory Data Analysis
(Bar Plot of class distribution, correlation matrix)
Feature Scaling
(Standard Scaler)
Data Augmentation
(SMOTE Technique)
Train/test Split
(80:20)
Deep Learning
(MLP, LSTM, GRU) and MLP+GridSearchCV HPO
Evaluative Parameters
(Accuracy, Loss, Precision, Recall, F1 score, AUC-ROC)
Class 1
(Healthy)
Class 2
(Mesotheolima)

Fig. 2 Proposed model to detect and classify mesothelioma disease

2.1 Dataset

For Mesothelioma disease, the dataset has been taken from the UCI machine learning repository and it was prepared at Dicle Univeristy Faculty of Medicine in Turkey. The dataset is available in the.xlsx format and is having the data records of three hundred and twenty-four individuals out of which 228 are non- Mesothelioma people and

96 are Mesothelioma patients which is classified as ‘**1**’ and ‘**2**’, respectively, with thirty-four attributes each. The attributes are named as gender, asbestos exposure, city, type of MM, diagnosis method, exposure, keep side, duration of symptoms, etc. [7].

2.2 Data Preprocessing

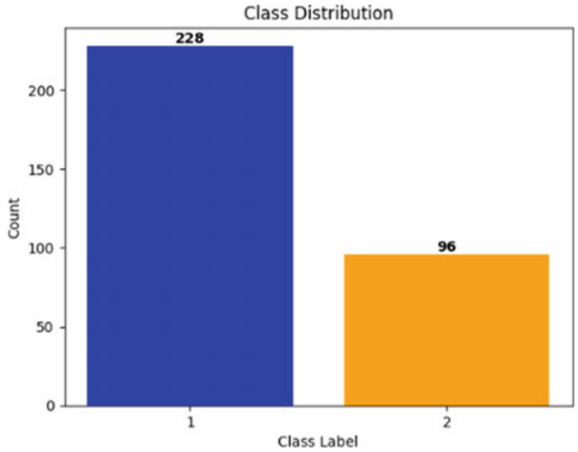
It is impossible to overestimate the significance of preprocessing the data in the development of any disease detection and classification model. In this study, the data has been initially translated from **.xlsx to .csv** format to simplify its structure. CSV files are less complicated and easier to deal with than .xlsx files, which may contain several sheets, formatting, and other advanced features. This is especially true when working with programming languages or tools that do not yet support .xlsx files. The dataset is then thoroughly checked for data quality to identify any missing or null values using the functions **isnull.any() and isnull.sum()**. During this process, it has been found that data does not contain any null or missing values which mean that the Mesothelioma dataset is clean.

2.3 Exploratory Data Analysis

The role of Exploratory Data Analysis (EDA) is to understand the pattern as well as the characteristic or property of any data. In this section, bar plot graph, as shown in Fig. 3, has been used to visualize the dataset records in the form of a class distribution where Class 1 defines the healthy people who are 228 in number, and on the contrary, Class 2 labels the Mesothelioma patients whose count is 96. In addition to this, this chart is also used to depict the presence of class imbalance or class balance in the dataset, and from the figure, it can be clearly seen that Class 2 is underrepresented as compared to Class 1.

A correlation matrix is typically used to measure the linear relationship between pairs of variables in a dataset. It provides a concise representation of the relationships in the dataset which allows exploring and visualizing patterns. By examining the correlation coefficients with the target variable, we can identify variables that have a strong linear relationship with the outcome. These variables can be considered as potentially important features for the classification model. Keeping this is in view, the correlation matrix of the Mesothelioma dataset has been also generated and is shown in Fig. 4.

Fig. 3 Bar plot class distribution of Mesothelioma dataset



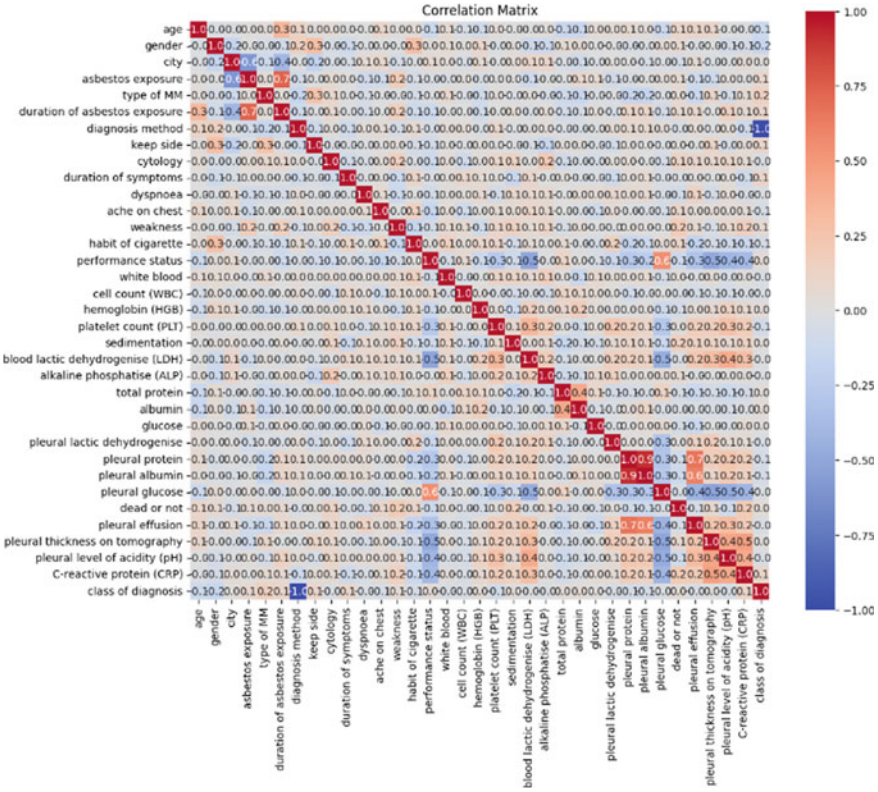


Fig. 4 Correlation matrix of Mesothelioma dataset

2.4 Feature Selection

In this phase, the preprocessed dataset has been taken for the selection of features to make sure that they fall on a same scale and can prevent one feature from dominating the others during model training. In this research, we have used standard scaling technique which is also known as Z-score normalization or standardization. It is a technique for scaling features and ensures that all the features have comparable scales and ranges [11]. Equation (1) is used to compute standard scaler technique mathematically.

$$X_{\text{scaled}} = \frac{(X - \mu(X))}{\sigma(X)}. \tag{1}$$

Here, X is the original set of feature values, μ (X) is mean, and σ (X) is the standard deviation of the feature values. Furthermore, while investigating the dataset, it came out that there are two attributes with identical values, so one attribute, i.e., **diagnosis method**, was dropped by using ***data.drop()*** in order to reduce the dimensions of the data.

2.5 Data Augmentation

In the Mesothelioma dataset, a class imbalance issue has been found as there are 228 negative instances as the majority class (non-Mesothelioma) and 96 positive samples as the minority class (Mesothelioma patients). To overcome it, SMOTE technique is implemented to address the class imbalance in datasets, especially when the minority class is substantially underrepresented relative to the majority class [7]. **Sampling strategy of 1** is used and k_neighbours = 5 is chosen as the number of nearest neighbors to guide interpolation, establishing a balance between preserving the local structure of the minority class and avoiding over-fitting or under-fitting. The **random state is set to 72**, which functions as a seed value to initialize the random number generator used by SMOTE, ensuring that synthetic examples generated by multiple runs of the same code are consistent and promoting reproducibility of results. Later, the data is divided into two subsets, training and testing, in **80:20 ratio**.

2.6 Applied Classifiers

In this study, four deep learning models as well as the hyper-parameter optimizer have been briefly described along with their hyper-parameters which are shown in Table 1.

Table 1 Hyper-parameters of deep learning models

Hyper-parameters	Values
Activation	Sigmoid
Optimizer	Adam
Learning rate	0.0001
Batch_size	32
Loss	Mean squared error/binary cross-entropy
Hidden layer size	64
Dense layer	1 unit
Dropout rate	0.2
GridSearchCV hyper-parameter optimizer	
Hidden layers	1, 2, 3
Hidden layer size	16, 32, 64
Learning rate	0.0001, 0.001, 0.01
Dropout rate	0.2, 0.3, 0.4

Multilayer Perceptron: A Multilayer Perceptron (MLP) is a type of artificial neural network (ANN) that has an input layer, one or more hidden layers, and an output layer in which each layer is made up of artificial neurons, also known as perceptrons [6]. These layers perform an affine transformation of a linear sum of input and are mathematically defined as shown in Eq. (2).

$$y = f(WxT + b), \tag{2}$$

where activation function is f, W defines the set of parameters in the layer, x is the input vector, and b is the bias vector. In fact, MLP can handle CSV datasets where each row represents a data point, and each column corresponds to a particular feature.

LSTM: Long Short-Term Memory is a kind of recurrent neural network (RNN) architecture that is designed for handling sequential data as well as capture long-term dependencies. The LSTM architecture consists of multiple memory cells and these memory cells are interconnected as well as capable of retaining information over long periods. There are three main components in each LSTM cell such as an input gate (I_T), a forget gate (F_T), and an output gate (O_T). These gates, within the LSTM as well as along with a cell state, regulate the flow of information and allow it to either forget information or selectively remember at each time step. LSTM can be useful for Mesothelioma disease detection by leveraging its capabilities in analyzing sequential data such as electronic health records, medical reports, and patient histories [12]. Mathematically, it can be defined by Eqs. (3–5).

$$I_T = \sigma\left(W_I\left[h_{T-1}, X_t\right] + b_I\right), \tag{3}$$

$$F_T = \sigma\left(W_F\left[h_{T-1}, X_t\right] + b_F\right), \tag{4}$$

$$O_T = \sigma\big(W_O\big[h_{T-1}, X_t\big] + b_O\big). \tag{5}$$

Here W_X—the weight of all gates, σ—sigmoid function, h_{T-1}—output of the previous LSTM block at timestamp $(T-1)$, X—neurons, b_X—biases for respective gates, and X_T—input at current timestamp.

GRU: Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture, but it is a simplified version of LSTM and has fewer gating mechanisms, making it computationally efficient. The architecture of a GRU also consists of memory cells, but unlike LSTM, it has two main gates: a reset gate (R) and an update gate (Z). These gates control the flow of information within the GRU and determine how much past information should be carried forward and how much new information should be incorporated [12]. Mathematically, it can be computed by using Eqs. (6, 7, 8, and 9).

$$Z_t = \sigma\big(W_z.\big[H_{t-1}, X_t\big]\big), \tag{6}$$

$$R_t = \sigma\big(W_z.\big[H_{t-1}, X_t\big]\big), \tag{7}$$

$$\widetilde{H_t} = \tanh(W.[R_t * H_{t-1}, x_t]), \tag{8}$$

$$H_t = (1 - Z_t) * H_{t-1} + Z_t * \tilde{h}_t, \tag{9}$$

where H and $\tilde{H}$ represent the output and intermediate memory, respectively, W represent parameter matrices, X is an input, and σ is an activation function.

GridSearchCV: It is a technique used to find the optimal hyper-parameters for a machine learning model. It is a brute-force approach that exhaustively searches through a predefined set of hyper-parameter combinations to identify the combination that yields the best performance [13]. In this research, to fine-tune the parameters of MLP model, the hyper-parameter tuning is performed using ***grid search***, where different combinations of hyper-parameters are tested to find the optimal configuration. The hyper-parameters being tuned include the number of *hidden layers*, *the size of each hidden layer, the dropout rate,* and *the learning rate.* The model is trained for a maximum of ***25 epochs*** with a ***batch size of 32. Early stopping*** is implemented to prevent over-fitting. The '***EarlyStopping' callback = 5*** is used, which monitors the validation loss and stops training if it does not improve for a specified number of epochs. The 'GridSearchCV' class from scikit-learn is employed for the grid search process. It performs cross-validation with three folds ***('cv = 3')*** and evaluates each parameter combination based on the training and validation data. The ***'n_jobs' parameter is set to − 1***, allowing the grid search to use all available CPU cores for parallel processing. The results which include training and testing scores are stored in '***grid_result***'.

Table 2 Evaluative parameters

Metrics	Description	Formulae
Accuracy	It reflects the ratio of successfully classified instances to all of the dataset's instances	$\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$
Loss	The difference between the model's actual output and its expected output is measured as loss	$\frac{(\mathrm{AV-PV})^2}{\text{Number of observations}}$
Precision	Precision measures how well the model can classify positive classes	$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$
Recall (TPR)	Recall measures the ability of the model for the identification of all positive instances	$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$
F1-score	F1-score provides a balancing measure for the accuracy of a model by combining recall and precision into a single value	$2\frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Recall} + \mathrm{Precision}}$
AUC-ROC	AUC is used to assess the effectiveness of binary classification models by distinguishing the positive from the negative class	$\int \frac{\mathrm{FPR}(T)}{\mathrm{TPR}(T)}$
$\mathrm{FPR}_{\text{Positive and Negative}}$	It is a performance metric used in binary classification tasks to evaluate the rate at which the model incorrectly classifies negative instances as positive	$\frac{\mathrm{FP}}{\mathrm{TN+FP}}$

Here, *TP* true positive, *FP* false positive, *AV* actual value, *FN* false negative, *PV* predicted value, *FPR* false positive rate, *TN* true negative, *TPR* true positive rate.

2.7 Performance Metrics

Various evaluative parameters such as accuracy, loss, precision, recall, F1-score, as well as AUC-ROC have been used to examine the performance of the deep learning models. The brief description about these parameters along with their formulae is provided in Table 2 [14].

3 Results and Analysis

After developing the models and applying dataset to them, their performance has been evaluated on the basis of parameters such as accuracy, loss, precision, recall, F1-score, and AUC-ROC as discussed in Sect. 2.7. These metrics provide a comprehensive assessment of the model's predictive capability and overall effectiveness in capturing the target variable.

Initially the models have been evaluated during training and testing phases for accuracy and loss whose results are shown in Table 3.

During training period, it has been found that LSTM and GRU computed the best accuracy by 98.87 and 98.07%, respectively. Similarly, during testing phase, these two models also worked well by obtaining 96.92 and 98.46% accuracies, respectively.

Table 3 Performance analysis of deep learning models

Model	Training		Testing	
	Accuracy (%)	Loss	Accuracy (%)	Loss
Multilayer perceptron	66.67	4.13	70.76	3.47
LSTM	98.87	0.06	96.92	0.08
GRU	98.07	0.07	98.46	0.04
MLP with GridSearchCV	98.68	0.04	98.97	0.01

But, on the contrast, Multilayer Perceptron showed that the least performance is case of both training and testing phases by computing only 66.67 and 70.76% accuracies which also includes 4.13 and 3.47 as their worst loss values by using its default parameters. Hence to explore it more, the parameters of MLP model have been fine-tuned using GridSearchCV optimizer, and after evaluating them, good results have been seen in both training and testing phases by computing 98.68 and 98.97% accuracies, respectively.

Likewise, the other parameters to examine the performance of the applied deep learning models along with the MLP+ GridSearchCV have been also computed in Table 4.

The superior performance of LSTM, GRU, and the MLP model with GridSearchCV as compared to the basic MLP model can be attributed to the specialized mechanisms in LSTM and GRU architectures that enable effective capture of long-term dependencies in sequential data and mitigating the vanishing gradient problem. Additionally, for MLP+ GridSearchCV, the hyper-parameter optimizer optimizes the performance of sequential MLP model by exploring different combinations and fine-tuning its parameters systematically using GridSearchCV. In a nutshell, this fine-tuning process helps to find the optimal configuration for the MLP model which leads to improve its precision, recall, F1-score, and AUC values.

The models were also evaluated as presented in Table 5, using the same metrics as shown in Table 4. However, this evaluation is conducted for two distinct classes as mentioned earlier: the negative class labeled as ‘1’, which indicates the absence of Mesothelioma, and the positive class labeled as ‘2’, which signifies the presence of the Mesothelioma disease.

In terms of precision, the MLP model has moderate precision and recall for both the negative and positive classes. The LSTM and GRU model shows higher precision

Table 4 Examining models using other metrics

Model	Precision	Recall	F1-score	AUC-ROC
MLP	0.50	0.70	0.58	0.60
LSTM	0.98	0.98	0.97	0.82
GRU	0.95	0.92	0.93	0.87
MLP with GridSearchCV	1.00	0.98	0.98	1.00

Table 5 Evaluating models for binary class

Model	Class	Precision	Recall	F1-score	AUC-ROC
MLP	Negative	0.75	0.50	0.62	0.80
	Positive	0.50	0.70	0.58	0.60
LSTM	Negative	0.98	0.98	0.97	0.87
	Positive	0.83	0.98	0.91	0.85
GRU	Negative	0.99	1.00	0.98	0.98
	Positive	0.95	0.92	0.93	0.87
MLP with GridSearchCV	Negative	1.0	0.98	0.99	1.0
	Positive	0.96	1.0	0.98	1.0

as well as recall for both classes, particularly for the negative class. The MLP model with GridSearchCV achieves perfect precision for the negative class and positive class while as perfect recall for the positive class. In case of F1-score, the LSTM model shows the highest F1-score for both classes, followed by the GRU model. The MLP models have relatively lower F1-scores but achieve a reasonably good AUC-ROC score, while the LSTM and GRU models exhibit higher AUC-ROC scores. The MLP model with GridSearchCV achieves perfect AUC-ROC scores.

Upon conducting an analysis, it has been observed that no doubt other models worked well for the dataset, but the initial training of the Multilayer Perceptron (MLP) model without fine-tuning its parameters yielded the lowest results. However, a significant improvement has been observed when the same model was re-evaluated after fine-tuning its parameters using a hyper-parameter optimizer, i.e., GridSearchCV. By employing GridSearchCV, which systematically explores different combinations of hyper-parameters, the MLP model was able to achieve superior results in terms of accuracy, precision, recall, F1-score, and AUC. This optimization process allowed the model to fine-tune its internal parameters, adjusting them to the optimal values for the given task.

4 Conclusion

The use of deep learning models for Mesothelioma detection presents a promising approach with significant potential in improving patient outcomes and survival rates. In this study, various models such as MLP, LSTM, and GRU had been used to develop the system for the detection and classification of Mesothelioma disease. These models showed an impressive performance metrics, in terms of high accuracy, precision, F1-score, recall, and AUC-ROC values except for MLP. Hence to explore the MLP model in depth, its parameters were optimized using GridSearchCV hyper-parameter optimizer which boosted its performance in terms of all performance metrics. However, Mesothelioma detection using applied models does come with certain challenges. One of the primary challenges is the availability of high-quality and well-annotated datasets. Obtaining a sufficiently large and diverse dataset with reliable ground truth labels for training and testing purposes remains a challenge in the field of Mesothelioma research. Moreover, the issue of class imbalance within the dataset poses a significant challenge. Although this issue had been overcome by using SMOTE technique, but it can also be addressed using other oversampling/ undersampling methods so that the models can effectively learn from both positive and negative instances, to ultimately improve its generalization ability.

In terms of future scope, several areas can be explored to further enhance Mesothelioma detection using deep learning models. The inclusion of other advanced AI learning algorithms, such as ensemble methods or deep learning architectures and fine-tuning their parameters using hyper-parameter optimizers, may further improve the accuracy and robustness of Mesothelioma detection.

References

1. Kumar Y (2020) Recent advancement of machine learning and deep learning in the field of healthcare system. Comput Intell Mach Learn Healthc Inform 1:77–98
2. Saxena K et al (2022) Appropriate supervised machine learning techniques for mesothelioma detection and cure. Biomed Res Int 2022:1–11
3. Mestrovic T (2023) What is mesothelioma? News-Medical.net. https://www.news-medical.net/health/What-is-Mesothelioma.aspx. Last accessed 14 Aug 2023
4. Gupta S, Gupta MK, Shabaz M, Sharma A (2022) Deep learning techniques for cancer classification using microarray gene expression data. Front Physiol 13:1–14
5. Hu X, Yu Z (2018) Diagnosis of mesothelioma with deep learning. Oncol Lett 17:1483–1490
6. Courtiol P et al (2019) Deep learning-based classification of Mesothelioma improves prediction of patient outcome. Nat Med 25:1519–1525
7. Gupta S, Gupta MK (2023) Computational model for prediction of malignant Mesothelioma diagnosis. Comput J 66(1):86–100
8. Alam TM et al (2021) A machine learning approach for identification of malignant mesothelioma etiological factors in an imbalanced dataset. Comput J 65:1740–1751
9. Zadsafar F et al (2022) A model for Mesothelioma cancer diagnosis based on feature selection using Harris hawk optimization algorithm. Comput Methods Program Biomed Update 2:1–7
10. Mukherjee S (2018) Malignant mesothelioma disease diagnosis using data mining techniques. Appl Artif Intell 32:293–308
11. Kumar Y et al (2021) Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things. Wirel Commun Mob Comput 2021:1–16
12. Sinha A, Kayaalp F (2021) Classification performance evaluation on diagnosis of breast cancer. Springer eBooks 76:237–245
13. Di Genova A et al (2022) A molecular phenotypic map of malignant pleural Mesothelioma. GigaScience 12:1–13
14. Koul A, Bawa RK, Kumar Y (2022) Artificial intelligence techniques to predict the airway disorders illness: a systematic review. Arch Comput Method Eng 30:831–864

A Systematic Literature Survey on IoT in Health Care: Security and Privacy Threats

Aryan Bakliwal, Deepak Panwar, and G. L. Saini

Abstract With the advancements in IoT technology, it is becoming a part of our life more than yesterday. IoT technology is revolutionizing the Healthcare sector by helping medical professionals in providing better care and remote monitoring to patients. Medical IoT devices generate very sensitive data from the patients, and security of this data is crucial for privacy of patients. Security solutions for Healthcare IoT systems need to be tailored specifically for these resource-limited IoT devices. This study surveys the various solutions given in the past few years for Healthcare IoT security. In this study, articles having solutions based on latest technologies such as blockchain, RFID are reviewed and analyzed to get better insights about the threats and their countermeasures' results and features. In addition to this, the directions for future study are also mentioned. The aim of this study is to help everyone interested in the domain by providing a summary of latest schemes and generate interest among new scholars toward security concerns in Healthcare IoT.

Keywords IoT · Health care · Security · Privacy · Cyber-threats

1 Introduction

Internet of Things (IoT) has seen growth exponentially in every sector. From small sensors to complex and large Healthcare ecosystems, IoT has made possible having real-time patient monitoring and much more without the need of much human involvement which was not possible before. Personal and Clinical can be the broad categorization of Healthcare IoT where personal includes devices for general use by the user without medical professional and clinical includes devices which are intended strictly for medical use involving medical professional [1]. Healthcare IoT devices, whether personal or clinical generate enormous amount of data from the patient, which is very sensitive and if not managed properly, can be a threat to the

A. Bakliwal · D. Panwar · G. L. Saini (✉)
Manipal University Jaipur, Jaipur, Rajasthan, India
e-mail: glsaini86@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_13

security and privacy of a person to whom the data belongs to and can be used for wrong purposes by the hackers. So, the IoT systems must be made secure to threats and attacks without compromising the functioning of the devices. More than 80% of participants of a survey are ready to adapt or have already done to the changes in IoT-based Smart Health care [2]. Apart from helping the patients, the data from the patients also helps the medical professionals in studying about their condition and gaining knowledge. These reasons make the patients' sensitive data very important and must be securely transferred over the cloud, and this arises the need of a robust and efficient solution for data privacy and security which fulfills all requirements and is lightweight. Recently, a security breach incident involving 3 million patients happened at Advocate Aurora Health. The breach happened due to Pixels—an online tool by which sensitive information was transferred to unauthorized companies.

Healthcare IoT consists of three main layers—Perception layer (physical IoT device, sensors, etc.), Network and Data Processing layer (network devices), and Application layer (services and applications). There is no standard protocol for communication in IoT, and use of less secure protocols such as SigFox, LoRa, ZigBee can cause privacy problems [3]. The security measures must consider the threats of every layer for better security and privacy.

The rest of the study is structured as follows: Sect. 2 presents IoT Healthcare security challenges, threats, and requirements. Section 3 contains overview of latest technologies and literature review of security systems based on them with their methodology and results, Sect. 4 concludes the survey, and Sect. 5 provides scope for future research.

2 Threats, Challenges, and Requirements

2.1 Threats

Of all the threats, Distributed Denial of Service (DDoS) is the most frequent and most destructive [4]. In Session med jacking attack, the attacker tries to get access to the system using a valid session of an authorized user [4]. Schemes that involve session key establishment are vulnerable to this threat. Some other attacks based on layer-wise classification are shown in Fig. 1.

2.2 Challenges

For implementation of a secure and robust mechanism, several challenges need to be considered. Standardization is a big challenge as there are several different methods and protocols for use, so there is a need for a standard architecture for security [3]. Another major challenge is the resource-limited nature of IoT devices due to which

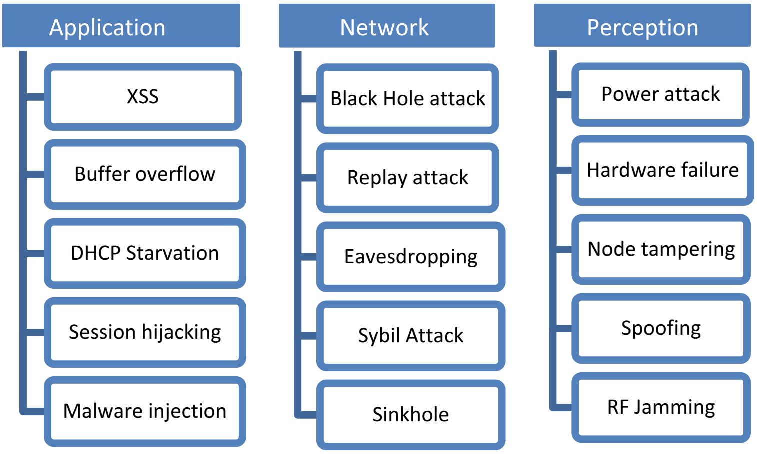


Fig. 1 Layer-wise IoT security threats

Table 1 IoT security challenges [3]

Data	Confidentiality
	Authenticity
Communication	Authentication
	Access control
End application	Privacy concern
	Forensic challenges
	Social and legal challenges

complex and heavy schemes that require more resources cannot be implemented on these devices. Some other security challenges are mentioned in Table 1.

2.3 Requirements

IoT Healthcare systems should have immunity against major cyber-attacks. They should be scalable as the Healthcare ecosystems and IoT devices keep on increasing day by day, so they should be able to connect to other networks [5]. They should also be lightweight, energy efficient, and low-cost.

3 Techniques Review

There are several various operations all taking place simultaneously in an IoT device and a good security solution is which considers prevention of every possible threat. But constructing such a system is an ideal case scenario, and in the real-world, there are many challenges to this. So, the researchers try to get as close as possible with their solutions and some of the recent schemes proposed by them are reviewed as follows.

3.1 Blockchain

Blockchain is a peer-to-peer network of connected devices. A hash is generated at the time of block creation and the details of the data stored in this block are dependent on hash of the same and previous block as shown in Fig. 2. Once stored, the data inside a block is very difficult to change thus making it immutable. Blockchain is a decentralized system which makes it immune to single point failure. Using blockchain can enhance the security of IoT systems [6].

Blockchain uses different consensus mechanisms, namely Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA). Comparison of these mechanisms is shown in Table 2 [7].

In [7], the authors address the importance of medical data privacy and propose development of a triple encryption system for EMR which uses blockchain. The proposed model Secure Electronic Medical Record Exchange Structure (SEMRES)

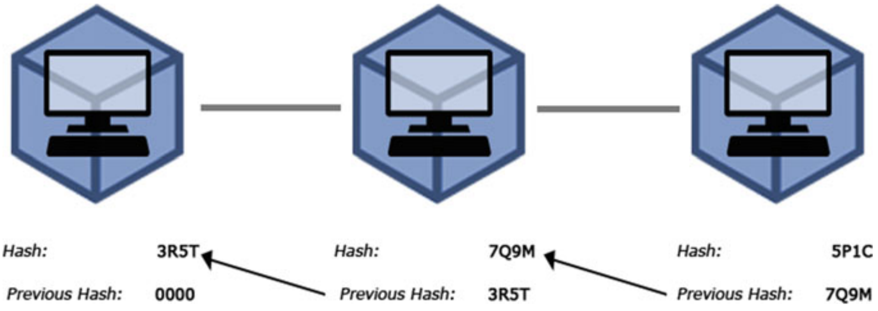


Fig. 2 Structure of blockchain network

Table 2 Comparison of consensus mechanisms

Mechanism	Speed	Resource need	Privacy
PoW	Slow	High	Low
PoS	Moderate	Moderate	Moderate
PoA	Fast	Low	High

is implemented in three modules. The first module is hybrid dual encryption method based on AES and RSA, second is Decentralized EMR Repository (DERy), and third is a blockchain architecture for data validation. The results show that the proposed system provides a secure and efficient way to exchange encrypted medical data between different agencies.

In [8], the authors try to provide medical practitioners with real-time accurate and trusted data by proposing a blockchain and ML-based framework. The framework has three modules—IoT module, which fetches the data. Blockchain module uses two different blockchain networks for managing the data, one maintained by the patient and other for the doctor and the ML module detects any anomaly in the data being generated. The framework ensures authenticity of data, helps multiple stakeholders interact with the data and provide remote monitoring of patients.

In [9], the authors propose and evaluate Fuzzy and Blockchain-based Adaptive Security for Healthcare IoTs (FBASHI), an adaptive security system based on fuzzy logic and blockchain technology and developed using Hyperledger platform. The analysis of the framework is done by designing and testing it in MATLAB and then applying the tested logic to Hyperledger for validity. The results show that the precision of output is directly proportional to number of transactions. The system is scalable and transaction throughput is 10,000 transactions per second. In the future, the authors plan to explore blockchain's AI capability.

In [10], the authors are concerned about the security issues that arise due to the distributed nature of edge computing and propose a secure user-centric edge computing system using blockchain. The goal is to capture user data transaction history on the blockchain. User anonymity is provided by the edge node's retention of user data privacy, and data can only be sent between registered IoT devices. The proposed system is evaluated and compared with three other blockchain-based systems and incorporates important security features such are data filtering, edge computation, accountability, user-centric, privacy and anonymity of user, implementation, and use of cryptographic functions.

In [11], the authors express a need for a secure authentication system and propose a secure and robust authentication tool using Lamport Merkle Digital Signature (LMDS) helped with blockchain. The LMDS technique is used for verification, and only upon successful verification, the data is stored in the server. The results show that using the LMDS technique, the computational overhead and communication time decreased and security increased as compared to other existing models.

In [12], the authors aim to develop BlocMedCare, a blockchain-based security system especially for remote monitoring. The model has three parts—patient, medical team, and InterPlanetary File System—which is used for storing encrypted data. Smart contracts are used for access control. Proof of Authority (PoA) based on Ethereum is used for fast transfer. The results show that using PoA increases the throughput thus giving a fast-processing time of 45 transactions per second. The system is secure and scalable. In the future, the authors plan to implement the proposed system using Hyperledger platform and integrate AI to it.

In [13], the problem addressed is the frequent attacks on Healthcare IoT systems and security of Electronic Medical Report (EMR) of patients and "Medichain"—a blockchain-based framework is proposed. Using recursive method, a Merkle tree is formed for storing the medical record after hashing them to improve data security and privacy. The results show that the average execution time and memory usage vary linearly and increase with increase in the number of blocks in the blockchain.

The authors of [14] also address the problem of security of Electronic Medical Report (EMR) and Electronic Health Records (EHRs) and need of their emergency access. They propose a conceptual framework for security of medical data and having emergency access capabilities using blockchain. InterPlanetary File System is used to store encrypted data in decentralized manner and Attribute-Based Access Control is used for storing rights and roles for medical data and provide break-glass capabilities.

In [15], the authors are concerned about the use of centralized security solutions having single point failure problem and propose the development of a blockchain system. The methodology is employed by arranging clusters of IoT devices in a hierarchical method and the load of processing is shared. A consensus scheme based on Identity-based encryption derived from Proof of Authentication is used for authentication. The maximum message size and maximum CPU load are less in multi-level blockchain as compared to single-level blockchain. The time required to create a block and authentication is also low and the framework is adaptable, scalable, and secure to numerous attacks as shown by the results.

The authors in [16] propose a task offloading strategy with emergency handling capacity. The proposed system is implemented using Software-Defined Networking (SDN) and blockchain. The system has a two-layer multidimensional security approach. To make the system time efficient, a blockchain sharding scheme is also designed. The results show that this mechanism is efficient and reliable in real-time usage. The single point failure problem of SDN is also solved using blockchain.

Discussion

Being decentralized, blockchain prevents the single point failure problem. The PoA consensus mechanism is a good choice as it suits the need of IoT devices being lightweight, fast, and secure. But the issues that arise with using blockchain must also be solved such as scalability, cost to make the system more efficient.

3.2 SEA Architecture with Smart Gateways

SEA is smart and efficient authentication and authorization architecture using distributed Smart e-Health Gateways. In SEA, Datagram Transport Layer Security (DTLS) handshake protocol is employed. The smart gateways autonomously perform local data storage and processing. By utilizing data aggregation, embedded machine learning, and compression techniques, smart gateways may quickly deliver preliminary results and decrease the redundant remote transmission to cloud servers. The main components of the SEA architecture are Medical Sensor Network, Smart e-Health Gateways, and Back-End system [17].

In [18], the authors try to prevent DDoS attack in Session Resumption SEA architecture by proposing a DDoS prevention system in session resumption-based end-to-end security. The SEA architecture with session resumption is used in the proposed. But this is vulnerable to DDoS attacks due to its dependence on resource-constrained sensors. To overcome this, the author utilizes fog-computing-based three-layer architecture. The results of the research demonstrate the effectiveness of the model in preventing replay and DDoS attacks and increasing the efficiency of the resource-constrained sensors by reducing the load on them.

Discussion

The SEA architecture with smart gateways can be used to increase IoT security. The smart gateways equipped with local databases decrease the response and storage time and better service. By using the session resumption scheme with SEA architecture, the DDoS attack vulnerability can also be solved [18].

3.3 Medical Image Encryption

In [19], the authors propose an image encryption scheme to protect sensitive medical data and compare it with other schemes. The proposed scheme uses three rounds of high-speed scrambling and pixel adaptive diffusion concept for shuffling random neighboring pixels. Arithmetic modulo operations are used to implement pixel adaptive diffusion. The results show that the cipher images have uniform pattern and information entropy close to 8.

In [20], the authors propose a medical image encryption framework. The proposed framework is based on Logistic equation, Hyperchaotic equation, and DNA encoding. To convert encrypted image to share, Lossless Computational Secret Image Sharing (CSIS) method is used. The parameters of the image are changed slightly by XORing Pseudo Random Numbers generated by logistic equation with image sequence. The results show that the encrypted cipher images are small, have uniform histogram, entropy of more than 7.99, and SSIM very close to 0. In future, the authors plan to integrate GPUs with IoT to alleviate drawbacks of CSIS such as time requirement and high computational complexity.

Discussion

Encryption of sensitive medical images is very important and can be easily implemented with other existing security systems without increasing computational cost much and overall robustness of the system can be increased.

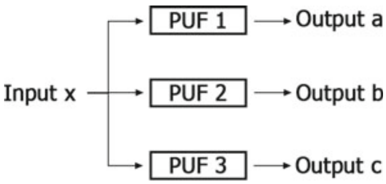


Fig. 3 Physical Unclonable Functions (PUFs)

3.4 Physical Unclonable Functions (PUFs)

PUFs are hardware security measures that are produced during IC manufacturing process. PUFs generate unique keys, and if same input is given to any number of PUFs, the outputs will be different and no two can be same as shown in Fig. 3. This feature makes them resistant to cloning attacks.

Authors in [21] try to overcome the problem of lack of physical security measures by proposing Healthcare Authentication protocol using Resource-constrained IoT devices (HARCI) a two-way stage authentication protocol using PUFs. The HARCI is a lightweight protocol which has three layers. For each stage of authentication, unique session keys are generated for end-to-end authentication. The results show that HARCI is secure against numerous cyber-attacks and takes less computation cost compared to other protocols.

In [22], the authors are also concerned about the various physical attacks and propose the development of RapidAuth, an authentication protocol based on Physical Unclonable Functions (PUFs). Elliptic Curve Cryptography is used for mutual authentication. The results show that the proposed framework has low computational complexity and communication overhead, high communication efficiency and reduces authentication delay.

Discussion

PUFs are a good way to solve cloning attacks as every PUF is unique. A problem with PUFs is that as it is produced during manufacturing process of the IC, it cannot be implemented with existing systems. Also, the physical damage to device is the most important problem to look after in PUFs.

3.5 Radio Frequency Identification (RFID)

In [23], as shown in Fig. 4, the authors mention the vulnerability and threats in an existing RFID authentication system and proposing the development of an improved security system based on RFID which overcomes the challenges in it. The methodology of the proposed system is implemented using RFID authentication scheme. For the enhancement, Elliptic Curve Cryptography (ECC) encryption and elliptic curve digital signature with message recovery are used. The results show that the proposed

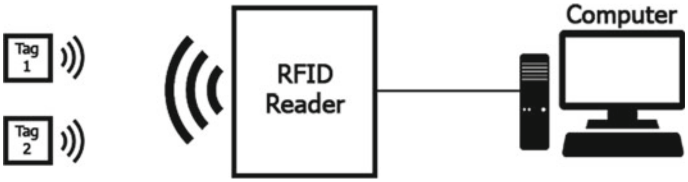


Fig. 4 RFID System which consists of tags and reader

system is resistant to numerous threats and attacks and has very low computation cost.

Discussion

RFID is a good approach for authentication and access control in security systems, but the electromagnetic waves can have effect with other devices and interfere with their working. Moreover, the initial cost of setting up RFID authentication system is high, and it suffers from cloning attacks.

3.6 Holochain

Holochain is an open-source distributed network infrastructure without the large-scale data interchange and storage requirements, like blockchain. Holochain uses two techniques—Distributed Hash Table (DHT) and hash chain. Holochain offers many benefits over blockchain such as more scalability, less traffic, low complexity, more efficient mechanisms, memory and time efficient, and cost-effective [24].

In [24], the authors propose to develop a holochain-based security and privacy framework more efficient than blockchain-based solutions. Blockchain causes duplicate processing overheads for IoT Healthcare systems with limited resources. Therefore, to overcome this problem, a practically viable solution is needed. It is crucial to guarantee that the DLT is implemented in the dispersed IoT Healthcare context in a way that is less complicated and resource-intensive, yet intrinsically safe and privacy-preserving. The results show that the memory and computation costs of holochain are much better than blockchain. In future, the authors plan to work on real-time cryptocurrency monitoring, threat detection, balancing of load and quick response to threat.

Discussion

Holochain can be an even better solution for security as it surpasses blockchain in many ways such as scalability, efficiency, but it needs sufficient testing and implementation before it is set up on a large scale.

3.7 Encryption Techniques

In [25], the authors mention the difficulty of encrypting data with complex cryptographic techniques and propose to develop an energy efficient block cipher technique which uses less computation power. The proposed technique is based on Matrix Rotation, Expansion function, and XoR. The performance analysis is done on an Arduino Uno board and the results show that the proposed technique takes almost equal time for encryption and decryption but uses significantly less amount of memory as compared to other cryptographic techniques.

In [26] also, the authors propose to develop a novel lattice-based secure cryptosystem for smart Health care (LSCSH). The LSCSH is built on an improvement to the BDH key exchange protocol which is vulnerable to forward secrecy attack. To deal with this, complex session keys are derived on both ends. The evaluation shows that computation and communication costs are 57.7 ms and 6400 bits, respectively. The proposed system is resilient to all kinds of attacks.

In [27], the authors propose the development of a Robust and Efficient Secure data Aggregation Scheme in Health care using the IoT (RESDA) which is energy efficient and lightweight. The RESDA scheme is implemented in four steps—Deployment, Encryption, Data aggregation, and Decryption. Homomorphic encryption is used in this scheme. The results show that it is very efficient in terms of energy consumption for its functions without compromising security. The comparative analysis indicates that it is a good choice for resource-constrained IoT devices. The potential direction for future research mentioned is the improvement of data aggregation in wireless networks.

Discussion

The methods and results of the techniques are summarized in Table 3.

Table 3 Encryption schemes analysis

References	Methods	Result
[25]	Matrix rotation	Lightweight
	Expansion function	Energy and time efficient
	XoR	
[26]	Lattice-based	Resilient to attacks
	BDH Key exchange	Low computation cost
[27]	Homomorphic Encryption	Lightweight

3.8 *Other Techniques*

In [28], the authors mention vulnerability in the token-based access-controlled methods in conventional multi-layered systems and propose to develop a token-based multi-layered security model which is two-way authentication centric. The proposed model is implemented on cloud level which completes in two steps. The results show that the proposed model has better security and less complexity as compared to other multi-layered systems, and even if number of resources increase, the time complexity remains almost same.

In [29], the authors propose to design an end-to-end security middleware for cloud-fog communication that is also flexible and has good data transfer performance at network edge. The proposed scheme consists of a core cloud, gateways, and edge IoT devices. There are two middle wares—primary and secondary. The primary middleware uses Session Resumption for intermittent security and static PSKs for encryption of data. The middleware is implemented on a GENI cloud testbed. The schemes compared are DTLS and TLS using PSKs and certificates. The results show that DTLS-PSK is the fastest scheme.

The authors in [30] propose a secure End-to-End key establishment protocol which is energy efficient and requires low computational power. The methodology is to use the neighboring trusted devices by offloading the heavy cryptographic functions to them. The results show that the proposed scheme is scalable, secure from various cyber-attacks, and energy and memory efficient. The future work includes implementing the scheme to actual hardware and analyze it practically.

In [31], the authors review Wireless Body Area Network (WBAN) and propose the development of a framework for ubiquitous Healthcare IoT devices. The proposed framework provides security on three levels based on sensor location. The results show that the proposed framework is secure and efficient and provides flexibility to the patients. Future research may include enhancing security in the proposed framework with private cloud network security.

In [32], the authors propose a Healthcare monitoring framework by using Named Data Networking (NDN) and Edge Cloud (IE) for efficient data retrieval. Hierarchical architecture is used to employ the proposed framework. A medical data caching algorithm uses NDNs in network caching and a medical data delivery algorithm to enable multiple users to retrieve data. The data communication latency and cost are reduced. But the system is dependent on the cluster head, and this can cause single point failure. The authors aim to solve this problem by cluster head re-election mechanism. The IoT Healthcare security techniques' summary is shown in Table 4.

Table 4 Summary of IoT Healthcare security techniques

Paper references	Technique	Advantages	Disadvantages
[7–16, 35]	Blockchain	Decentralized Data integrity and immutability Transparency	High memory and computational cost Scalability challenge Legal challenges
[17, 18]	SEA architecture with smart gateways	Increased speed Scalability	Single point failure Resource requirements
[19, 20, 36]	Medical image encryption	Sensitive data security Patients' privacy	Loss of data possible
[21, 22]	Physical unclonable functions (PUFs)	Uniqueness	Vulnerable to physical attacks [33]
[23]	RFID	Efficient tracking	High cost Effect of electromagnetic interference (EMI) on other devices [34]
[24, 37]	Holochain	Low computation and memory requirements More scalable as compared to blockchain	Lack of real-world implementation and testing
[25–27, 38]	Encryption schemes	Better authentication Confidentiality	Complexity

4 Conclusion

This study surveyed the security and privacy threats and their countermeasures. According to the review, holochain has great potential for providing security solution in IoT Health care, but it needs to be researched properly before implementation. The number of IoT devices are in billions and increasing day by day, and according to a research report by Precedence Research, the market size of IoT Health care is expected to reach over USD 900 billion by the year 2030. But still, there remain many challenges in the Healthcare sector for IoT. Attackers may find new ways to exploit privacy of patients. These issues need to be addressed and researchers should think of new challenges that could arise by new technologies and overcome them. IoT is an emerging field and has been expanding at a tremendous rate. The market opportunities are also growing and there is a great scope for researchers as well as entrepreneurs. In essence, IoT for Health care is not free from challenges, but proper research work can help boost the potential of the same.

5 Future Scope

The need for a standard protocol for communication can be a potential direction of research in making the Healthcare IoT more scalable and increase integration. Another budding area can be optimization of efficient security techniques which require more resources by making them able to work efficiently on resource-constrained IoT devices. Other works may include, but are not limited to, integration with the latest technologies such as AI and ML to increase its capability many folds.

References

1. Habibzadeh H, Dinesh K, Rajabi Shishvan O, Boggio-Dandry A, Sharma G, Soyata T (2020) A survey of healthcare internet of things (HIoT): a clinical perspective. IEEE Internet Things J 7:53–71. https://doi.org/10.1109/JIOT.2019.2946359
2. Saha G, Singh R, Saini S (2019) A survey paper on the impact of "Internet of Things" in healthcare. In: 2019 3rd international conference on electronics, communication and aerospace technology (ICECA), pp 331–334. IEEE. https://doi.org/10.1109/ICECA.2019.8822225
3. Iqbal W, Abbas H, Daneshmand M, Rauf B, Bangash YA (2020) An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. IEEE Internet Things J 7:10250–10276. https://doi.org/10.1109/JIOT.2020.2997651
4. Djenna A, Eddine Saidouni D (2018) Cyber attacks classification in IoT-based-healthcare infrastructure. In: 2018 2nd cyber security in networking conference (CSNet), pp 1–4. IEEE. https://doi.org/10.1109/CSNET.2018.8602974
5. Bhuiyan MN, Rahman MM, Billah MM, Saha D (2021) Internet of Things (IoT): a review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities. IEEE Internet Things J 8:10474–10498. https://doi.org/10.1109/JIOT.2021.3062630
6. Alam SR, Jain S, Doriya R (2021) Security threats and solutions to IoT using blockchain: a review. In: 2021 5th international conference on intelligent computing and control systems (ICICCS), pp 268–273. IEEE. https://doi.org/10.1109/ICICCS51141.2021.9432325
7. Lee Y-L, Lee H-A, Hsu C-Y, Kung H-H, Chiu H-W (2022) SEMRES—a triple security protected blockchain based medical record exchange structure. Comput Methods Programs Biomed 215:106595. https://doi.org/10.1016/j.cmpb.2021.106595
8. Chakraborty S, Aich S, Kim H-C (2019) A secure healthcare system design framework using blockchain technology. In: 2019 21st international conference on advanced communication technology (ICACT), pp 260–264. IEEE. https://doi.org/10.23919/ICACT.2019.8701983
9. Zulkifl Z, Khan F, Tahir S, Afzal M, Iqbal W, Rehman A, Saeed S, Almuhaideb AM (2022) FBASHI: fuzzy and blockchain-based adaptive security for healthcare IoTs. IEEE Access 10:15644–15656. https://doi.org/10.1109/ACCESS.2022.3149046
10. Bosri R, Uzzal AR, Al Omar A, Bhuiyan MZA, Rahman MS (2020) HIDEchain: a user-centric secure edge computing architecture for healthcare IoT devices. In: IEEE INFOCOM 2020—IEEE conference on computer communications workshops (INFOCOM WKSHPS), pp 376–381. IEEE. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162729
11. Alzubi JA (2021) Blockchain-based Lamport Merkle digital signature: authentication tool in IoT healthcare. Comput Commun 170:200–208. https://doi.org/10.1016/j.comcom.2021.02.002
12. Azbeg K, Ouchetto O, Jai Andaloussi S (2022) BlockMedCare: a healthcare system based on IoT, Blockchain and IPFS for data management security. Egypt Inform J 23:329–343. https://doi.org/10.1016/j.eij.2022.02.004

13. Johari R, Kumar V, Gupta K, Vidyarthi DP (2022) BLOSOM: Blockchain technology for security of medical records. ICT Exp 8:56–60. https://doi.org/10.1016/j.icte.2021.06.002
14. Saberi MA, Adda M, Mcheick H (2022) Break-glass conceptual model for distributed EHR management system based on Blockchain, IPFS and ABAC. Procedia Comput Sci 198:185–192. https://doi.org/10.1016/j.procs.2021.12.227
15. Al Ahmed MT, Hashim F, Jahari Hashim S, Abdullah A (2022) Hierarchical blockchain structure for node authentication in IoT networks. Egypt Inform J 23:345–361. https://doi.org/10.1016/j.eij.2022.02.005
16. Ren J, Li J, Liu H, Qin T (2022) Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT. Tsinghua Sci Technol 27:760–776. https://doi.org/10.26599/TST.2021.9010046
17. Moosavi SR, Gia TN, Rahmani A-M, Nigussie E, Virtanen S, Isoaho J, Tenhunen H (2015) SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. Procedia Comput Sci. 52:452–459. https://doi.org/10.1016/j.procs.2015.05.013
18. Rajagopalan A, Jagga M, Kumari A, Ali ST (2017) A DDoS prevention scheme for session resumption SEA architecture in healthcare IoT. In: 2017 3rd international conference on computational intelligence & communication technology (CICT), pp 1–5. IEEE. https://doi.org/10.1109/CIACT.2017.7977361
19. Khan J, Li J, Haq AU, Parveen S, Khan GA, Shahid M, Monday HN, Ullah S, Ruinan S (2019) Medical image encryption into smart healthcare IOT system. In: 2019 16th international computer conference on wavelet active media technology and information processing, pp 378–382. IEEE. https://doi.org/10.1109/ICCWAMTIP47768.2019.9067592
20. Sarosh P, Parah SA, Bhat GM, Muhammad K (2021) A security management framework for big data in smart healthcare. Big Data Res 25:100225. https://doi.org/10.1016/j.bdr.2021.100225
21. Alladi T, Chamola V (2021) Naren: HARCI: a two-way authentication protocol for three entity healthcare IoT networks. IEEE J Sel Areas Commun 39:361–369. https://doi.org/10.1109/JSAC.2020.3020605
22. Aman MN, Chaudhry SA, Al-Turjman F (2021) RapidAuth: fast authentication for sustainable IoT. https://doi.org/10.1007/978-3-030-69431-9_7
23. Izza S, Benssalah M, Drouiche K (2021) An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment. J Inf Secur Appl 58:102705. https://doi.org/10.1016/j.jisa.2020.102705
24. Zaman S, Khandaker MRA, Khan RT, Tariq F, Wong K-K (2022) Thinking out of the blocks: holochain for distributed security in IoT healthcare. IEEE Access 10:37064–37081. https://doi.org/10.1109/ACCESS.2022.3163580
25. Chaudhary RRK, Chatterjee K (2020) An efficient lightweight cryptographic technique for IoT based E-healthcare system. In: 2020 7th international conference on signal processing and integrated networks (SPIN), pp 991–995. IEEE. https://doi.org/10.1109/SPIN48934.2020.9071421
26. Chaudhary R, Jindal A, Aujla GS, Kumar N, Das AK, Saxena N (2018) LSCSH: lattice-based secure cryptosystem for smart healthcare in smart cities environment. IEEE Commun Mag 56:24–32. https://doi.org/10.1109/MCOM.2018.1700787
27. Soufiene BO, Bahattab AA, Trad A, Youssef H (2019) RESDA: robust and efficient secure data aggregation scheme in healthcare using the IoT. In: 2019 international conference on internet of things, embedded systems and communications (IINTEC), pp 209–213. IEEE. https://doi.org/10.1109/IINTEC48298.2019.9112125
28. Aski VJ, Gupta S, Sarkar B (2019) An authentication-centric multi-layered security model for data security in IoT-enabled biomedical applications. In: 2019 IEEE 8th global conference on consumer electronics (GCCE), pp 957–960. IEEE. https://doi.org/10.1109/GCCE46687.2019.9015217
29. Mukherjee, B., Neupane, R.L., Calyam, P.: End-to-End IoT Security Middleware for Cloud-Fog Communication. In: 2017 IEEE 4th international conference on cyber security and cloud computing (CSCloud), pp 151–156. IEEE. https://doi.org/10.1109/CSCloud.2017.62

30. Iqbal MA, Bayoumi M (2016) Secure end-to-end key establishment protocol for resource-constrained healthcare sensors in the context of IoT. In: 2016 international conference on high performance computing & simulation (HPCS), pp 523–530. IEEE. https://doi.org/10.1109/HPCSim.2016.7568379
31. Al Alkeem E, Yeun CY, Zemerly MJ (2015) Security and privacy framework for ubiquitous healthcare IoT devices. In: 2015 10th international conference for internet technology and secured transactions (ICITST), pp 70–75. IEEE. https://doi.org/10.1109/ICITST.2015.7412059
32. Wang X, Cai S (2020) Secure healthcare monitoring framework integrating NDN-based IoT with edge cloud. Futur Gener Comput Syst 112:320–329. https://doi.org/10.1016/j.future.2020.05.042
33. Helfmeier C, Boit C, Nedospasov D, Tajik S, Seifert J-P (2014) Physical vulnerabilities of physically unclonable functions. In: Design, automation & test in Europe conference & exhibition (DATE), pp 1–4. IEEE conference publications, New Jersey. https://doi.org/10.7873/DATE.2014.363
34. Haddara M, Staaby A (2018) RFID applications and adoptions in healthcare: a review on patient safety. Procedia Comput Sci 138:80–88. https://doi.org/10.1016/j.procs.2018.10.012
35. Saini GL, Panwar D, Singh V (2021) Software reliability prediction of open source software using soft computing technique. Recent Adv Comput Sci Commun (Formerly: Recent Patents on Computer Science) 14(2):612–621
36. Saini GL, Panwar D, Kumar S, Singh V, Poonia RC (2021) Predicting of open source software component reusability level using object-oriented metrics by Taguchi approach. Int J Software Eng Knowl Eng 31(02):147–166
37. Panwar D, Saini GL, Agarwal P, Singh P (2022) Firefly optimization technique for software quality prediction. In: Soft computing: theories and applications: proceedings of SoCTA 2021. Springer Nature Singapore, Singapore, pp 263–273
38. Panwar D, Saini GL, Agarwal P (2022) Human eye vision algorithm (HEVA): a novel approach for the optimization of combinatorial problems. In: Artificial intelligence in healthcare, pp 61–71

Hybrid Deep Learning Framework for Glaucoma Detection Using Fundus Images

Royce Dcunha, Aaron Rodrigues, Cassandra Rodrigues, and Kavita Sonawane

Abstract Glaucoma is a chronic eye condition that develops because intraocular pressure in the eye damages the visual nerve. One of the causes of blindness around the globe is due to it. Glaucoma does not initially cause vision loss, but if the condition worsens, it may leave a person permanently blind. Measurement of intraocular pressure, testing of the visual field, or inspection of the optical disc of fundus pictures are all methods used in the clinical setting to diagnose glaucoma. Early detection of glaucoma is crucial in reducing the risk of eye damage. VGG19, VGG19 + LSTM, Inceptionv3, and Inceptionv3 + LSTM are used to study the identification of glaucoma. ACRIMA is the dataset used, and it consists of 705 fundus images (396 glaucomatous images and 309 healthy images). The models are worked using data augmentation and K-fold cross-validation. The extracted features classify the input image as glaucomatous or healthy. The VGG19 + LSTM model performed the best out of all the models.

Keywords Deep learning · Machine learning · Artificial intelligence · Fundus image · ConvNet

1 Introduction

Glaucoma, a chronic eye condition, primarily caused by intraocular pressure, poses a significant threat to global eye health [12], affecting individuals worldwide [22]. This paper introduces a novel approach to early glaucoma detection, addressing the limitations of existing diagnostic techniques. In clinical settings, diagnosis traditionally involves measuring intraocular pressure, assessing visual fields, and examining fundus images. However, these methods are labor-intensive and may not be readily accessible to all populations. According to the World Health Organization (WHO), it currently affects over 80 million individuals globally, with an anticipated increase

R. Dcunha · A. Rodrigues (✉) · C. Rodrigues · K. Sonawane
St. Francis Institute of Technology, Borivali (West), Mumbai, India
e-mail: aaronaldo0605@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_14

of reaching 111.8 million people by 2040 [21]. High myopia, diabetes, eye surgery, and hypertension are all factors that contribute to this illness. The advantages and limitations of several papers are studied in this paper. We present a comparative analysis of four architectures: Inceptionv3, Inceptionv3 + LSTM, VGG19, and VGG19 + LSTM.

Each model undergoes K-fold cross-validation and data augmentation. This is done to overcome the limitation of having a small dataset [17]. The proposed models extract attributes from input fundus images [4] to classify them as healthy or glaucomatous. Performance evaluation parameters are used to compare the models and assess their accuracy in early glaucoma detection. In this study, we employ the ACRIMA dataset, comprising 705 fundus images with a 90–10 split for training and testing. The aim is to contribute to the early detection of ocular diseases, particularly glaucoma, offering a promising approach [19] to combat this widespread and potentially blinding condition.

2 Literature Review

Sharmila and Shanthi [1] investigated the application of transfer learning. The approach involved reusing a pre-existing model developed for one task as a starting point for another. The study used the Inceptionv3 model pre-trained on the ImageNet dataset for glaucoma diagnosis. Transfer learning was implemented using both feature extraction and fine-tuning techniques. The training and testing of the automated glaucoma diagnostic model used the ORIGA dataset. It achieved an accuracy of 91.36% in predicting the two classes: glaucoma and not glaucoma. The evaluation included sensitivity, specificity, and accuracy parameters, resulting in 82.60% sensitivity and 95.30% specificity. Higher accuracy was achievable with minimal training epochs due to the small dataset size. However, further studies using additional datasets are necessary to enhance the accuracy of the automated glaucoma detection algorithm.

In another study by Arkaja Saxena, Abhilasha Vyas, Lokesh Parashar, and Upendra Singh [2], they developed a deep learning-based architecture for reliable glaucoma diagnosis using CNNs. The CNNs provided a hierarchical visual structure for discriminating between glaucoma-affected and healthy human eyes. The suggested method of testing has consisted of six layers. Each of these layers included tens of thousands of photos. The usage of a dropout mechanism improved the approach's performance. The objective was to find the most comparable patterns in healthy and glaucoma-affected eyes. SCES and ORIGA were the datasets used. It yielded a detection rate of 82.2% for ORIGA and 88.2% for SCES. The results showed that the ORIGA dataset outperformed SCES in diagnosing stages of glaucoma.

Ali Serener and Sertan Serte [3] used deep convolution neural networks in the study to detect distinct stages of glaucoma in fundus pictures. The fundus pictures utilized in the study are either healthy (no glaucoma), early glaucoma, or advanced glaucoma. The classification used two deep learning models, which are ResNet50 and GoogLeNet. The models were trained using a single NVIDIA GeForce GTX 1080Ti GPU running the Caffe deep learning framework. The training was performed on a unique dataset to address the limitation of limited testing data. The RIM-ONE dataset assessed the performance of the ResNet50 and GoogLeNet models in accuracy, sensitivity, specificity, and area under the ROC curve. The results reveal that GoogLeNet beats ResNet50 for early, advanced, and total glaucoma detection.

Table 1 presents a comprehensive comparison of research papers focused on glaucoma. The analysis aims to identify the most promising model for achieving improved performance compared to existing approaches. We have highlighted key metrics and insights derived from these studies to guide our selection process.

Table 1 Literature review

No.	Algorithm and year of issue	Advantages	Datasets	Accuracy%	Evaluation parameters
1	Inceptionv3 (2021) [1]	Classification of early glaucoma	ORIGA	91.36	Sensitivity (Sen%): 82.60, Specificity (Spe%): 95.30
2	CNN (2020) [2]	Higher detection capability and high accuracy	SCES, ORIGA	82.2–88.2	Sen%: 21–29, Spe: 91–93
3	GoogLeNet and ResNet (2019) [3]	Large training dataset	RIM-ONE	85–86	Sen%: 21–29, Spe%: 91–93, ROC: 0.75
4	CNN + LSTM (2021) [4]	Reduced glaucoma prediction time and high accuracy	RIM-ONE, DRISHTI GS, DRIONS	Around 90	Sen%: 95.4, Spe%: 96.7, AUC: 0.984
5	MobileNet and Inceptionv3 (2021) [5]	High accuracy	ORIGA, SCES	86 for MobileNet and 90 for Inceptionv3	Precision, Recall, F1-score: 0.87–0.90, AUC: 0.831–0.887
6	Inceptionv3 (2021) [6]	Reduced Glaucoma prediction time and high accuracy	ACRIM-A, LAG	85.29	Sen%: 95.4, Spe%: 96.7, AUC: 0.984
7	AlexNet, SVM (2020) [7]	High accuracy	HRF, ORIGA	91.21	Sen%: 90.8, Spe%: 85
8	VGG, ResNet (2021) [8]	Uses backward propagation and forward LAG	LAG	80–87. Best accuracy: 86.9	Precision: 0.869
9	Mnet, DeNet (2019) [9]	Detection of datasets with different brightness	ORIGA SCES	Around 85	Sen%: 76–84, Spe%: 83

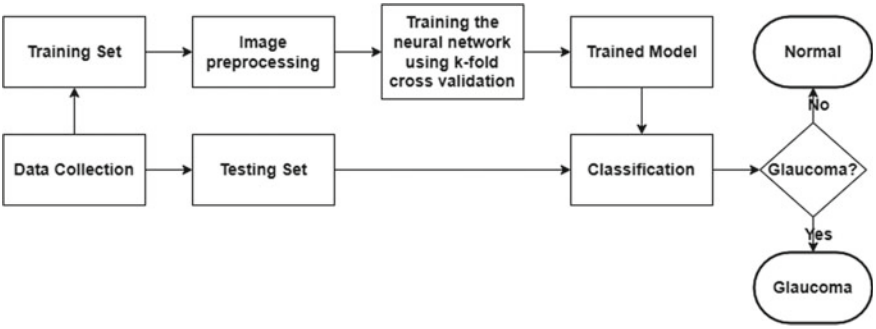


Fig. 1 Proposed architecture system

3 Methodology

3.1 Design

Figure 1 shows the system overview of the proposed architecture system. The data is collected and divided into a training set and a testing set. The images from the training dataset further undergo image preprocessing. The models are trained with K-fold cross-validation and data augmentation. The extracted features are used to classify the input image and projected to be either glaucomatous or normal.

3.2 Architecture

A comparison study is done between four architectures: Inceptionv3, Inceptionv3 + LSTM, VGG19, and VGG19 + LSTM. These architectures are chosen because they have much experience with medical image categorization.

Inceptionv3 Architecture. The purpose of Inceptionv3 is to reduce computing resource usage by modifying previous Inception models [14, 15]. It has a lower error rate compared to its predecessors and has 42 layers. Inception Networks (GoogLeNet/Inceptionv1) are computationally more efficient when compared to VGGNet, both in terms of the associated financial cost (memory and other resources) and the number of parameters generated by the network. The Inceptionv3 model can achieve more than 78.1% accuracy on the ImageNet dataset. Convolutions, average pooling, max pooling, concatenations, dropouts, and fully linked layers are the symmetric and asymmetric building components that make up the model itself [23].

VGG19 Architecture. A convolution neural network that has 19 layers is called VGG19. The VGG19 model contains 16 convolution layers, 5 MaxPool layers, 3 fully linked layers, and a sigmoid layer. The first 16 layers are convolution and maximum pooling layers and extract the spatial features. The final three layers are for image categorization. The input to a VGG19 network is a fixed RGB image with a size of (224 * 224). Hence, the matrix has a shape of (224, 224, 3). For instance, in 112 * 112 * 128, 128 is the number of filters or kernels, and 112 * 112 is the size. The convolution layers' filter size is 3 * 3, and the stride is 1. The max pooling layers' filter sizes are 2 * 2, and the stride is 2. After the final pooling layer, 7 * 7 * 512 volume flattens into a fully connected (FC) layer with 4096 channels. The sigmoid classifies an image as glaucomatous or normal [24].

VGG19 + LSTM and Inceptionv3 + LSTM Architecture. Spatial and temporal data extraction is done by a combination of CNN (Inceptionv3 and VGG19) and RNN (LSTM) architecture [25]. Recurrent neural networks are a particular kind of artificial network with loops that enable data storage. RNNs use knowledge of older events to make predictions. The completion of tasks that use visual sequences, therefore, necessitates a sophisticated architecture. As a result, the CNN-RNN architecture is chosen [16]. The CNN extracts spatial data from each video by converting it into consecutive pictures. The outputs detect temporal properties inside the image sequence using a recurrent sequence learning model (LSTM). Finally, the aggregated features are sent to a fully linked layer that predicts categorization for the input sequence [26].

3.3 Working of the Project

Figure 2 illustrates the architecture of the proposed model used in this paper. This method for glaucoma detection combines the capabilities of the Long Short-Term Memory (LSTM) recurrent neural network and the VGG19 convolutional neural network (CNN). This hybrid design enables glaucoma detection through the analysis of ocular pictures. The overview of the algorithm is below:

Data Preprocessing

- Obtained a collection of eye photographs classified as having glaucoma or not.
- Created training and testing sets from the dataset.

VGG19 Model

- Loaded the pre-trained VGG19 model and eliminated the fully connected layers.
- Froze the convolutional layer weights to preserve the learned features.
- Included a new connected layer with appropriate classification units.
- Compiled the model using the proper optimizer and loss function.

LSTM Model

- Designed an LSTM model for sequence classification.
- Preprocessed the eye image data to create sequential input data for the LSTM.
- Defined the LSTM architecture with a suitable input shape and LSTM units.

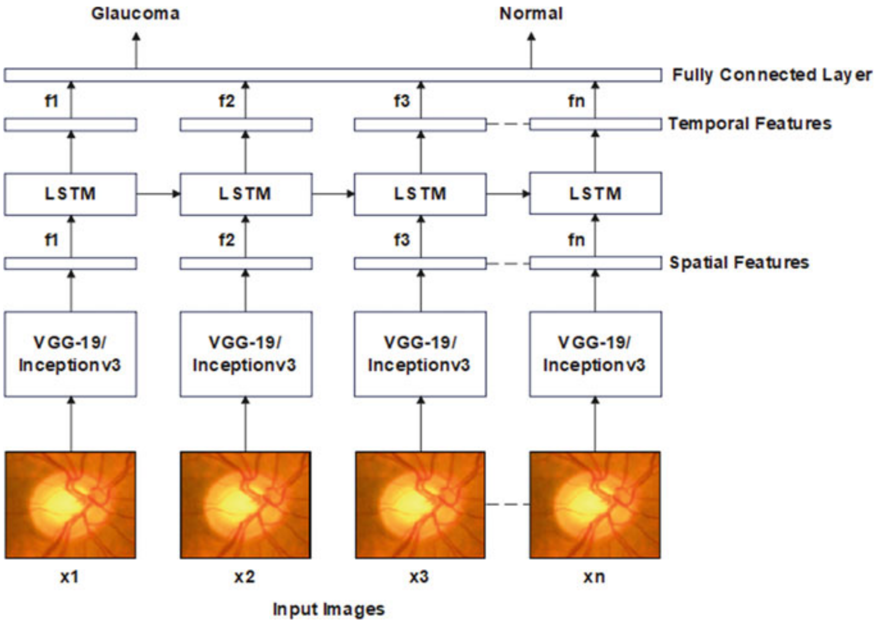


Fig. 2 VGG19 + LSTM and Inceptionv3 + LSTM architecture [20]

- Added fully connected layers and output layers for classification.
- Compiled the LSTM model with an appropriate loss function and optimizer.

Training

- Fed the eye images into the VGG19 model to extract visual features.
- Used these features as input to the LSTM model for glaucoma detection.
- Trained the combined VGG19 + LSTM model using the labeled training data

Testing and Evaluation

- Evaluated the trained model using the labeled testing data.
- Calculated performance metrics such as accuracy, precision, recall, and F1-score.

Prediction

- Used the trained model to predict the presence of glaucoma for new eye images.
- Preprocessed the new eye image and extracted visual features with VGG19.
- Fed these features into the LSTM model to obtain the glaucoma prediction.

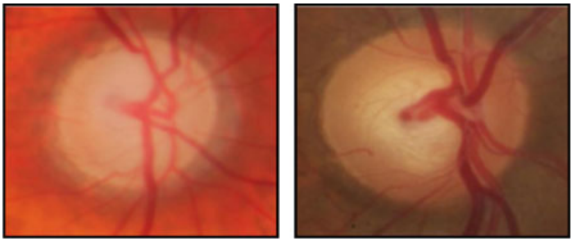


Fig. 3 Glaucomatous image

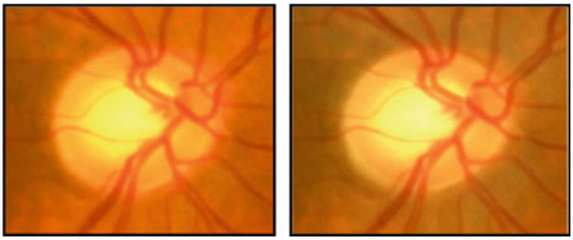


Fig. 4 Normal image

4 Results and Discussions

4.1 Experimental Setup

The ACRIMA dataset is used for this research. It contains 705 fundus images (396 glaucomatous and 309 normal images). For training, 632 images are used. For testing, 73 images are used. The dataset has a 90–10 split [10]. The sample glaucoma images are displayed in Fig. 3. The sample normal images are displayed in Fig. 4.

4.2 Performance Evaluation Parameters

Accuracy. The proportion of the correct predictions to the overall number of predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}. \tag{1}$$

F1-Score. The harmonic mean of the precision and recall of the model.

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{2}$$

Precision. It divides the fraction of true positives by the total number of true positives and false positives.

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

Recall. It is determined by dividing the total number of true positives and false negatives by the number of true positives and false negatives.

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

4.3 Results

A few samples have been taken from the ACRIMA and RIM-ONE datasets and tested against the proposed model and the results are in Table 2.

VGG19 model observation. The model exhibited a precision of 0.85 for normal images and 1.00 for glaucoma images. The recall rates were 1.00 for normal images and 0.84 for glaucoma images. Regarding the F1-score, normal images scored 0.92, while glaucoma images scored 0.91. Overall, the model attained an accuracy of 91.78%. The model performed better on normal images.

VGG19 + LSTM model observation. The model exhibited a precision of 0.90 for normal images and 1.00 for glaucoma images. The recall rates were 1.00 for normal images and 0.89 for glaucoma images. Regarding the F1-score, normal images scored 0.95, while glaucoma images scored 0.94. Overall, the model attained an accuracy of 94.52%. The model performed better on normal images.

Inceptionv3 model observation. The model exhibited a precision of 0.88 for normal images and 0.92 for glaucoma images. The recall rates were 0.91 for normal images and 0.90 for glaucoma images. Regarding the F1-score, normal images scored 0.90,

Table 2 Test case results

Test no.	Test case (actions performed)	Expected result	Actual outcome
1	ACRIMA dataset sample	Normal	Normal
2	RIM-ONE dataset sample	Normal	Glaucomatous
3	ACRIMA dataset sample	Glaucomatous	Glaucomatous
4	ACRIMA dataset sample	Glaucomatous	Glaucomatous
5	RIM-ONE dataset sample	Glaucomatous	Glaucomatous

Table 3 Performance results of the proposed models

Model	Class	Precision	Recall	F1-score	Accuracy (%)
VGG19	Normal	0.85	1.00	0.92	91.78
VGG19	Glaucoma	1.00	0.84	0.91	91.78
VGG19 + LSTM	Normal	0.9	1.00	0.95	94.52
VGG19 + LSTM	Glaucoma	1.00	0.89	0.94	94.52
Inceptionv3	Normal	0.88	0.91	0.9	90.41
Inceptionv3	Glaucoma	0.92	0.9	0.91	90.41
Inceptionv3 + LSTM	Normal	0.91	0.94	0.93	93.15
Inceptionv3 + LSTM	Glaucoma	0.95	0.93	0.94	93.15

while glaucoma images scored 0.91. Overall, the model attained an accuracy of 90.41%. The model performed better on glaucoma images.

Inceptionv3 + LSTM model observation. The model exhibited a precision of 0.91 for normal images and 0.95 for glaucoma images. The recall rates were 0.94 for normal images and 0.93 for glaucoma images. Regarding the F1-score, normal images scored 0.93, while glaucoma images scored 0.94. Overall, the model attained an accuracy of 93.15%. The model performed better on glaucoma images.

The highest precision is for the VGG19-based models for glaucoma images at 0.91. Likewise, the VGG19 + LSTM model achieved the highest recall for normal images at 1.00. The highest F1-score is achieved by the VGG19 + LSTM model, with a score of 0.95 for normal images. Ultimately, the VGG19 + LSTM model achieved the highest overall accuracy of 94.52%.

The VGG19-based models perform better on normal images, and Inceptionv3-based models perform better on glaucoma images. Each row in Table 3 corresponds to various approaches used during training. The k-fold cross-validation technique is for training purposes with $k = 3$. After training the model for 35 epochs, the average training accuracy achieved was 96 and 99% for VGG19-based and Inceptionv3-based models, respectively. The average testing accuracy was 94.52 and 93.15% for VGG19-based and Inceptionv3-based models, respectively. The addition of LSTM to VGG19 and Inceptionv3 helped to increase the training as well as testing accuracy. The VGG19 + LSTM model has the best results with the highest accuracy.

Table 4 compares the accuracy of the proposed approach with the existing approaches. It is noticeable that the proposed method has resulted in a higher accuracy compared to the other existing approaches to detect glaucoma.

Table 4 Comparison study between various approaches

Author	Model	Dataset	Accuracy (%)
Sharmila et al. [1]	Inceptionv3	ORIGA	91.36
Saxena et al. [2]	CNN	SCES	82.20
Saxena et al. [2]	CNN	ORIGA	88.20
Serner et al. [3]	ResNet	RIM-ONE	86.00
Serner et al. [3]	GoogLeNet	RIM-ONE	85.00
Proposed approach	VGG19	ACRIMA	91.78
Proposed approach	VGG19 + LSTM	ACRIMA	94.52
Proposed approach	Inceptionv3	ACRIMA	90.41
Proposed approach	Inceptionv3 + LSTM	ACRIMA	93.15

5 Conclusion

Conclusion. The mechanism used is a combination of CNN and RNN. The models used are VGG19, VGG19 + LSTM, Inceptionv3, and Inceptionv3 + LSTM. After performing a comparative study on the ACRIMA dataset, it is noticeable that VGG19 + LSTM is the best-performing model. The VGG19 + LSTM model predicted two classes (glaucoma and non-glaucoma) with an accuracy of 94.52%. Accurate results are due to transfer learning and data augmentation. Data augmentation helped us to tackle the problems while training CNNs. The VGG19-based models perform better on normal images, and Inceptionv3-based models perform better on glaucoma images. This project aims to provide an effective, fast, and efficient solution for exposure to glaucoma in the eye. Because a glaucoma detection model plays a crucial role in early identification, efficient screening, and improved management of glaucoma cases, it has the potential to save vision, reduce healthcare costs, and enhance overall eye healthcare delivery.

Future Scope. This model will be beneficial to both ophthalmologists and patients. Real-life data and larger datasets could be collected from hospitals all over India to analyze any changes in the retina due to different geographical locations. Also, we can work on the severity categories of glaucoma [13]. While these advancements show promise, further research, validation, and regulatory approvals are mandatory before widespread implementation in clinical practice [11, 18].

References

1. Sharmila C, Shanthi N (2021) Retinal image analysis for glaucoma detection using transfer learning. In: Advances in electrical and computer technologies. Springer, Singapore, pp 235–244. https://doi.org/10.1007/978-981-15-9019-1_21

2. Saxena A, Vyas A, Parashar L, Singh U (2020) A glaucoma detection using convolutional neural network. In: 2020 international conference on electronics and sustainable communication systems (ICESC), pp 815–820. https://doi.org/10.1109/ICESC48915.2020.9155930
3. Serener A, Serte S (2019) Transfer learning for early and advanced glaucoma detection with convolutional neural networks. In: 2019 medical technologies congress (TIPTEKNO), pp 1–4. https://doi.org/10.1109/TIPTEKNO.2019.8894965
4. Demir F, Taşcı B (2021) An effective and robust approach based on R-CNN+LSTM model and NCAR feature selection for ophthalmological disease detection from fundus images. J Personal Med 11(12):1276. https://doi.org/10.3390/jpm11121276
5. Olivas LG, Alférez GH, Castillo J (2021) Glaucoma detection in Latino population through OCT's RNFL thickness map using transfer learning. Int Ophthalmol 41(11):1–15. https://doi.org/10.1007/s10792-021-01931-w
6. Afroze T, Akther S, Chowdhury MA, Hossain E, Hossain MS, Andersson K (2021) Glaucoma detection using inception convolutional neural network V3. Springer International Publishing. Cham, pp 17–28. https://doi.org/10.1007/978-3-030-82269-9_2
7. Ajitha S, Judy MV, Meera N, Rohith N (2020) Automated identification of glaucoma from fundus images using deep learning techniques. Eur J Mol Clin Med 7(2)
8. Sallam A, Gaid ASA, Saif WQA, Kaid HAS, Abdulkareem RA, Ahmed KJA, Saeed AYA, Radman A (2021) Early detection of glaucoma using transfer learning from pre-trained CNN models. In: 2021 international conference of technology, science and administration (ICTSzA), pp 1–5. https://doi.org/10.1109/ICTSA52017.2021.9406522
9. Fu H, Cheng J, Xu Y, Liu J (2019) Glaucoma detection based on deep learning network in fundus image. In: Advances in computer vision and pattern recognition, pp 119–137. https://doi.org/10.1007/978-3-030-13969-8_6
10. Diaz-Pinto A, Morales S, Naranjo V, Köhler T, Mossi J, Navea A (2019) CNNs for automatic glaucoma assessment using fundus images: an extensive validation. BioMed Eng OnLine 18(1). https://doi.org/10.1186/s12938-019-0649-y
11. Garg H, Gupta N, Agrawal R, Shivani S, Sharma B (2022) A real-time cloud-based framework for glaucoma screening using EfficientNet. Multimed Tools Appl 81(24):34737–34758. https://doi.org/10.1007/s11042-021-11559-8
12. Li L, Xu M, Liu H, Li Y, Wang X, Jiang L, Wang Z, Fan X, Wang N (2020) A large-scale database and a CNN model for attention-based glaucoma detection. IEEE Trans Med Imag 39(2):413–424. https://doi.org/10.1109/TMI.2019.2927226
13. Dhillon A, Verma GK (2020) Convolutional neural network: a review of models, methodologies, and applications to object detection. Prog Artif Intell 9:85–112. https://doi.org/10.1007/s13748-019-00203-0
14. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: 2016 IEEE conference on computer vision and pattern recognition
15. Pattern Recognition (CVPR) Las Vegas, NV, USA, pp 2818–2826. https://doi.org/10.1109/CVPR.2016.308.
16. Islam MZ, Islam MM, Asraf A (2020) A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using x-ray images. Inform Med Unlocked 20:100412. https://doi.org/10.1101/2020.06.18.2013471
17. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. https://doi.org/10.48550/arXiv.1207.0580
18. Tu Z, Bai X (2010) Auto-context and its application to high-level vision tasks. IEEE Trans Pattern Anal Mach Intell 32(10):1744–1757. https://doi.org/10.1109/TPAMI.2009.186
19. Xu Y, Duan L, Lin S, Chen X, Wong DWK, Wong TY, Liu J (2014) Optic cup segmentation for glaucoma detection using low-rank superpixel representation. In: Golland P, Hata N, Barillot C, Hornegger J, Howe R (eds) Medical image computing and computer-assisted intervention (MICCAI) 2014, Part I. LNCS, vol 8673, pp 788–795. Springer, Heidelberg. https://doi.org/10.1007/978-3-319-10404-1_98

20. Cheng J, Liu J, Wong DWK, Yin F, Cheung CY, Baskaran M, Aung T, Wong TY (2011) Automatic optic disc segmentation with peripapillary atrophy elimination. In: IEEE international conference engineering in medicine and biology society, pp 6224–6227. https://doi.org/10.1109/IEMBS.2011.6091537
21. Gheisari S, Shariflou S, Phu J, Kennedy PJ, Agar A, Kalloniatis M, Golzan SM (2021) A combined convolutional and recurrent neural network for enhanced glaucoma detection. Sci Rep 11(1):1945. https://doi.org/10.1038/s41598-021-81554-4
22. Epidemiology of Glaucoma: The past, present, and predictions for the future. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7769798/. Accessed 3 Sept 2021
23. Glaucoma: causes, types, symptoms, diagnosis, and treatment. WebMD, WebMD. https://www.webmd.com/eye-health/glaucoma-eyes. Accessed 21 Sept 2021
24. Narein AT (2021) Inception V3 model architecture. OpenGenus IQ: Computing Expertise & Legacy, OpenGenus IQ: Computing Expertise & Legacy, 8 Oct 2021. https://www.iq.opengenus.org/inception-v3-model-architecture/. Accessed 23 Nov 2021
25. Kaushik A (2020) Understanding the VGG19 architecture. OpenGenus IQ: computing expertise & legacy. https://iq.opengenus.org/vgg19-architecture/. Accessed 16 Dec 2021
26. Brownlee J (2019) CNN long short-term memory networks, 14 Aug 2019. https://www.machinelearningmastery.com/cnn-long-short-term-memory-networks/. Accessed 14 Jan 2022

Sunflower Optimization with Elite Learning Strategy (SFO-ELS) for Antenna Selection in Massive MIMO Subarray Switching Architecture

Snehal Gaikwad and P. Malathi

Abstract Massive MIMO is a promising technology used by fifth generation of wireless technology to increase the channel capacity significantly. But the use of RF transceivers for every antenna at the base station increases hardware complexity and implementation cost of the system making it very challenging for deployment. This paper focuses on addressing the hardware complexity and cost challenges associated with Massive Multiple-Input Multiple-Output systems. To mitigate these challenges, an efficient antenna selection algorithm is essential for identifying a subset of antennas that contribute maximum to the channel capacity. By employing advanced antenna selection schemes, this study aims to identify the most effective approach for optimizing antenna selection in Massive MIMO technology. The Sunflower Optimization algorithm, combined with the elite learning strategy, offers a novel approach to antenna selection in Massive MIMO subarray switching architecture. It leverages the benefits of both the SFO algorithm and the elite learning strategy to improve the selection of antennas, thereby optimizing system performance. Sunflower Optimization with elite learning strategy has been proposed and evaluated the effectiveness of the simulated SFO algorithm by comparing it with traditional approaches. The result shows that the proposed method for antenna selection significantly improves upon other methods offering a more efficient and effective approach for enhanced channel capacity in a Massive MIMO system.

Keywords Massive MIMO · Antenna selection · Sunflower optimization · Elite learning strategy · Branch and bound · Bit error rate · Signal-to-noise ratio

S. Gaikwad (✉) · P. Malathi
D. Y. Patil College of Engineering, Pune, India
e-mail: snehal.academics@gmail.com

P. Malathi
e-mail: pjmalathi@dypcoeakurdi.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_15

1 Introduction

1.1 Massive MIMO

In the recent years, pandemic has changed everyone's life in several ways either it is work or education. The extensive acceptance of remote work by IT organizations and development of online courses which offers flexible learning opportunities to everyone have gained significantly. As a result, people's demand for high data rate increases all over the world for various purposes. To fulfill this increasing demand, the next generation of technology uses Massive Multiple-Input Multiple-Output (mMIMO) as a vital technology for enhancing data rates. Massive MIMO offers significant improvements in data transmission capabilities of wireless communication which transforms the connectivity in the digital era [1].

Massive MIMO technology has arisen as a solution in wireless communication by significantly expanding the capabilities of conventional MIMO by several orders of magnitude. By arranging a large number of antennas at the base station (BS), typically in the order of hundreds, massive MIMO enables multi-user MIMO, where a base station is capable of serving multiple single-antenna terminals simultaneously. This brings several benefits including improved connection reliability, enhanced spectrum quality, and increased energy efficiency through effective use of radiated energy [2, 3].

Figure 1 depicts a Massive MIMO system with A_T transmitting and A_R receiving antennas [3]. The output of the MIMO system is given by the following equation [3]:

$$R_y = H.X_i + n, \tag{1}$$

where R_y is $1 \times A_R$ receiving matrix, X_i is $1 \times A_T$ transmitting matrix, H is $A_R \times A_T$ channel matrix, and n is channel noise. The channel capacity C_n of Massive MIMO system is given by [3]

$$C_n = \log_2 \det\left(I_{\mathrm{NT}} + \rho H^H H\right), \tag{2}$$

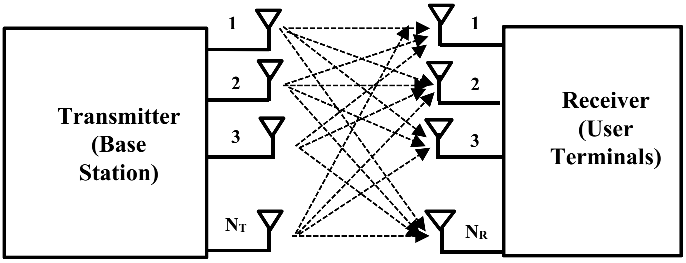


Fig. 1 Massive MIMO system [3]

where I_{NT} is an identity matrix, ρ is the signal-to-noise ratio, and H is the channel matrix.

1.2 Antenna Selection in Massive MIMO

This extension of receiving or transmitting antennas and connected radio frequency (RF) chains at the BS introduces practical limitations, complexity, and cost implications. RF chains containing mixers, analog-to-digital converters, and amplifiers require a substantial portion of the base station's transceiver power. To overcome these challenges and optimize system performance, it becomes necessary to reduce the number of RF transceiver chains [4]. By using optimal antenna selection (AS), it becomes possible to choose a specific set of antenna elements from the pool of available antennas at the base station. This subset increases the channel capacity to its maximum potential under ideal channel conditions, while still benefiting from the advantages provided by a full antenna system.

Figure 2 illustrates the antenna selection process in a Massive MIMO system [4]. The A_S number of antennas are selected from the total A_T antennas which is determined based on the number of RF chains employed in the system. The channel matrix denoted as H consists of $A_R \times A_T$ coefficients which represents channel gain for all the antennas. On the other hand, the selected channel matrix H_S is a subgroup of H, containing channel gain only for selected subset of antennas.

The channel capacity C_S for A_S selected antennas can be calculated as [5]

$$C_s = \log_2 \det\left(I_{\text{NT}} + \rho H_s^H H_s\right), \tag{3}$$

where H_s represents the channel matrix of A_S selected antennas.

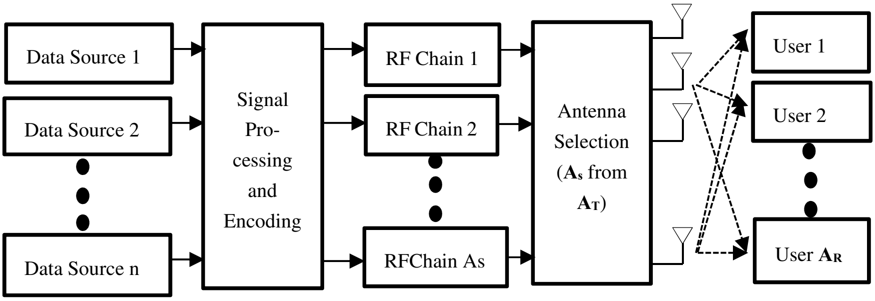


Fig. 2 Antenna selection in Massive MIMO system [4]

1.3 Subarray Switching Architecture

Antenna selection is a proficient technique used by Massive MIMO to improve energy efficiency and reduced system complexity. Additionally, a switching system known as Subarray Switching system is utilized, where RF switches link the selected antenna to one of the RF chains. It offers significant advantages over the full array antenna architecture. In the subarray switching antenna architecture, a single RF switch is utilized to connect one antenna from a subarray. This design reduces the number of required RF switches significantly compared to the full array antenna architecture. The number of RF switches in subarray architecture is equal to the number of subarrays, whereas, in the full array architecture, it is equal to the product of the number of transmitting or receiving antennas and the number of subarrays. By minimizing the number of RF switches, the subarray switching architecture simplifies the system design and reduces power consumption. Moreover, it enhances the overall performance by enabling efficient connectivity between subarrays and RF chains [6].

In search of efficient antenna selection techniques for Massive Multiple-Input Multiple-Output (MIMO) systems, researchers have explored the application of intelligent searching algorithms to overcome challenges associated with local optima of search spaces.

2 Literature Survey

In the literature, the authors have contributed multiple papers that explore antenna selection algorithms for both conventional and Massive MIMO systems, providing a comprehensive analysis of their performance parameters.

The paper [7] introduces an antenna selection algorithm for Massive MIMO systems, aiming to achieve high channel capacity while minimizing computational complexity. The algorithm utilizes optimization techniques such as bidirectional Branch and Bound (BAB) searching to identify the globally optimal solution for the channel submatrix that has the largest maximum singular value. But the complexity goes on increasing with the increase in a number of antennas [7]. In the paper [8], authors implemented antenna selection techniques which utilize a computationally efficient greedy search algorithm known as matching pursuit (MP). It reduces the computational complexity associated with the antenna selection process significantly, but may not always provide the optimal subset of antennas in environments with spatially correlated channels. In the papers [9, 10], the authors introduced an antenna selection algorithm employing both Branch and Bound and greedy search methods, combined with a subarray switching architecture. The proposed Subarray-based Antenna Selection Scheme (SAS) reduces the hardware and optimize energy efficiency by minimizing the number of RF switches used to connect antennas to the base station.

Intelligent searching algorithms offer the advantage of avoiding local optima by exploring the search space more effectively. These algorithms utilize various optimization techniques inspired by natural processes and population-based strategies. In the paper [11], researchers developed two algorithms for antenna selection in a Massive MIMO system, namely Artificial Bee Colony (ABC) and Tree Search (TS). A comparative analysis of these algorithms was conducted with the popular genetic algorithm (GA) and Particle Swarm Optimization (PSO) methods, showing their capacity performance and complexity. The TS technique performed better than all the bio-inspired algorithms, exhibiting significantly lower complexity while achieving higher capacity compared to GA, PSO, and ABC algorithms. In the paper [12], a Biogeography-Based Optimization algorithm was implemented for the selection of antennas in Multiple-Input Multiple-Output system to maximize the channel capacity. This algorithm is applied for both transmitter and receiver antenna selections. The results of BBO algorithm demonstrate its efficiency and applicability as compared with the results of ant colony and genetic algorithm. In the paper [13], the authors proposed an antenna selection scheme based on a genetic algorithm, and also a heuristic beam-forming technique is used to obtain high performance with a low computation complexity. Genetic algorithms offer the flexibility in their complexity and precision as needed just by adjusting the iterative volume. In case of time-sensitive scenario, fewer iterations can save time at the cost of accuracy, but more iterations can improve accuracy.

Sunflower Optimization is one of the metaheuristic optimization algorithms used to solve various optimization problems which is inspired by the natural behavior of sunflowers. It shows the promising results in terms of convergence rate and solution quality as compared to other optimization algorithms [14, 15]. The paper [16] introduced improved Sunflower Optimization algorithm for the clustering process in IoT networks, where the Sunflower Optimization (SFO) algorithm was combined with the levy flight operator. This algorithm demonstrated its superiority in terms of energy consumption and the network's lifetime.

The proposed work has primarily concentrated on implementing an antenna selection algorithm utilizing the Sunflower Optimization technique with elite learning in order to attain the maximum sum rate or channel capacity for Massive MIMO systems. Also, the research work focuses on utilizing the Subarray Switching (SAS) architecture for antenna selection algorithm to improve upon the power efficiency of the system.

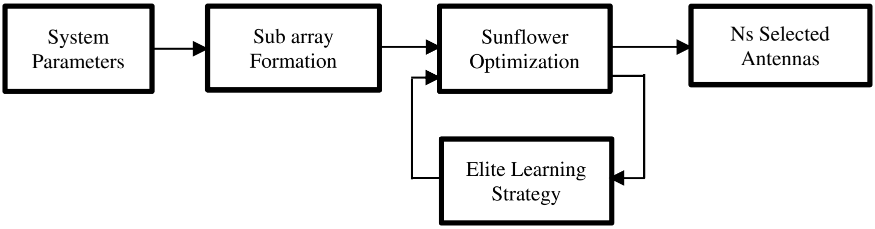


Fig. 3 Block diagram of proposed system

3 Proposed System

3.1 Block Diagram (Fig. 3 Shows)

3.2 Sunflower Optimization

Sunflower Optimization (SFO) is a metaheuristic optimization algorithm inspired by the behavior of sunflowers in nature. This algorithm works on the idea where sunflowers adjust their florets position toward sun which maximizes their exposure to sunlight. The SFO algorithm utilizes a sunflower representation to solve optimization problems by presenting the solution in a unique manner. The center of the sunflower represents the best solution found so far, and the florets represent candidate solutions. The floret position in each iteration is updated based on the position of the center and the position of the neighboring florets [16].

The basic principle of the SFO algorithm is to replicate the movement of sunflowers to catch the best sun following, which is done every morning at sunrise. The life cycle of a sunflower is focused by the law of radiation [16].

$$H_u = \frac{S_x}{4\pi d_u^2}, \tag{4}$$

where H_u is the Heat Energy, S_x is the solar energy, d_u is the distance from the best flower in the current population. Each sunflower adjusts its position to the sun [16].

$$\overrightarrow{S_u} = \frac{P^* - P_U}{\|P^* - P_U\|}, U = 1, 2, \ldots N_p, \tag{5}$$

where P^* represents the best flower in the current population, P_u corresponds to each individual solution, and NP denotes the population size.

The incorporation of Sunflower Optimization (SFO) algorithm with antenna selection is used to identify a subset of antennas from the available antennas at the base station which maximizes the system performance.

3.3 Elite Learning Strategy

The elite learning strategy is a technique used by evolutionary algorithms and metaheuristic algorithms during the optimization process to achieve a better balance between exploration and exploitation of the search space. Primary purpose of elite learning strategy is to preserve best solutions to enhance the algorithm's performance by giving them higher priority and influence outcome of subsequent iterations. These elite solutions play an important role in each iteration. They guide the search process by influencing the generation of new candidate solutions, thus enhancing the overall performance of the optimization process [17].

3.4 Sunflower Optimization with Elite Learning Strategy (SFO-ELS)

In the proposed work, the elite learning strategy has been employed in algorithm Sunflower Optimization named as SFO-ELS to enhance the quality of the search process of selected antennas in a Massive MIMO system. This strategy operates by maintaining a set of elite solutions, typically the best-performing solutions encountered during the search process. These elite solutions are given preferential treatment and are preserved throughout the iterations. In the antenna selection process, the elite learning strategy helps the algorithm to identify subsets of antennas by considering and refining the elite solutions during each iteration, offering superior performance in terms of metrics such as sum rate, capacity, or energy efficiency [18].

The SFO-ELS algorithm offers a novel approach to antenna selection in Massive MIMO subarray switching architecture. It leverages the benefits of both the SFO algorithm and the elite learning strategy to improve the selection of antennas, thereby optimizing system performance. By further integrating the SFO-ELS algorithm into the subarray switching architecture, significant improvements in energy efficiency can be achieved. This approach enables efficient utilization of antenna resources, leading to enhanced performance and reduced hardware complexity in Massive MIMO systems.

3.5 Algorithm

The following are the algorithmic steps to achieve optimal solution by combining Sunflower Optimization with elite learning strategy (SFO-ELS).

Step 1: Initialization: Initialize the system parameters Nr, Ns, pollination factor (p), mortality rate (m), Lower bound ($\mathrm{L_b}$), and Upper bound ($\mathrm{U_b}$) for channel capacity.

Step 2: Assignment: Initialize the population of candidate solutions using the SFO algorithm, randomly assigning antennas to each solution.

Step 3: Evaluate fitness: Evaluate the fitness of each candidate solution using an objective function that reflects the channel capacity performance [6].

$$C = \log_2\left(\det\left(I_{\mathrm{Ns}} + \frac{\mathrm{SNR}}{N_s}\left(H_s^H H_S\right)\right)\right), \tag{6}$$

where N_s is a number of selected antennas, I_{NS} is the identity matrix. Select the maximum fitness value as a base value $f_{\min}$.

Step 4: Identify elite solutions: Select the top-performing candidate solutions as elite solutions based on their fitness scores.

Step 5: Sunflower Optimization: Perform iterations of the SFO algorithm, updating the positions of the candidate solutions based on the SFO equations [13].

$$h_u = \lambda q_u (P_u + P_{u-1}).(P_u + P_{-1}), \tag{7}$$

where λ is inertial displacement. Adjust the exploration and exploitation parameters, such as step size, displacement factor, angle increment, and radius increment, to balance the search process.

Step 6: Evaluate fitness and select solutions: After each iteration, evaluate the new fitness f_{new} using Eq. (3) of the updated candidate solutions, including both the elite solutions and the non-elite solutions generated through SFO. If $f_{\mathrm{new}} > f_{\min}$, then replace the previous solution with the new one.

Step 7: Select the best solutions from the updated population based on their fitness values. This selection process can be biased toward preserving the elite solutions while considering the performance of the newly generated solutions.

Step 8: Termination: Check termination conditions such as the number of iterations and the number of selected antennas to decide whether to stop the algorithm.

Step 9: If the termination condition is not satisfied, the process returns to Step 4 and continues with the iterations of the SFO algorithm.

4 Result and Discussion

The simulation results are obtained using MATLAB simulation software. The simulation results demonstrate the effectiveness of the SFO-ELS algorithm in achieving optimal antenna selection in the subarray switching architecture.

Figure 4 illustrates the graph of the sum rate achieved by the SFO-ELS algorithm with an increase in a number of selected antennas. The number of selected antennas is varied from 0 to 64. From the graph, it is observed that there is a significant improvement in the sum rate as the number of selected antennas increases. This observation aligns with the fundamental principle of Massive MIMO systems, where increasing the number of antennas contributes to enhanced system performance. The graph also demonstrates the effect of an increase in signal-to-noise ratio. With the increase in SNR value, sum rate increases. The maximum value of the sum rate of 29.12 bps/Hz has been achieved for Nr = 64 and SNR = 10 dB.

Figure 5 shows the graph representing the sum rate performance across a range of values for Ns, denoting the number of selected antennas. It has been observed from the graph that there exists a positive correlation between the number of antennas selected and the achieved sum rate. This includes variations in the number of selected antennas, from 4 to 32, while the total number of receiving antennas remains constant as 64.

Figure 6 shows the improvement in the sum rate as the signal-to-noise ratio (SNR) and the number of receiving antennas (Nr) increase. The highest sum rate is attained when Nr equals 64 and SNR equals 20 dB. This observation emphasizes that the sum

Fig. 4 Performance of sum rate versus Ns with varying SNR

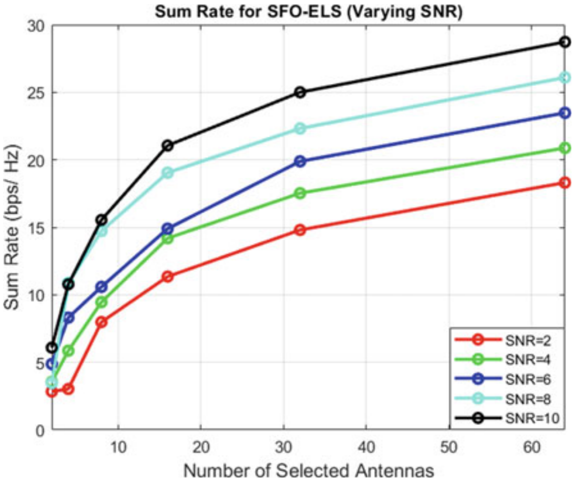


Fig. 5 Performance of sum rate versus Nr with varying Ns

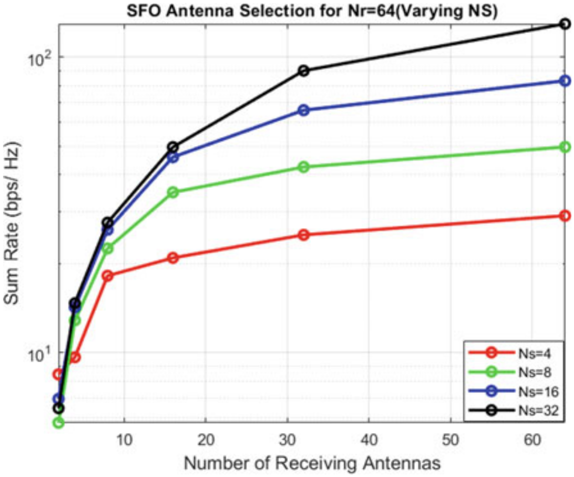
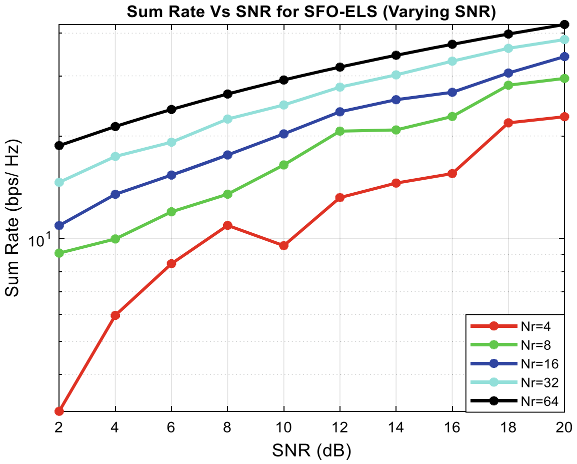


Fig. 6 Performance of sum rate versus SNR with varying Nr



rate experiences a consistent increase with higher values of both receiving antennas and signal-to-noise ratio.

Figure 7 shows the subarray antenna structure of 64 receiving antennas for SNR = 10 dB with different configurations based on the number of selected antennas, denoted as Ns. The structure consists of red and blue colored dots, where the red dots represent the positions of the selected antennas. The change in the number of columns in the subplots indicates the number of subarrays and it is determined by the value of Ns. For instance, when Ns = 16, there are 16 columns, each representing a subarray containing four antennas. The red-colored dots within each column indicate the locations of the selected antennas, with one red dot per column, representing one selected antenna out of the four antennas in each subarray.

Figure 8 provides a comprehensive representation of the sum rate performance across a broad range, spanning from Nr = 8 to Nr = 256. This result offers a thorough insight into how the sum rate improves with the number of selected antennas, the number of receiving antennas, and the signal-to-noise ratio, collectively contributing to a comprehensive understanding of the system's performance.

Figure 9 shows the spectral efficiency performance achieved by various antenna selection algorithms, namely Sunflower Optimization with elite learning strategy (SFO-ELS), Sunflower Optimization (SFO), MCNN, and MCTS for various values of signal-to-noise ratio (SNR) varying from 1 to 20 dB. The results demonstrate that the SFO-ELS algorithm, utilizing an elite learning strategy, exhibits improved performance compared to other antenna selection algorithms. The algorithm achieves optimal solutions in terms of maximum sum rate.

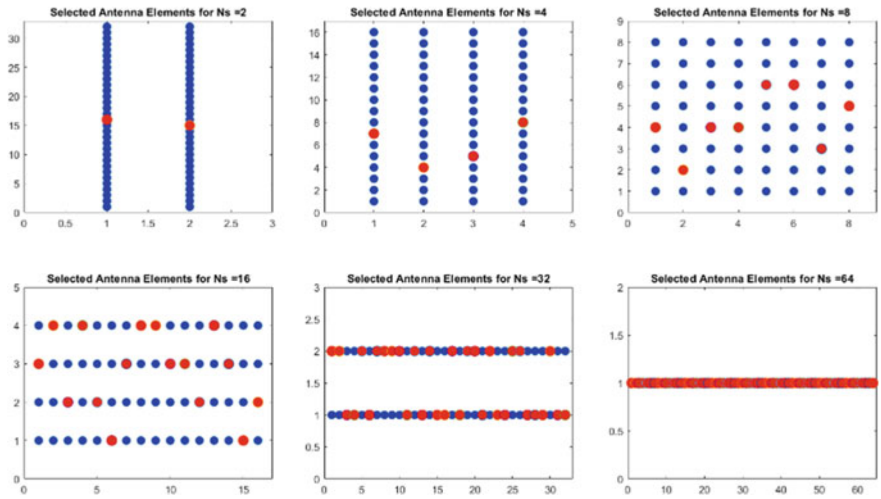


Fig. 7 Subarray antenna structure for Nr = 64

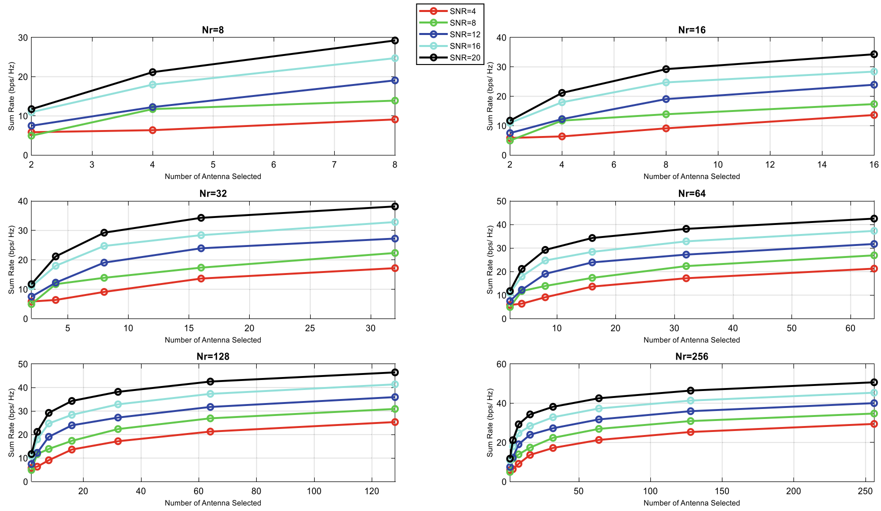
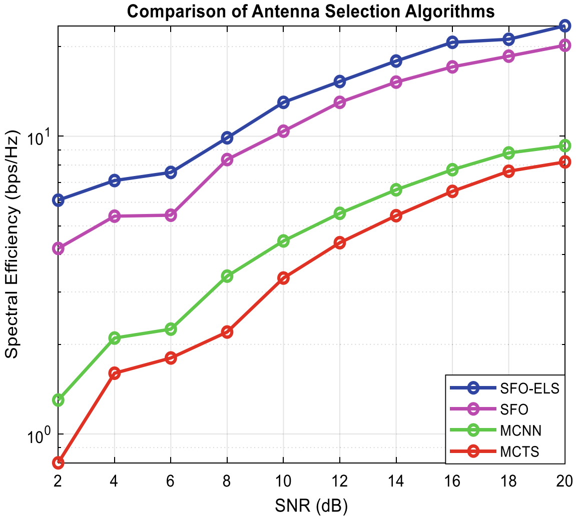


Fig. 8 Sum rate performance for Nr = 8 to Nr = 256

Fig. 9 Comparison graph of various antenna selection algorithms



5 Conclusion

The Sunflower Optimization Algorithm with elite learning strategy (SFO-ELS) demonstrates significant potential for the antenna selection process in Massive MIMO systems. Through the incorporation of the SFO algorithm with an elite learning strategy, the SFO-ELS algorithm offers a novel approach that enhances the antenna selection process. The algorithm intelligently explores the search space, using the elite learning strategy to preserve and prioritize the best-performing solutions encountered during the search process. This approach allows the algorithm to efficiently identify the best solutions that maximize system performance in terms sum rate, spectral efficiency, and energy efficiency. Overall, the SFO-ELS algorithm offers a valuable approach for optimizing antenna selection in Massive MIMO systems and contributing to the advancement of wireless communication technologies.

6 Future Scope

Researchers might explore advanced optimization algorithms or hybrid approaches that combine SFO-ELS with other optimization techniques to further enhance the efficiency and effectiveness of antenna selection. This could lead to improved better solution quality, and adaptability to different system scenarios. The integration machine learning techniques could provide more intelligent and adaptive antenna selection. In future, it could be used to predict optimal antenna configurations as per adaptive channel conditions, traffic patterns, and user requirements.

References

1. Varshney R, Jain P, Vijay S (2018) Massive MIMO systems in wireless communication. In: 2nd international conference on micro-electronics and telecommunication engineering 2018, pp 39–44. IEEE, Ghaziabad
2. Gheorghe C, Dragomir R, Alexandru D (2019) Massive MIMO technology for 5G adaptive networks. In: 11th international conference on electronics, computers and artificial intelligence, ECAI-2019, pp 1–4. IEEE, Romania
3. Surya KNR, Prasad V, Hossain E, Bhargava VK (2017) Energy efficiency in massive MIMO-based 5G networks: opportunities and challenges. IEEE Trans Wirel Commun 24(3):86–94
4. Saddam Hussain Sk, Yaseen SM, Barman K (2016) An overview of massive MIMO system in 5G. Int J Circ Theor Appl 9(11):4957–4968
5. Gao Y, Vinck H, Kaiser T (2018) Massive MIMO antenna selection: switching architectures, capacity bounds, and optimal antenna selection algorithms. IEEE Trans Signal Process 66(5):1346–1360
6. Gao Y, Jiang W, Kaiser T (2015) Bidirectional branch and bound based antenna selection in massive MIMO systems. In: 26th annual international symposium on personal, Indoor, and mobile radio communications (PIMRC), pp563–568. IEEE, Hong Kong
7. Wang BH, Hui HT, Leong M-S (2010) Global and fast receiver antenna selection for MIMO systems. IEEE Trans Commun 58(9):2505–2510
8. Mendonç MOK, Diniz PSR, Ferreira TN, Lovisolo L (2020) Antenna selection in massive MIMO based on greedy algorithms. IEEE Trans Wirel Commun 19(3):1868–1881
9. Gao Y, Kaiser T (2016) Antenna selection in massive MIMO systems: full-array selection or subarray selection? In: IEEE sensor array multichannel signal processing workshop, pp 1–5. IEEE, Brazil
10. Azeem H, Ullah MA (2019) Sub-array based antenna selection scheme for massive MIMO in 5G. In: International conference on cyber-living, cyber-syndrome and cyber-health, pp 38–50. Springer, Beijing
11. Abdullah Z, Tsimenids CC, Johnston M (2016) Tabu search Vs. bio-inspired algorithms for antenna selection in spatially correlated massive MIMO uplink channels. In: 24th European signal processing conference (EUSIPCO), pp 41–45. IEEE, Budapast
12. Fountoukidis KC, Siakavara K (2017) Antenna selection for MIMO systems using biogeography-based optimization. In: International workshop on antenna technology: small antennas, innovative structures, and applications (iWAT), pp 319–322. IEEE, Athens
13. Du L, Li L, Xu Y (2016) A genetic antenna selection algorithm with heuristic beamforming for massive MIMO systems. In: 19th international symposium on wireless personal multimedia communications, pp 49–52. IEEE, Shenzhen
14. Mohamed A, Shaheen M, Hasanien HM, Mekhamer SF, Talaat HEA (2019) Optimal power flow of power systems including distributed generation units using sunflower optimization algorithm. IEEE Access 7:109289–109300
15. Gomes GF, Cunha SSD, Ancelotti AC (2019) A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Springer J Eng Comput 35(5):619–626
16. Raslan AF, Ali AF, Darwish A, El-Sherbiny HM (2021) An improved sunflower optimization algorithm for cluster head selection in the internet of things. IEEE Access 9:156171–156186
17. Shaheen AM, Elattar EE, El-Sehiemy RA, Elsayed AM (2021) An improved sunflower optimization algorithm-based Monte Carlo simulation for efficiency improvement of radial distribution systems considering wind power uncertainty. IEEE Access 9:2332–2344
18. Zhan Y, Li Y, Zhao H, Zhou H (2020) Adaptive multi-objective particle swarm optimization based on competitive learning. In: 11th international conference on prognostics and system health management, pp 226–231. IEEE, Jinan, China

Machine Learning Models for Human Activity Recognition: A Comparative Study

Anshul Sheoran, Ritu Boora, and Manisha Jangra

Abstract With the advancements in machine learning, human activity recognition has found its applications in several emerging areas such as robotics healthcare, surveillance, smart environment etc. This paper aims to study and evaluate the performance of some popularly used machine learning algorithms in classifying human activities. We have selected K-NN, SVM and XGBoost methods in this study and the performance of the methods has been evaluated for 19 different activities which were performed by eight random persons. The required data was recorded using 5 MTx 3-DOF orientation trackers. The raw data was processed before feature extraction and then fed as input to the machine learning models. On performance comparison of these methods, it has been found that the SVM method when implemented with a polynomial kernel, outperforms the other state-of-the-art methods. It classified the different activities with an accuracy of 96.9%.

Keywords Machine learning · Extreme gradient boosting (XGBoost) · Principal component analysis (PCA) · K-nearest neighbour (K-NN) · Human activity recognition (HAR) · Support vector machine (SVM)

1 Introduction

Human Activity Recognition is a method of recognizing and identifying human actions. Due to the advent of technology, the demand for activity recognition (AR) is rapidly increasing and its applications are all around. Now it has become vital to social domains such as healthcare [1], sports science, security, and robotics. In healthcare, it can be used for monitoring purposes such as at nursing homes [2, 3], for detecting abnormal activities of a person with mental disorders [4] and many more. It assists the athlete in adopting a more data-driven training and performance approach, resulting in better outcomes and fewer injuries in sports [5]. Moreover, HAR is also

A. Sheoran · R. Boora (✉) · M. Jangra
Guru Jambheshwar University of Science and Technology, Hisar, India
e-mail: rituboora@gjust.org

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_16

used by smart surveillance and safety devices [6] which are significantly required for military purposes worldwide [7]. Although much research has already been done on human activity recognition, it remains a complex and daunting task because it needs to address the challenge of ambiguous interpretability where human activities encompass diverse movements and intensities like dancing, jumping, playing etc. Additionally, recognizing simultaneous activities such as eating food and watching television concurrently poses a significant hurdle in activity recognition systems. This paper presents a comparative study of the popularly used HAR algorithms and measures their performance on a given dataset.

Reference [8] conducted a comprehensive study on various variants of the K-nearest neighbour (K-NN) algorithm for disease prediction. Through implementations and experiments on benchmark datasets, different K-NN variants were analyzed. It employs a fine-tuned VGGNet-19 for feature extraction and a multi-class support vector machine for classification. In [9], a DIA-XGBoost model is proposed for identifying malicious URL-based assaults on social communication platforms. The suggested methodology addresses feature imbalance issues with effective feature extraction and selection, as well as a better strategy for managing concept drift and machine learning-based classification. Moreover, [10, 11] papers showcase the effectiveness of the XGBoost algorithm in domains. The author in [10] focused on disease prediction, highlighting its potential for accurate predictions and disease prevention. In [12] the author introduced a lightweight deep learning network for real-time Human Activity Recognition using WiFi Channel State Information (CSI), reducing computational complexity. Making it suitable for remote HAR applications.

From the literature, it is observed activity recognition has seen tremendous growth over the last decade. However, despite significant research on human activity recognition, existing techniques quite often misclassify human actions.

This paper is organized as follows: Sect. 2 explains the implementation of HAR using machine learning models. Section 3 discusses the performance parameters and performance evaluation of the machine learning models over a range of tunable parameters. The performance analysis and comparison of the discussed methods with their optimum parameter is presented in Sect. 4.

2 Human Activity Recognition Using Machine Learning

Human Activity Recognition with the help of machine learning is an emerging research area. It aims to develop machine learning algorithms and models to classify human activities based on either sensor data or video data. Activity recognition can broadly be implemented in these steps: data acquisition, pre-processing, feature extraction and recognition. These steps are illustrated in Fig. 1 and are crucial for accurate and reliable activity recognition.

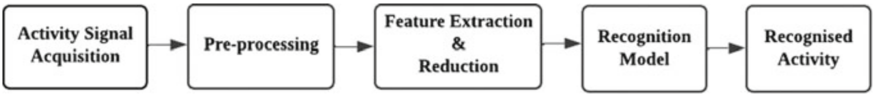


Fig. 1 Steps of human activity recognition

2.1 Data Acquisition

To carry out this study, a benchmark dataset from [13–15]consists of five MTx 3-DOF (Degree of Freedom) orientation trackers for motion sensing was used. Eachnd a 3-axial accelnetometer, a gyroscope, and a 3-axial accelerometer to measure and track changes in the orientation of a body in 3-D space.

Raw data has been gathered from 8 individuals who performed 19 activities namely sitting (S1), standing (S2), lying on their back (L1) and the right side (L2), ascending (A1), and descending stairs (D1), standing in an elevator (S3), and moving around (S4), walking in a parking lot (W1), walking on a treadmill with flat (W2) and inclined positions (W3), running on a treadmill (R1), exercising on a stepper (E1), exercising on a cross trainer (E2), cycling on an exercise bike in horizontal (C1) and vertical positions (C2), rowing (R2), jumping (P1), and playing basketball (P2). The collected dataset contains 45 recordings per subject per activity obtained from 5 sensor units each with three tri-axial sensing devices. Each recording is of 5 min duration. Subsequently, the dimension of the raw dataset is 7500 × 45 sampled at 25 Hz.

2.2 Pre-processing

In data Pre-processing, the raw data is formatted and structured in such a way that it becomes suitable for modelling and training.

Pre-processing includes the segmentation of the data into multiple frames, called segments [16]. In this work, we segmented the 5-min signal acquired by each sensor's units in 5-s frames. Each signal segment is a discrete sequence of N_s samples where N_s=125 (5 × 25). The segmented dataset contains 45 signals per subject per activity (9 persons × 5 sensors). So, the dimension of each segment per subject per activity is 125 × 45.

2.3 Feature Extraction

Feature extraction is a process of extracting relevant features or attributes from the primary data such that the selected data can be used as input for Machine Learning model training and testing [17]. In this work, we have applied the following feature extraction techniques on the segmented dataset $\boldsymbol{x_i} = [x_{1i}, x_{2i}, x_{3i}, x_{4i}, \ldots x_{Nsi}]^T$ for ith signal. Five statistical features are obtained: variance, mean value ($\mu_{\boldsymbol{x}}$), skewness, kurtosis, minimum and maximum values, represented by Eqs. (1–4) respectively where x_{ji} is the j th term of signal $\boldsymbol{x_i}$ and $E\{\cdot\}$ denotes the expectation operator.

$$\text{mean}(\boldsymbol{x_i}) = E\{\boldsymbol{x}\} = \mu_{\boldsymbol{x}} = \frac{1}{N_s}\sum_{j=1}^{N_s} x_{ji}, \tag{1}$$

$$\text{Variance}(\boldsymbol{x_i}) = \sigma^2 = E\left\{(\boldsymbol{x} - \mu_{\boldsymbol{x}})^2\right\} = \frac{1}{N_s}\sum_{j=1}^{N_s}\left(x_{ji} - \mu_{\boldsymbol{x}}\right)^2, \tag{2}$$

$$\text{skewness}(\boldsymbol{x_i}) = \frac{E\left\{(\boldsymbol{x} - \mu_{\boldsymbol{x}})^3\right\}}{\sigma^3} = \frac{1}{N_s\sigma^3}\sum_{j=1}^{N_s}\left(x_{ji} - \mu_{\boldsymbol{x}}\right)^3, \tag{3}$$

$$\text{Kurtosis}(\boldsymbol{x_i}) = \frac{E\left\{(\boldsymbol{x} - \mu_{\boldsymbol{x}})^4\right\}}{\sigma^4} = \frac{1}{N_s\sigma^4}\sum_{j=1}^{N_s}(x_{ji} - \mu_{\boldsymbol{x}})^4. \tag{4}$$

In this study, the mentioned five statistical features were extracted from each of 45 signals per segment per activity and cascaded to form a column vector of dimension 225 × 1 per segment. Along with these, five Discrete Fourier Transform (DFT) peaks of the $\boldsymbol{x_i}$ were obtained with the corresponding frequencies. Therefore 225 (5 × 45) Fourier peaks and 225 frequencies of corresponding peaks were extracted from each segment generating a feature vector of dimension 450 × 1 from each segment. 11 autocorrelation samples including the first sample and every 5th sample up to the 50th sample were extracted from each of 45 signals in a segment leading to 495 features per segment.

After feature extraction processes, a total of 1170 (225 + 225 + 225 + 475) features are extracted from each 5-s signal segment. The dimension of the dataset retrieved from feature extraction techniques is 9120 × 1170 as a total of 9120 (8 × 19 × 60) instances are available in the dataset from 8 subjects, 19 activities and 60 segments per subject per activity.

2.4 Feature Reduction

Dealing with high dimensional datasets can be computationally intensive leading to problems such as overfitting of data, increased runtime, and memory requirements. Subsequently, different dimensionality reduction techniques have been popular among the researchers. In this work, a set of 1170 features has been reduced to a smaller set of 30 features using Principal Component Analysis (PCA). PCA is a multivariate technique which selects the principal features from the dataset such that its covariance is preserved.

2.5 Recognition Models

In this study, we have used 3 Machine learning models: K-Nearest Neighbour (K-NN), Support Vector Machines (SVM), and XGBoost Algorithm.

K-Nearest Neighbour Algorithm

K-Nearest Neighbour is an effective and one of the simplest machine learning algorithms. It works on the principle that similar data points are likely to have similar labels. In this, K stands for the number of considered neighbours while predicting a new data point. Euclidean distance is used for calculating the similarity measures with other data Points [18]. K-NN algorithm can model non-linear decision boundaries, which means that it can handle complex classification tasks. The value of hyperparameter K is crucial and sensitive in deciding the performance of the classifier [19].

Support Vector Machines (SVM)

It is a supervised algorithm based on a statistical approach to find a decision boundary called Hyperplane. This plane maximally separates the n-dimensional feature space into two classes [20]. The optimal hyperplane is chosen such that the margin, which is the distance between the hyperplane and the nearer data points of each class, is maximized (see Fig. 2 depicting linear data classification of two feature vectors X1 and X2). Margin-defining data points that lie nearest to the hyperplane are called support vectors. While dealing with non-linearly separable problems, SVM uses a special function known as the kernel. Kernal maps the data to a higher dimensional feature space where it becomes linearly separable [21]. In this study, various SVM kernel functions, including linear, radial basis function (RBF), sigmoid and polynomial, have been used to perform Support Vector Classification.

Extreme Gradient Boosting (XGBoost)

XGBoost is an ensemble learning algorithm. It combines the predictions of multiple weak models to create a stronger optimized model. It is based on a gradient-boosting framework where errors in existing models are minimized by sequentially adding new models to the ensemble. While training, the loss function, which is the difference between predicted and actual values, is minimized [22].

As its name suggests, it is a powerful method that can effectively handle complex and large datasets. The key feature of XGBoost is its tunability of hyperparameters which enables users to fine-tune the model for optimal performance. It allows tuning of several parameters such as $gamma$ (width control), $maximum\_depth$, $n\_estimators$ (number of trees in the model) and $learning\_rate$.

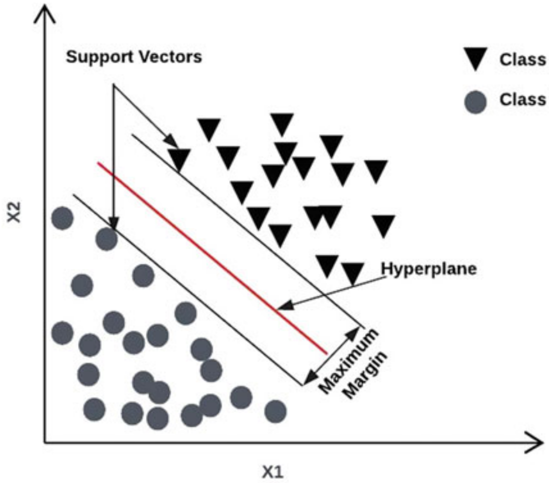


Fig. 2 Linear data classification of feature vectors X1 and X2 by SVM

3 Experimental Analysis

For the validation of the model's performance, *P-fold* cross-validation technique is used which involves dividing a dataset into subsets. In the *P-fold* technique, (*P-1*) subsets are used for training a model while the remaining subset is used for testing and the process is repeated *P* times to obtain a more reliable result. In this paper, for cross-validation*, P* is selected as 10.

Here, performance of different models is evaluated based on the accuracy. It can be calculated by using Eq. (5) where TP, FP, FN, and TN denote True Positive, False Positive, False Negative and True Negative respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5}$$

3.1 K-*Nearest Neighbour*

The K-NN algorithm is implemented with different values of K, where K varies from 3 to 19. The performance metrics for different K are plotted in Fig. 3. Results indicate that the highest mean accuracy (91.92%) of the K-NN model is achieved for K equal to 5 and decreases with a further increase in K. The drop in accuracy is mainly due to the overfitting of data for $K > 5$. Thus, the best value of K is found to be 5 corresponding to maximum mean accuracy.

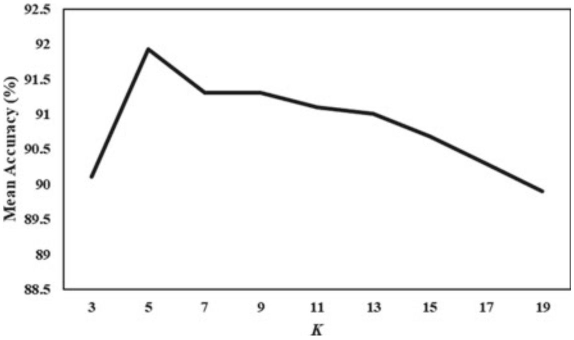


Fig. 3 Performance assessment of K-NN over a range of K

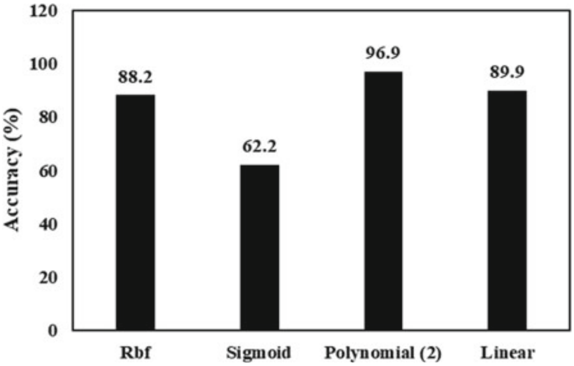


Fig. 4 Performance assessment of SVM with different types of Kernels

3.2 Support Vector Machines

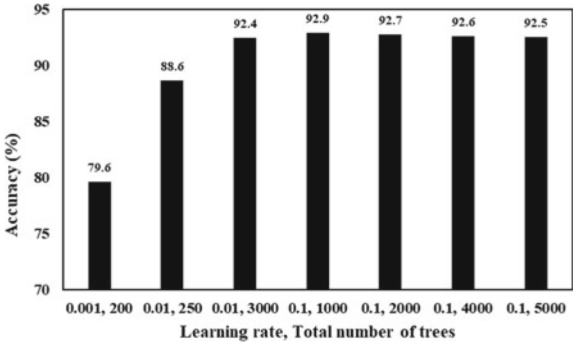
The SVM models are trained with different SVM kernel functions, including linear, radial basis function (RBF), sigmoid and polynomial of 2nd degree.

It can be observed from Fig. 4 that the SVM model for the polynomial kernel with degree 2, outperformed the other functions and achieved the highest classification accuracy of 96.9%. For the other kernel functions, the accuracy values are in the order Linear > RBF > Sigmoid.

3.3 XGBoost

For XGBoost, models are trained with different values of $n\_estimators$ and $Learning\_rate$ while keeping the $maximum\_depth$ equal to 3. Results depicted in Fig. 5 demonstrate that a peak accuracy of 92.9% is achieved with 0.1 $Learning\_rate$ and 1000 $n\_estimators$. Accuracy reduced for higher values of $n\_estimators$ due to overfitting of data.

Fig. 5 Performance of XGBoost Model over a range of tunable total number of trees and learning rate



3.4 General Performance Comparison

In this section, a performance analysis of best-performing models from each ML algorithm is carried out namely K-NN with K value 5, Polynomial SVM of degree 2, and XGBoost with 0.1 $Learning\_rate$ and 1000 total number of trees. The performance of these models has been tested on the above-described database.

The results from Fig. 6 show that the Support Vector Machine model with degree 2 polynomial kernel outperformed the other classification models with an accuracy of 96.9%. Whereas, a 1% difference is between the accuracy values of XGBoost and K-Nearest Neighbour. Furthermore, Table 1 in the study illustrates the overall confusion matrix of the SVM model. In the matrix, a higher true positive value means the model correctly predicted more activities. Meanwhile, having fewer false negatives also indicates accurate predictions, resulting in a higher accuracy score. The robust performance of the SVM model with a polynomial kernel can be attributed to its ability to effectively differentiate and classify the various human activities.

Fig. 6 Performance comparison of classification models

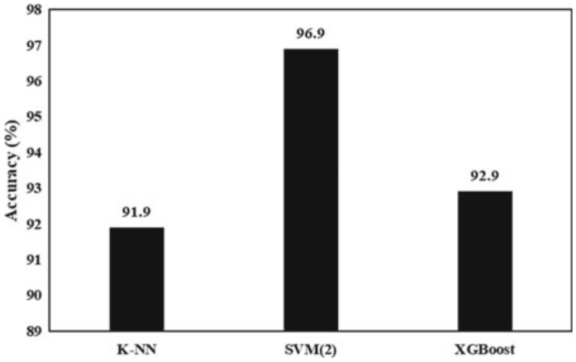


Table 1 Confusion Matrix of SVM Model with Polynomial Kernel and degree as 2

True	Predicted activity by SVM model																		
Label	S1	S2	L1	L2	A1	D1	S3	S4	W1	W2	W3	R1	E1	E2	C1	C2	R2	P1	P2
S1	448	0	30	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
S2	0	453	0	0	0	0	23	4	0	0	0	0	0	0	0	0	0	0	0
L1	31	0	444	3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
L2	1	1	3	475	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A1	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	13	72	3	1	0	0	380	11	0	0	0	0	0	0	0	0	0	0	0
S4	4	20	2	2	0	0	36	416	0	0	0	0	0	0	0	0	0	0	0
W1	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0
W2	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0
W3	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
R1	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
E1	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0
C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
R2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
P1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
P2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480

4 Conclusion

This research study presents a comparative analysis of different machine learning-based algorithms for classifying human activities. The algorithms were evaluated using a dataset comprising 19 different human activities. Comparatively, the classification accuracy of the XGBoost algorithm was only approximately 1% higher than that of the *K*-Nearest Neighbours (*K*-NN) algorithm but significantly lower than that of the SVM model. The results indicate that the Support Vector Machine (SVM) model with a polynomial kernel achieved the highest accuracy of 96.9% in classifying the activities. Therefore, the SVM model with a polynomial kernel outperformed other state-of-the-art methods in accurately classifying human activities.

References

1. Liu X, Liu L, Simske SJ, Liu J (2016) Human daily activity recognition for healthcare using wearable and visual sensing data. In: IEEE international conference on healthcare informatics, ICHI 2016, pp 24–31. https://doi.org/10.1109/ICHI.2016.100
2. Jalal A, Kamal S, Kim D (2014) A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments. Sensors 14:11735–11759. https://doi.org/10.3390/s140711735
3. Xu H, Pan Y, Li J, Nie L, Xu X (2019) Activity recognition method for home-based elderly care service based on random forest and activity similarity. IEEE Access 7:16217–16225. https://doi.org/10.1109/ACCESS.2019.2894184
4. Gayathri KS, Elias S, Ravindran B (2015) Hierarchical activity recognition for dementia care using Markov Logic Network. Pers Ubiquitous Comput 19:271–285. https://doi.org/10.1007/s00779-014-0827-7
5. Direkoglu C, O'Conner NE (2012) Team activity recognition in sports. In: ECCV 2012. Lecture notes in computer science, vol 7578, pp 69–83
6. Vishwakarma S, Agrawal A (2013) A survey on activity recognition and behavior understanding in video surveillance. Visual Comput 29:983–1009. https://doi.org/10.1007/s00371-012-0752-6
7. Turaga P, Chellappa R, Subrahmanian VS, Udrea O (2008) Machine recognition of human activities: A survey. IEEE Trans Circuits Syst Video Technol 18:1473–1488. https://doi.org/10.1109/TCSVT.2008.2005594
8. Uddin S, Haque I, Lu H, Moni MA, Gide E (2022) Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. Sci Rep 12:1–11. https://doi.org/10.1038/s41598-022-10358-x
9. Patil SS, Dinesha HA (2022) URL redirection attack mitigation in social communication platform using data imbalance aware machine learning algorithm. Indian J Sci Technol 15:481–488. https://doi.org/10.17485/IJST/v15i11.1813
10. Li M, Fu X, Li D (2020) Diabetes prediction based on XGBoost algorithm. In: IOP conference series: materials science and engineering. Institute of Physics Publishing. https://doi.org/10.1088/1757-899X/768/7/072093
11. Sri Chandrahas N, Choudhary BS, Vishnu Teja M, Venkataramayya MS, Krishna Prasad NSR (2022) XG boost algorithm to simultaneous prediction of rock fragmentation and induced ground vibration using unique blast data. Appl Sci (Switzerland) 12. https://doi.org/10.3390/app12105269

12. Deng F, Jovanov E, Song H, Shi W, Zhang Y, Xu W (2023) WiLDAR: WiFi signal-based lightweight deep learning model for human activity recognition. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2023.3294004
13. Altun K, Barshan B (2010) Human activity recognition using inertial/magnetic sensor units. In: HBU 2010. Lecture notes in computer science. Springer, Berlin, vol 6219, pp 38–51. https://doi.org/10.1007/978-3-642-14715-9_5
14. Altun K, Barshan B, Tunçel O (2010) Comparative study on classifying human activities with miniature inertial and magnetic sensors. Pattern Recognit 43:3605–3620. https://doi.org/10.1016/j.patcog.2010.04.019
15. Barshan B, Yüksek MC (2013) Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. Comput J 57:1649–1667. https://doi.org/10.1093/comjnl/bxt075
16. Chowdhary CL, Acharjya DP (2020) Segmentation and feature extraction in medical imaging: a systematic review. Procedia Comput Sci 167:26–36. https://doi.org/10.1016/j.procs.2020.03.179
17. Hakak S, Alazab M, Khan S, Gadekallu TR, Maddikunta PKR, Khan WZ (2021) An ensemble machine learning approach through effective feature extraction to classify fake news. Futur Gener Comput Syst 117:47–58. https://doi.org/10.1016/j.future.2020.11.022
18. Zhang S, Cheng D, Deng Z, Zong M, Deng X (2018) A novel kNN algorithm with data-driven k parameter computation. Pattern Recognit Lett 109:44–54. https://doi.org/10.1016/j.patrec.2017.09.036
19. Bansal M, Goyal A, Choudhary A (2022) A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. Decis Anal J 3. https://doi.org/10.1016/j.dajour.2022.100071
20. Kurani A, Doshi P, Vakharia A, Shah M (2023) A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. Annal Data Sci 10:183–208. https://doi.org/10.1007/s40745-021-00344-x
21. Armaghani DJ, Asteris PG, Askarian B, Hasanipanah M, Tarinejad R, Huynh VV (2020) Examining hybrid and single SVM models with different kernels to predict rock brittleness. Sustainability (Switzerland) 12:1–17. https://doi.org/10.3390/su12062229
22. Qiu Y, Zhou J, Khandelwal M, Yang H, Yang P, Li C (2022) Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration. Eng Comput 38:4145–4162. https://doi.org/10.1007/s00366-021-01393-9

A K-Means Variation Based on Careful Seeding and Constrained Silhouette Coefficients

Libero Nigro, Franco Cicirelli, and Francesco Pupo

Abstract K-Means is well-known clustering algorithm very often used for its simplicity and efficiency. Its properties have been thoroughly investigated. It is emerged that K-Means heavily depends on the seeding method used to initialize the cluster centroids and that, besides the seeding procedure, it mainly acts as a *local refiner* of the centroids and can easily become stuck around a local sub-optimal solution of the objective function cost. As a consequence, K-Means is often repeated many times, always starting with a different centroids' configuration, to increase the likelihood of finding a clustering solution near the optimal one. In this paper, the Hartigan and Wong variation of K-Means (HWKM) is chosen because of its increased probability to ending up near the optimal solution. HWKM is then enhanced with the use of careful seeding methods and by an incremental technique which constrains the movement of points among clusters according to their Silhouette coefficients. The result is HWKM+ which, through a small number of restarts, is capable of generating a careful clustering solution with compact and well-separated clusters. The current implementation of HWKM+ rests on Java parallel streams. The paper describes the design and development of HWKM+ and demonstrates its abilities through a series of benchmark and real-world datasets.

Keywords Clustering · Hartigan and Wong K-Means · Careful seeding · Silhouette coefficients · Compact and well-separated clusters · Java parallel streams

L. Nigro (✉) · F. Pupo
University of Calabria, DIMES, 87036 Rende, Italy
e-mail: l.nigro@unical.it

F. Pupo
e-mail: f.pupo@unical.it

F. Cicirelli
CNR—National Research Council of Italy, Institute for High Performance Computing and Networking (ICAR), 87036 Rende, Italy
e-mail: f.cicirelli@icar.cnr.it

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_17

1 Introduction

Clustering has been defined as the “art” of splitting the data of an application domain of machine learning, bioinformatics, Artificial Intelligence, and so forth, into groups called clusters, in such a way that data points within the same cluster are similar to each other, and data points in distinct clusters are dissimilar. It has been demonstrated that, in its combinatorial basis, the clustering problem is NP-hard [1]. Therefore, except for very small datasets which can be solved optimally, only approximate solutions can be generated by heuristic algorithms.

A well-known heuristic is K-Means [2–4], very often preferred to more sophisticated algorithms because of its simplicity and efficiency. K-Means properties have been extensively studied in the last years [5, 6], to the point that the use of the algorithm comes with an awareness of its benefits and limitations.

1.1 Definitions and Lloyd’s K-Means

We have a dataset $X = \{x_1, x_2, \ldots, x_N\}$ of N data points $x_i \in R^D$, that is each data point is a vector with D numerical attributes (coordinates). The data points have to be partitioned in K (assumed an input parameter) clusters $\{C_1, C_2, \ldots, C_K\}$, $2 \leq K \ll N$. Each cluster is represented by its center or centroid. So there are K centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$. Similarity among numerical data points is normally expressed by the Euclidean distance.

K-Means assigns data points to clusters according to minimal distance to centroids: $C_j = C_j \cup \{x_i\}$if $\mu_j = nc(x_i)$ where $nc(x_i)$ denotes the nearest centroid to x_i, that is:

$$\mu_j = nc(x_i) \text{ with } j = \text{argmin}_{1 \leq h \leq K} \text{d}(x_i, \mu_h). \tag{1}$$

The goal of K-Means clustering is the minimization of the objective function cost *Sum Squared Errors* (SSE):

$$\text{SSE} = \sum_{(i=1)}^{N} \sum_{(j=1, x_i \in C_j)}^{K} \text{d}(x_i, \mu_j)^2, \text{ with } \mu_j = nc(x_i) \tag{2}$$

The basic behavior of the classical Lloyd's K-Means is summarized in Algorithm 1.

Algorithm 1 Pseudo-code of Lloyd's K-Means

```
Input: the dataset $X$ and the number of clusters $K$
Output: Final centroids and corresponding partitions of data points, together with some clustering accuracy indexes including the $SSE$
1. Initialization. Initialize centroids by some seeding method (e.g., uniform random)
2. Partitioning. Assign data points of $X$ to clusters according to the nearest centroid
3. Updating. Redefine centroids as the mean point of each cluster: $\mu'_j = \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i$
4. Termination. Check convergence or the maximum number of iterations was executed. If not termination, repeat from 2
```

At the initialization step, different seeding methods can be used as described later in this paper. The new centroids emerging at step 3 become current at the next iteration. The termination step either checks centroids' stability (convergence), that is the fact the distance between current centroids and those of the previous iteration is below a numerical threshold (e.g., 10^{-8}) or a maximum number of iterations was reached.

As demonstrated in [5, 6], the quality of a clustering solution obtained by K-Means heavily depends on the seeding method used at step 1. However, an intrinsic limitation of K-Means concerns the centroids movement at step 3, which can be difficult or impossible when multiple centroids are wrongly associated to a real cluster area, whereas centroids are missing in some other data space area, and the two areas are far each other and separated by intermediate clusters. This behavior explains the basic attitude of K-Means to easily become blocked in a local minimum of the function cost. The situation can be improved by repeating a certain number of times K-Means, said Repeated K-Means or K-Means with Restarts. At each repetition, a different seeding of the initial centroids is used. The more are the repetitions, higher is the chance for the clustering solution to end up near the optimal solution.

1.2 Examples of Seeding Methods

Uniform. A basic initialization procedure is Uniform Random. Centroids are defined by randomly picking K distinct data points in the dataset. The method does not exclude multiple centroids to be chosen which are close to one another, or that outliers are selected as centroids. The seeding method is simple but necessarily requires to be used with Repeated K-Means.

Other stochastic seeding methods [7] used in this work depend on an incremental technique for defining centroids. At a given time, L centroids are defined: $1 \leq L \leq K$. As a rule, the first centroid is established by a uniform random choice in the dataset. The seeding method terminates when $L > K$.

Let $D(x_i)$ be the minimal distance of a point x_i to currently defined centroids.

Maximin. A new centroid is defined as a point $x^* \in X$, not already selected, which has maximal $D(x^*)$.

K-Means++. Selects the next centroid by a *random switch* of the data points of the dataset, after associating to each point the probability of being chosen as:

$$\pi(x_i) = \frac{D(x_i)^2}{\sum_{j=1}^{N} D(x_j)^2}. \tag{3}$$

Greedy K-Means++. At each subsequent centroid selection, this method executes S times the K-Means++ procedure and chooses the next centroid as the data point, among the S candidates, which, combined with the existing centroids, mostly reduces the objective cost (the greedy step). As in [8, 9], the value $S = \lfloor 2 + \log K \rfloor$ is chosen in the experiments which is a trade-off between the improved seeding and the extra computational cost. Greedy K-Means++ proves effective in the practical case to provide careful seeding. It is capable of diminishing the number of restarts in Repeated K-Means [8–11] for finding a solution near to the optimal one.

1.3 Internal Versus External Clustering Indexes

In this work, besides the SSE, possibly used in a normalized way: $nMSE = SSE/(N * D)$ where N is the numerosity of the dataset X and D is the number of coordinates per data point, another used internal index is the Silhouette Index [12, 13].

Silhouette Index (SI). Given a clustering partitioning, the SI_x coefficient of a data point x is defined as:

$$SI_x = \frac{b_x - a_x}{\max(b_x, a_x)}, \tag{4}$$

where a_x is the internal *cohesion factor*, that is the average sum of the distances of x to all the remaining points of the same cluster:

$$a_x = \frac{1}{|C_j| - 1} \sum_{y \in C_j, y \neq x} \mathrm{d}(x, y). \tag{5}$$

If x is the sole point of its cluster, a_x is defined to be 0.

The b_x component measures the *separation factor* of x. In particular b_x is the minimal average sum of the distances of x (belonging to cluster C_j) to the points belonging to other clusters:

$$b_x = \text{argmin}_{1 \le h \le K, h \ne j} \frac{1}{|C_h|} \sum_{y \in C_h} \mathrm{d}(x, y). \tag{6}$$

Finally, the SI of the clustering solution is defined as the average of the Silhouette coefficients of the various points:

$$SI = \frac{1}{N} \sum_{x \in X} SI_x. \tag{7}$$

The SI value ranges between -1 and 1. An $SI = 1$ denotes well-separated clusters (minimal overlapping). An $SI = 0$ indicates the maximum of the overlapping among clusters. An SI value which tends to -1 simply expresses an incorrect clustering.

A clustering with minimal SSE (Eq. 1) (or nMSE) and maximal SI (Eq. 7) is characterized by compact and well-separated clusters [13]. Of course, it is not possible to optimize the two internal indexes at the same time.

Although its usefulness, the use of SI is challenging, particularly for large datasets. In fact, it costs $O(N^2)$ for the need to calculate all the pairwise distances of the data points. To reduce the computation overhead, a parallel implementation framework can be used. In addition, in this work, an incremental strategy is used which keeps updated the Silhouette coefficients of data points as they move from one cluster to another.

Centroid Index (CI). It was proposed in [14, 15] as a formal measure of the similarity/dissimilarity between two clustering solutions. The CI is particularly useful when comparing a clustering solution generated by a given algorithm, with a reference solution (*ground truth*) available for a benchmark or synthetic dataset. Ground truth information can be of two types: *ground truth centroids* (GTC) or *ground truth partitions* (GTP) (or labels). In the first case, the designer of a benchmark dataset furnishes the optimal centroid points around which the remaining points of the dataset are located according to a specific distribution (e.g., Gaussian). In the second case, the ideal ground truth partitions (clusters) are provided by indicating, for each data point, the label (centroid index) of the belonging cluster.

When using the GTC, the $CI(C_1, C_2)$ where C_1 and C_2 are centroid vectors, one being the GTC and the other is that generated by the used algorithm, can be calculated as follows. First each centroid C1[i] is mapped on the centroid C2[k] having minimal distance from C1[i]. After that, the number of *orphans* in C_2 is counted, which are the elements of C_2 upon which no element of C_1 is mapped on. In a similar way, the elements of C_2 are subsequently mapped onto C_1 and the resultant number of orphans in C_1 is counted. Then

$$CI(C_1, C_2) = \max(\text{orphans}(C_1 \to C_2), \text{orphans}(C_2 \to C_1)). \tag{8}$$

When the GTP are used, the elements to map are partitions that are set of points (clusters). The CI can be easily defined by using, for example, the classical Jaccard distance between sets (see, e.g., [16]).

A $CI = 0$ is a precondition for a correct solution structure. It should be noted that it is not required that the found centroids be coincident with ground truth centroids, but only that the above-mentioned bijection holds. A $CI > 0$ indicates how many centroids were incorrectly determined by the chosen algorithm.

1.4 Paper Contribution

The original contribution of this paper consists in the design and efficient implementation in Java of an extension of the Hartigan and Wong [17, 18] variation of K-Means (HWKM), named HWKM+. HWKM+ main features are the following:

- It can work with careful seeding methods.
- It inherits from HWKM the ability to improve classical Lloyd's K-Means clustering, by its powerful management of centroids which avoids in many practical cases to end up in a local sub-optimal solution.
- It owns a novel technique based on the incremental computation of individual Silhouette coefficients [13], which contributes to the definition of well-separated clusters.

Although HWKM+ inherits from its HWKM originator an intrinsic sequential behavior, many supporting internal operations, including the calculation of accuracy indexes like the Silhouette Index SI (see Eq. 7), can be carried out in parallel.

The paper demonstrates the performance of HWKM+ through a series of simulation experiments carried out on both benchmark and real-world datasets.

The rest of the paper is organized as follows. Section 2 briefly reviews the basic features of Hartigan and Wong's K-Means algorithm, upon which the HWKM+ tool proposed in this paper is based. Section 3 describes the design rationale of HWKM+ and outlines its Java implementation. Section 4 illustrates the experimental framework adopted for assessing the properties of HWKM+ by simulation experiments. Section 5 reports the achieved experimental results. Section 6, finally, concludes the paper by highlighting ongoing and future work.

2 Hartigan and Wong K-Means

The "modus operandi" of HWKM is summarized in Algorithm 2. It starts from an initialization of centroids by a given seeding method, and then the corresponding partitioning is realized, that is, points are assigned to clusters according to the nearest centroid rule.

The main course of the algorithm, which sharply distinguishes its behavior from that of K-Means, is captured by the for-loop at step 4 in Algorithm 2. Here, every individual point x_i of the dataset, firstly, is removed from its source cluster, with the centroid of the source cluster which is then updated on the basis of the remaining points in the cluster. Due to the new centroid configuration, the point x_i can possibly be assigned to a different cluster, indicated as the destination cluster, whose centroid gets updated accordingly. Of course, there are cases when the data point x_i effectively moves from a cluster to another distinct one and others in which x_i is re-assigned to its source cluster.

At the loop end, if at least one point changed its cluster, the algorithm is repeated from step 4.

The iterations of the algorithm terminate when centroids and clusters stabilize.

Algorithm 2 Operation of Hartigan and Wong's K-Means

```
Input: the dataset $X$ and the number $K$ of clusters
Output: the final $K$ centroids and associated clusters/partitions, together with some clustering accuracy indexes, including the SSE/nMSE
1. Define the initial $K$ centroids by a seeding method
2. Partition the dataset $X$ according to initial centroids and the $nc(.)$ rule
3. Set $s = true$
4. For each data point $x_i \in X$ do
   (a) remove $x_i$ from its source cluster $C_{src}$
   (b) update the centroid $\mu_{src}$ of modified $C_{src}$
   (c) assign $x_i$ to destination cluster $C_{dst}$ where $\mu_{dst} = nc(x_i)$
   (d) update the centroid $\mu_{dst}$ of modified cluster $C_{dst}$
   (e) if $src \neq dst$, set $s = false$
5. if $s = false$, set $s = true$ and go to step 4
```

It is worthy of note that the use of a stochastic seeding method (see Sect. 1.2) can require HWKM to be also restarted, as for the basic K-Means, a certain number of times. However, the number of repetitions (independent runs) required to approach an accurate clustering solution is normally smaller than those needed by K-Means. Moreover, the number of repetitions can also reduce in the case a careful seeding (for instance, Greedy K-Means++) is adopted.

3 Design Rationale of HWKM+ in Java

Actions from 4(a) to 4(d) in Algorithm 2 refer to processing a single data point of the dataset, and data points must be managed necessarily one at a time. A first incremental strategy in the Java implementation of HWKM+ was aimed, by suitable data structures, to accelerate the update operations during the movement of a data point from one cluster (src) to another (dst). Data points are instances of a DataPoint class which exposes methods for computing the Euclidean distance between two points, for adding/subtracting arithmetically points (while keeping the count of the number of additions/subtractions), for calculating the mean point of a sum of data points and so forth. A particular field of DataPoint is CID (CLuster iDentifier) which stores the index of the nearest centroid, that is the identity of the belonging cluster (from 0 to $K - 1$).

3.1 Accelerating the Operations of Data Points for Switching Between Clusters

Following the initial partitioning (step 2 in Algorithm 2) which assigns to each point its starting label (CID), the points belonging to each cluster/partition are collected in a separate list (an element of the $cluster[]$ array of K lists) and the total sum of the points of a cluster is held in a particular data point (an element of a $centre[]$ array of K DataPoint). As a consequence, removing a point x_i from its source cluster (CID=src) reduces to very few operations: (i) removing the point identifier i from the list $cluster[src]$; (ii) subtracting the point from the sum held in $centre[src]$; (iii) updating the centroid of cluster src as the mean point through the sum held in $centre[src]$. Dual operations are carried out when a point x_i is to be added to a cluster dst (see step 4(c) in Algorithm 2); (iv) the point is added to the list $cluster[dst]$; (v) the point is added to the sum held in $centre[dst]$; (vi) the new centroid of dst is calculated as the mean point through the sum held in $centre[dst]$.

The described operations significantly reduce the time required by each iteration of the for-loop (step 4 in Algorithm 2). Algorithm 3 shows a Java stream-based implementation [4, 16, 19] of the partitioning at step 2 of Algorithm 2.

Algorithm 3 Stream-based implementation of initial partitioning

```
Stream < DataPoint > p_stream = Stream.of( dataset);
if( PARALLEL) p_stream = p_stream.parallel();
p_stream
  .map(p - > {
        double md = Double.MAX_VALUE;
        for(int k = 0; k < K;++k) {
          double d = p.distance( centroids[k]);
          if(d < md) { md = d; p.setCID(k);}
        }
        cluster[p.getCID()].add(p); //add p to its cluster
        return p;
    })
    .for Each (p- > { });
```

A subtle point in Algorithm 3 concerns the fact that the dataset points can be processed in parallel and that each point modifies itself (its CID field) except when the $cluster[CID]$ list is updated by adding to it the point p. To avoid data inconsistency, the $cluster[]$ lists are implemented as $ConcurrentLinkedQueue$s. This way, a same linked list can be safely accessed simultaneously by multiple threads.

3.2 Constraining Data Point Movement to Silhouette Coefficients

Another incremental technique was added to HWKM+ in order to constrain the actual movement of a data point from one cluster to another, to occur only when the movement does not create a penalty in the Silhouette coefficient (SC) of the point. The idea is always trying to improve the Silhouette index during the operation of HWKM+ so as to possibly generate a solution of well-separated clusters. More in particular, the transfer of point x_i from cluster src to cluster dst can commit only when the $SC_{x_i}^{dst} \geq SC_{x_i}^{src}$, that is the SC value of x_i in dst, should not be smaller than the SC value x_i had in src. Otherwise, the movement does not take place.

To support the SC-based incremental technique, each data point p maintains in itself two additional fields: S which always holds the sum of the distances from p to all the remaining points of the same cluster ($p.CID$); $B[]$ which is an array of K elements, where $B[h]$, $h \neq p.CID$, holds the sum of the distances from p to all the points in the external cluster h. Of course, the values of S and $B[.]$ permit to calculate the SC of point p. Such data are initialized after the initial partitioning and are kept updated during the movement (switch) of a point from one cluster to another. Three methods were developed: $switchOk(\ldots)$ which returns true if a switch from src to dst can occur; $undoSwitch(\ldots)$ to undo changes in the S and $B[]$ variables of an attempted but not acceptable switch; $finalizeSwitch(\ldots)$ which commits a switch.

From the individual values of S and $B[]$ fields continuously maintained updated in the data points, it is possible to calculate (in parallel) the overall Silhouette Index (SI) as shown in Algorithm 4. First the SC of each point is computed and held in the SC field. Then, the SC of all the dataset points is added by the $reduce()$ operation and is returned as part of a data point sc, whose SC value is finally divided by N

Algorithm 4 Calculating the SI index from the individual Silhouette coefficients

```
Stream <DataPoint> pStream = Stream.of(dataset);
if( PARALLEL) pStream = pStream.parallel();
DataPoint si = pStream
  .map( p- > {
      p.setSC( p.computeSC());
      return p;
  })
  .reduce( new DataPoint(), (p1,p2)- > {
      DataPoint p = new DataPoint();
      p.setSC( p1.getSC() + p2.getSC());
      return p;
  });
return si.getSC()/N;
```

3.3 HWKM+ Parameters

The setting of a few parameters allows the Repeated HWKM+ version to search for a “best” clustering solution by optimizing (minimizing) the $nMSE$ (default) or (maximizing) the Silhouette Index SI. Obviously, even when the $nMSE$ is chosen as the optimization function cost, the constraining strategy based on the Silhouette coefficients applies behind the scene to favor the achievement of well-separated clusters. Other parameters permit to specify the maximum number of repetitions and the particular seeding method to use.

4 Experimental Setup

HWKM+ was tested by using both synthetic and real-world datasets. Table 1 collects the parameters N (dataset dimension), D (number of coordinates), K (number of centroids), and the kind of ground truth information available: GTC (ground truth centroids) or GTP (ground truth partitions), for the synthetic datasets used for the simulation experiments [20]. Some benchmark datasets are very challenging to be clustered under the optimization of $nMSE$. For the Aggregation dataset, ground truth centroids were preliminarily extrapolated from the GTP information, by finding the *medoid* (the data point in a cluster which has a minimal sum of the distances from the remaining points in the cluster) corresponding to the assigned partitions.

Some non-synthetic datasets, which are without ground truth information, selected for the experiments are reported in Table 2.

Table 1 Parameters of selected synthetic datasets

Dataset	N	D	K	GTC/GTP	Source
Sd1	1450	2	4	GTP	[13]
Aggregation	788	2	7	GTC	[20]
Spiral	312	2	3	GTP	[20]
Path based	300	2	3	GTP	[20]
Jain	273	2	2	GTP	[20]

Table 2 Parameters of selected real-world datasets

Dataset	N	D	K	Source
Miss America	6480	16	256	[20]
Iris	150	4	3	[20]
Olivetti	400	4096	40	[8, 9]
House	34,112	3	256	[20]
Bridge	4096	16	256	[20]

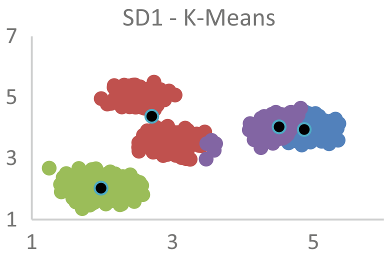
5 Experimental Results

The goal of the execution runs was to check the effectiveness and the accuracy of HWKM+ driven by careful seeding together with the constrained strategy on the Silhouette coefficients. It is worth noting, though, that in some cases, HWKM+ was able to find a solution near the optimal one by only relying on the Greedy K-Means++ seeding procedure.

All the experiments were carried out on a Win10 Pro, Dell XPS 8940, Intel i7-10700 (8 physical + 8 virtual cores), CPU@2.90 GHz, 32GB Ram, and Java 17.

Figures 1 and 2 show the simple dataset SD1 ([13], page 6) clustered by using K-Means and HWKM+ with uniform random seeding. Only one execution was used. As one can see from Fig. 2, HWKM+ was able (exactly as in [13]) to correctly solve SD1.

Fig. 1 SD1 by K-means with random



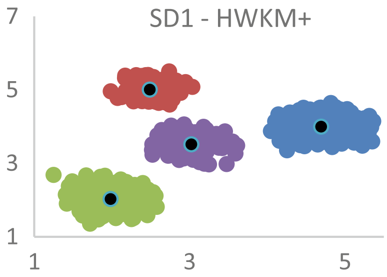


Fig. 2 SD1 by HWKM+ with random

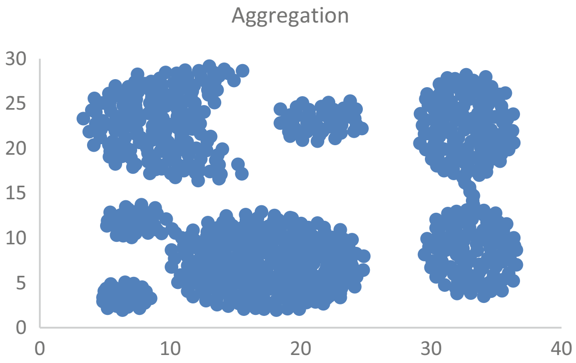


Fig. 3 Challenging Aggregation dataset

Figure 3 shows the challenging Aggregation dataset, which is representative of a class of data spaces which cannot be correctly clustered by minimizing the $nMSE$ cost.

Repeated K-Means (RKM), with 1000 repetitions and G-K-Means++ (GKM++) seeding, terminates with a $CI = 1$, that is: one centroid is incorrectly predicted. By using HWKM+ with G-K-Means++, 20 repetitions and asking to optimize the Silhouette Index SI, the approximate solution shown in Fig. 4 is generated, which has a $CI = 0$. The ideal solution which computes partitions exactly as the macro areas in Fig. 3 can be generated by Density Peaks-based clustering algorithms [21] which are not guided by the minimization of $nMSE$.

To the same class of Aggregation belong the other three datasets Spiral, Path based, and Jain of Table 1, whose generated solutions by HWKM+, all having a $CI = 0$, are presented in the figures from Figs. 5, 6 and 7. Black dots are the centroids proposed by HWKM+. The best solution (minimal $nMSE$) emerged with RKM with GKM++ and 1000 repetitions suggests a $CI = 1$ for Spiral and $CI = 0$ for Pathbased and Jain.

For the non-synthetic datasets, the observed $nMSE$ and SI of the best solution detected after 100 repetitions of HWKM+ with G-K-Means++ seeding are collected in Table 3, where also the PET, that is the Parallel Elapsed Time, in sec, and the average number of iterations per run (aIT) are reported. The clustering results are in

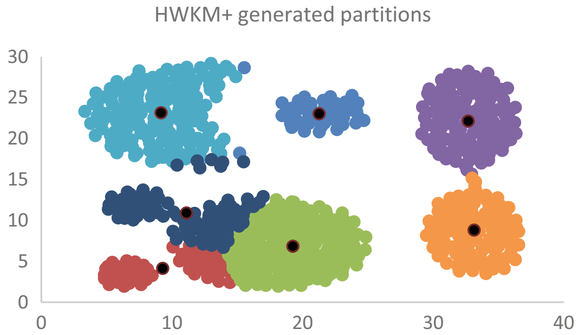


Fig. 4 Proposed solution for Aggregation; black dots are the emerged centroids

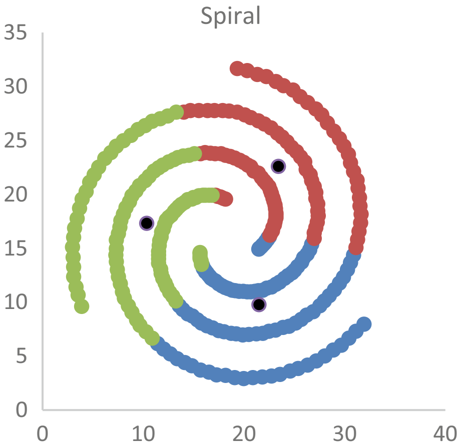


Fig. 5 Spiral clustering by HWKM+

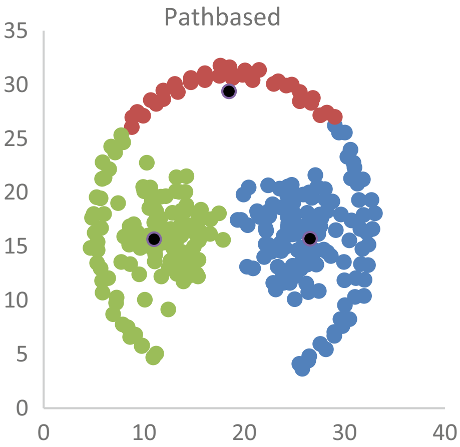
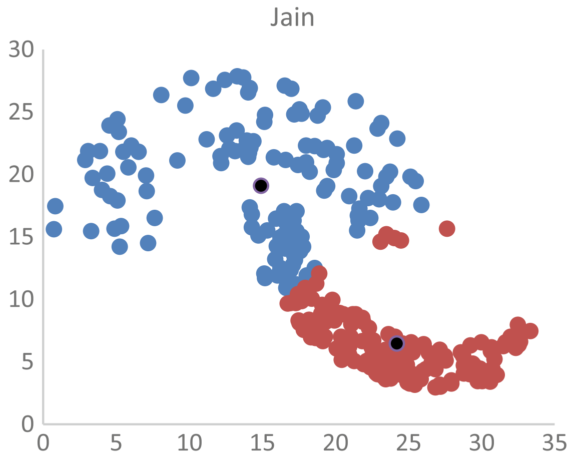


Fig. 6 Pathbased clustering by HWKM+

Fig. 7 Jain clustering by HWKM+



good agreement with similar results achieved with the evolutionary Recombinator K-Means in [8, 9].

For the Iris and Olivetti datasets, some ground truth information was available for assessing the accuracy of the clustering. In particular, for the Iris, at the minimum of $nMSE$, a $CI = 0$ has emerged. For the Olivetti dataset, which is concerned with the recognition of the facial images of 40 subjects reproduced in ten different poses (4096 are the pixels of each photo), a $CI = 7$ was observed in the best case, that is seven subjects are not recognized.

Finally, Table 4 reports the elapsed time (ET) and the total number of iterations (tIT) executed in each of five runs of HWKM+ with G-K-Means++ seeding, applied to the Olivetti dataset of Table 2, respectively, in sequential and parallel modes. From the total elapsed time and the total number of iterations, the average elapsed time per iteration in sequential (iET^{S}) and parallel (iET^{P}) mode can be derived. Then, the average iET on the five runs, in sequential and parallel, can be estimated as: aiET^{S} = 2.44 s, aiET^{P} = 0.59 s, with a resulting speedup of:

$$\text{speedup} = aiET^{S}/aiET^{P} = 2.44/0.59 = 4.14.$$

Table 3 HWKM+ results on the real-world datasets of Table 2, 100 repetitions and G-K-Means++ seeding

Dataset	nMSE	SI	PET(sec)	aIT
Miss America	5.39	0.049	551	18
Iris	2.33	0.46	0.5	4
Olivetti	0.0072	0.16	164	6
House	9.39	0.23	1916	33
Bridge	171.88	0.09	346	13

Table 4 Execution times of five runs of the Olivetti dataset (eight physical cores)

Run	SET(s)	tIT	iET^S (s)	PET(s)	tIT	iET^P (s)
1	154	63	2.44	34	53	0.64
2	154	56	2.75	35	61	0.57
3	155	68	2.28	35	61	0.57
4	154	59	2.61	35	62	0.56
5	155	74	2.09	34	58	0.59

The parallel efficiency is: $\eta = \frac{\text{speedup}}{\#\text{cores}} = \frac{4.14}{8} = 0.52$.

The limited speedup closely mirrors the limited parallelism degree existing in the HWKM+ behavior. Parallelism is mainly exploited within internal operations executed during the iterations of HWKM+, for example in the $finalizeSwitch(\ldots)$ method which is invoked each time a point movement from a cluster to another is finally committed because it preserves a non-decreasing Silhouette coefficient, and in the computation of the function cost $nMSE$ and the overall Silhouette Index SI.

6 Conclusions

This paper proposes a new clustering algorithm HWKM+ which leverages the Hartigan and Wong variation of K-Means [17, 18]. HWKM+ improves the operations for moving data points between clusters and integrates an original incremental technique which constrains the actual movement of a data point to the increment of its individual Silhouette coefficient [13]. All of these, paired with the use of careful seeding and the minimization of the Sum-of-Squared-Error (SSE) function cost, make it possible for HWKM+, with few repetitions, to favor the obtainment of a solution with compact and well-separated clusters.

The paper discusses the design and implementation in Java of HWKM+, which purposely depends on parallel streams [4, 16, 21], and presents some preliminary experimental results which confirm accurate clustering solutions can be achieved.

The prosecution of the research will be geared to the following points: first to extend the experimental framework with more challenging datasets. Second to port HWKM+ in the context of an evolutionary algorithm [11]. The goal is to exploit the constraining technique based on the Silhouette coefficients for better preparing the population of candidate centroids, which subsequently are recombined toward the achievement of a high-quality solution. Third to complete an actor-based implementation of HWKM+ on top of the Theater system [22].

References

1. Garey MR, Johnson DS, Witsenhausen HS (1982) The complexity of the generalized Lloyd-Max problem. IEEE Trans Inf Theory 28:255–256
2. Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recogn Lett 31(8):651–666
3. Lloyd SP (1982) Least squares quantization in PCM. IEEE Trans Inform Theory 28(2):129–137
4. Nigro L (2022) Performance of parallel K-means algorithms in Java. Algorithms 15(4):117
5. Fränti P, Sieranoja S (2018) K-means properties on six clustering benchmark datasets. Appl Intell 48(12):4743–4759
6. Fränti P, Sieranoja S (2019) How much can k-means be improved by using better initialization and repeats? Pattern Recogn 93:95–112
7. Vouros A, Langdell S, Croucher M, Vasilaki E (2021) An empirical comparison between stochastic and deterministic centroid initialization for K-means variations. Mach Learn 110:1975–2003
8. Baldassi C (2020) Recombinator K-means: a population-based algorithm that exploits k-means++ for recombination. arXiv:1905.00531v3, Artificial Intelligence Lab, Institute for Data Science and Analytics, Bocconi University, via Sarfatti 25, 20135 Milan, Italy
9. Baldassi C (2022) Recombinator K-Means: An evolutionary algorithm that exploits k-means++ for recombination. IEEE Trans Evol Comput 26(5):991–1003. https://doi.org/10.1109/TEVC.2022.3144134
10. Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 40(1):200–210. https://doi.org/10.1016/j.eswa.2012.07.021
11. Nigro L, Cicirelli F (2023) Performance of a K-means algorithm driven by careful seeding. In: Proceedings of the 13th international conference on simulation and modeling methodologies, technologies and applications, pp 27–36. ISBN 978-989-758-668-2, ISSN 2184-2841
12. Rousseeuw P (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
13. Bagirov AM, Aliguliyev RM, Sultanova N (2023) Finding compact and well-separated clusters: clustering using silhouette coefficients. Pattern Recogn 135:109144. https://doi.org/10.1016/j.patcog.2022.109144
14. Fränti P, Rezaei M, Zhao Q (2014) Centroid index: cluster level similarity measure. Pattern Recogn 47(9):3034–3045
15. Fränti P, Rezaei M (2016) Generalizing centroid index to different clustering models. In: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), pp 285–296, Springer, Berlin
16. Nigro L, Cicirelli F, Fränti P (2023) Parallel random swap: an efficient and reliable clustering algorithm in Java. Simul Model Pract Theory 124:102712
17. Hartigan JA, Wong MA (1979) Algorithm as 136: A k-means clustering algorithm. J Roy Stat Soc: Ser C (Appl Stat) 28(1):100–108
18. Slonim N, Aharoni E, Crammer K (2013) Hartigan's k-means versus Lloyd's k-means-is it time for a change? IJCAI 1677–1684
19. Urma RG, Fusco M, Mycroft A (2019) Modern Java in action. Manning, Shelter Island
20. Benchmark Datasets (2023). http://cs.uef.fi/sipu/datasets/. Last accessed on July 2023
21. Rodriguez R, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):14.92–14.96
22. Nigro L (2021) Parallel Theatre: a Java actor-framework for high-performance computing. Simul Model Pract Theory 106:102189

Satellite Image Analysis in Health Care—A Systematic Review

Bhushan Pawar, Vijay Prakash, Lalit Garg, Charles Galdies, Sandra Buttigieg, and Neville Calleja

Abstract Rapid urbanization, population expansion, and escalating pollution levels have given rise to novel environmental concerns that demand the application of creative, analytical methodologies and diverse data sources. To ensure efficient urban ecological management, it is crucial to consider three levels of phenomena: the environmental system, the physical environment, and the regional surroundings. Utilizing remote sensing data is a viable avenue for enhancing the global environment due to its convenient accessibility and ability to measure crucial physical features regularly. The primary objective of this research study is to ascertain and evaluate suitable methods for analyzing satellite imagery in the healthcare context. The study's aims encompass examining the diverse array of tools accessible in the market, scrutinizing the characteristics of each device, exploring research papers that have employed

B. Pawar · V. Prakash (✉) · L. Garg
Department of Computer Information Systems, Faculty of Information and Communication Technology, University of Malta, Msida 2080, MSD, Malta
e-mail: vijaysoni200@gmail.com

B. Pawar
e-mail: bhushan.d.pawar.21@um.edu.mt

L. Garg
e-mail: lalit.garg@um.edu.mt

C. Galdies
Environmental Management and Plan Division, Institute of Earth Systems, University of Malta, Msida, Malta
e-mail: charles.galdies@um.edu.mt

S. Buttigieg
Faculty of Health Sciences, University of Malta, Msida, Malta
e-mail: sandra.buttigieg@um.edu.mt

N. Calleja
Faculty of Medicine and Surgery, University of Malta, Msida, Malta
e-mail: neville.calleja@um.edu.mt

V. Prakash
Department of Computer Science, Graphics Era Hill University, Dehradun, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_18

these tools, and classifying the tools into open-source and commercial classifications. This study emphasizes the parameters associated with feature extraction and picture enhancement in satellite imagery, as these factors hold significant importance in the analysis of images.

Keywords Healthcare application · Satellite image · Remote sensing · Environmental factors

1 Introduction

Satellite data and its fusion with diverse approaches have become a potent resource in healthcare research. It provides vital insights into the intricate correlation between environmental aspects and health consequences [1]. Recently, many studies have used satellite imagery and advanced analytical methods to examine many health-related topics, including disease modeling, risk assessment, and environmental health Comprehending environmental factors' impact on health necessitates utilizing satellite imagery and remote sensing. The examination of climatic conditions and health consequences in Burkina Faso involved the utilization of a digital elevation model (DEM) and satellite data from SPOT-5, the Radar Topography Mission (version 4.1), TRMM, and MODIS [2]. The data comprehensively analyzed the complex correlation between environmental factors and health in the Kossi district of Burkina Faso.

Using satellite data and spatial–temporal features has enhanced disease prediction and understanding. A study in Sri Lanka [3] used a complex prediction model incorporating spatial and temporal data by applying AVNIR-2 and other Earth-observation satellite data. Researchers analyzed spatial and temporal features to predict disease patterns and identify hotspots, providing valuable information for public health initiatives. Hyperspectral imaging, a satellite-based technology, has shown promise in several healthcare applications. Analyzed in situ spectra data from the HJ1 environmental satellite and hyperspectral imager (HSI) were used to study China's Jiangsu and Zhejiang provinces [4]. This research study demonstrated the capability of hyperspectral images to extract relevant health-related data from the electromagnetic spectrum, highlighting the significance of satellite-based hyperspectral imaging for healthcare research. The utilization of the Maxent Algorithm in healthcare research has yielded valuable insights into the distribution of diseases and the assessment of risks. As an illustration, researchers in the Aburrá Valley of Colombia [5] utilized Landsat imagery with a spatial resolution of 30 m. They worked alongside municipal health departments to evaluate the distribution and threat of a particular health-related event. The results emphasized the appropriateness of utilizing the Maxent Algorithm and satellite data to detect disease hotspots, hence assisting in developing focused interventions and prevention initiatives. The findings highlighted the suitability of the Maxent Algorithm and satellite data for identifying disease hotspots, thereby guiding the design of targeted interventions and prevention strategies.

Malaria, a significant public health concern, has also been the focus of satellite-based healthcare research. In Burkina Faso, [6] utilized satellite data from the SPOT-5 satellite and ArcMap, along with the Region of Interest tool provided by ENVI software, to analyze malaria hotspots. By integrating satellite imagery and geospatial analysis tools, the researchers gained insights into the spatial distribution of malaria. They identified risk factors critical to implementing effective prevention and control measures. Satellite data integration has furthered our understanding of the relationship between environmental factors and health outcomes in diverse settings. In Côte d'Ivoire, [7] employed binary non-spatial models and Bayesian geostatistical logistic regression models to investigate this relationship in four regions. Integrating satellite datasets, including MODIS, WorldClime, FEWS NET, and STRM-WBD, revealed essential associations between satellite imagery and health outcomes, contributing to our understanding of disease patterns and risk factors. Satellite-based healthcare research has extended beyond epidemiology to encompass other aspects of public health. In Swaziland, [8] employed logistic regression and decision tree methodologies using datasets collected from the SPOT-5 satellite to understand the relationship between satellite imagery and malaria prevalence.

Satellite data have facilitated studies on the influence of land cover and topography on disease transmission dynamics. In India [9], researchers utilized topography modeling by including satellite data from SRTM (GTOP30), MODIS, and a digital elevation model (DEM) to evaluate the correlation between these parameters and the occurrences of diseases. This study illuminates the importance of topographic and environmental elements in influencing the patterns of disease spread. Research undertaken in northeastern Rwanda [9], the USA [10, 11], French Guiana [12], and Brazil [13] has contributed significantly to our comprehension of satellite-based healthcare research. These studies have employed satellite data from different sources, such as Thematic Mapper (TM), Landsat's Multispectral Scanner (MSS), France's Système Pour l'Observation de la Terre (SPOT), and NOAA's Advanced Very High-Resolution Radiometer (AVHRR), to examine the impact of meteorological parameters on public health.

The articles examined in this review emphasize the varied approaches, datasets, and geographical areas that researchers have utilized to enhance our comprehension of health-related phenomena. Using satellite imagery and sophisticated analytical methodologies, these investigations have yielded significant revelations regarding disease patterns, risk evaluation, and the influence of environmental elements on human well-being. The integration of satellite data has the potential to bring about a significant transformation in public health initiatives by guiding targeted actions and ultimately enhancing health outcomes on a global level. The aims of this research study are:

- To present a thorough summary of research papers that have employed satellite data and diverse approaches for healthcare applications.
- To provide a concise overview of each study's main findings and contributions, emphasize the methodology and sources of satellite data utilized.

- To identify prevalent patterns and developments in the utilization of satellite imagery for healthcare purposes, including integrating various satellite data sources and applying modern analytical methodologies.
- To emphasize the capacity of satellite data to influence public health initiatives, reaffirm disease surveillance, and deepen our comprehension of the intricate relationship between environmental factors and health consequences.

By addressing these objectives, this review research paper aims to provide a comprehensive and descriptive summary of current knowledge regarding using satellite imagery in healthcare research, emphasizing its significance, potential, and future directions.

2 Related Work

This section's primary objective is to establish the significance of satellite analysis methods and delineate the extent of novel analysis methods in many research domains. The preferred approaches and methods are outlined in Table 1.

3 Methodology

The below-given PRISMA diagram (Fig. 1) presents a diagrammatical representation of the literature matrix included in this section.

We investigated two resources for obtaining previously published articles; 285 citations were obtained from the SciHub database, and 100 citations were obtained from IEEE Xplorer. We excluded 82 citations due to duplication after reviewing the abstract section and titles of sources. We used the keyword "Remote Sensing in HealthCare" as an exclusion criterion and deleted studies that were not influential from our literature review. As a result, we received 188 citations for further investigation. To find more specialized research studies, we used additional inclusion criteria, "GIS Tools in Healthcare," which resulted in 65 citations being eliminated after the full-text screening, 56 publications being excluded during data extraction, and 49 articles being removed due to the year criteria (older than 2000). Finally, we obtained 18 distinct papers that met both requirements; as a result, we presented a complete analysis of these research works in our literature matrix.

Table 1 Summary of existing studies in the literature

References	Method/approach	Satellite/weather stations	Study area
[14]	The logistic and auto-logistic regression model	China Meteorological Data Sharing Service System (Collected by 727 meteorological Stations)	Mainland China
[15]	Dengue predictive model	AVHRR and MODIS	Loreto, Peru
[16]	Bivariate and multivariate logistic regression modeling	Pan-sharpened multispectral quick bird data (Data Analysis: ArcGIS and ArcInfo Workstation)	Dongola and Merowe (Northern Sudan)
[17]	Stepwise regression model	NASA's 90 m resolution Shuttle Radar Topography Mission data	Afghanistan
[18]	The malaria risk model (generated by multiple logistic regression)	Landsat 5 TM (multispectral images)	The northwest of the State of Mato Grosso, Brazil
[2]	A digital elevation model (DEM)	SPOT-5, Radar Topography Mission, MODIS, and TRMM	The Kossi district, Burkina Faso
[3]	Integrated spatial–temporal prediction model	AVNIR-2 instrument and Earth-observation satellites (RESTEC, 2014)	The northern region of Sri Lanka
[4]	A derivative model based on the in situ spectra	HJ1 and HIS	Jiangsu and Zhejiang provinces, China
[5]	The Maxent Algorithm	Landsat 7 imagery (spatial resolution 30 m)	Aburrá Valley, Colombia
[6]	Region of interest-tool (ROI) provided by ENVI	SPOT-5	Northwestern Burkina Faso
[7]	Binary non-spatial models and Bayesian geostatistical logistic regression models	MODIS, WorldClime, FEWS NET, STRM-WBD, etc.	Four regions of western Côte d'Ivoire
[8]	Logistic regression and Decision Tree methodology	Geo-database maintained by (NMCP)	Swaziland

(continued)

Table 1 (continued)

References	Method/approach	Satellite/weather stations	Study area
[19]	An Alternative Method (topography modeling using a digital elevation model (DEM))	SRTM (GTOP30) MODIS	Bihar, Jharkhand, West Bengal, and Uttar Pradesh (India)
[10]	Remote sensing-based models	Landsat 7: MSS and TM, NOAA: AVHRR), SPOT	The northeastern USA
[11]	A range standardized and point-to-point similarity metric (the DOMAIN procedure)	NOAA	USA
[12]	A climate-based forecasting model	Institut National de la Statistiques et des Etudes Economiques (www.insee.fr)	French Guiana, France
[13]	Predictive model (using CART algorithm)	A Landsat 5 Thematic Mapper (TM)	Parnaíba e Poti, Brazil
[9]	SEIR transmission model	Global Urban Footprint, Worldpop Rwanda 2010 100 × 100 m resolution population data, Influenza Sentinel Surveillance (ISS) system data and Strengthening Influenza Sentinel Surveillance data	Kigali, Rwanda

4 Result Analysis

This section provides a comprehensive overview of research studies utilizing satellite data and various methodologies for healthcare applications. These studies encompass a wide range of topics and geographic locations, underscoring the significant potential of satellite imagery in advancing our understanding of health-related issues. The findings and contributions of each study offer valuable insights that can inform policymaking, guide public health interventions, and drive further research in this domain. One notable study, conducted at the Centre de Recherche en Santé de Nouna in Burkina Faso (Study [2]), employed a digital elevation model (DEM) along with satellite data from SPOT-5, Radar Topography Mission (SRTM version 4.1), MODIS, and TRMM. By leveraging these datasets, the researchers aimed to investigate the impact of topography and environmental factors on health outcomes in the Kossi district of northwestern Burkina Faso. Integrating satellite imagery and topographic data enabled a comprehensive analysis of the complex relationship between ecological conditions and health.

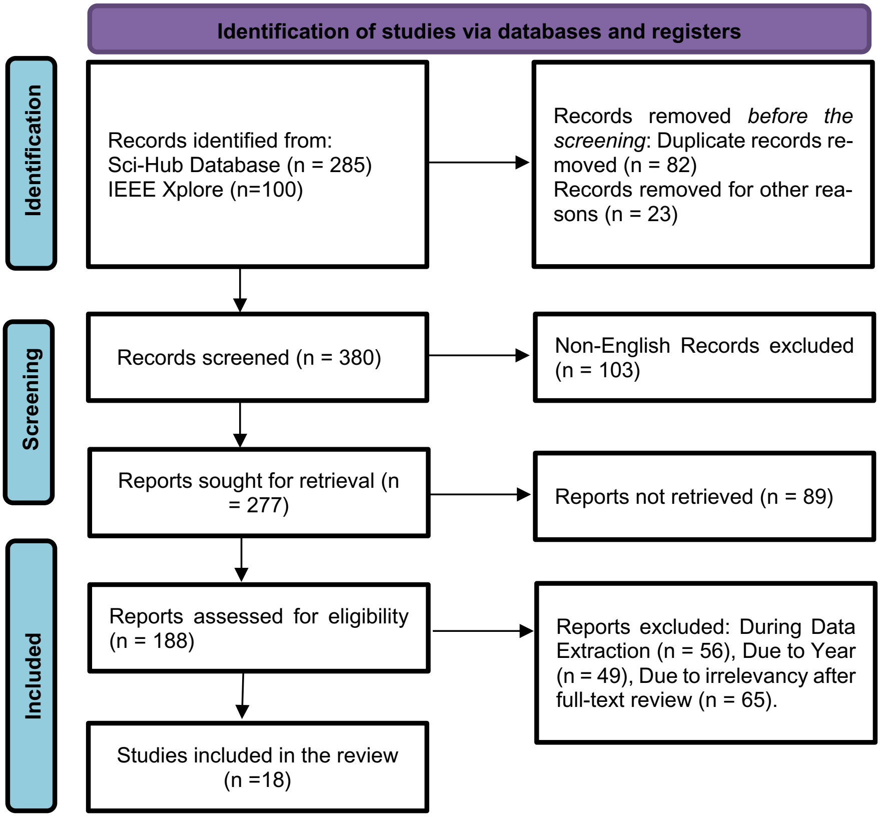

Identification of studies via databases and registers
Identification
Screening
Included
Records identified from:
Sci-Hub Database (n = 285)
IEEE Xplore (n=100)
Records removed *before the screening*: Duplicate records removed (n = 82)
Records removed for other reasons (n = 23)
Records screened (n = 380)
Non-English Records excluded (n = 103)
Reports sought for retrieval (n = 277)
Reports not retrieved (n = 89)
Reports assessed for eligibility (n = 188)
Reports excluded: During Data Extraction (n = 56), Due to Year (n = 49), Due to irrelevancy after full-text review (n = 65).
Studies included in the review (n =18)

Fig. 1 PRISMA diagram for the review process

Another study (Study [3]) developed an integrated spatial–temporal prediction model for the northern region of Sri Lanka, utilizing satellite data from the AVNIR-2 instrument and other Earth-observation satellites. The research, spanning from January 2010 to December 2013, aimed to predict and understand disease patterns in the region by incorporating spatial and temporal factors. The integration of satellite data allowed for a holistic analysis of the dynamic nature of disease transmission, facilitating the identification of potential hotspots and the implementation of targeted interventions. The study [4, 19] utilized in situ spectra data from the HJ1 environmental satellite and hyperspectral imager (HSI) in Jiangsu and Zhejiang provinces of China. By analyzing these spectral data between July and August of 2004 and 2005, the researchers developed a derivative model demonstrating hyperspectral imagery's potential in healthcare applications. This study sheds light on the utility of satellite-based hyperspectral imaging to extract valuable health-related information from the electromagnetic spectrum.

The Maxent Algorithm was also applied in Study [5] to assess the distribution and risk of a specific health-related phenomenon in the Aburrá Valley of Colombia. By leveraging Landsat imagery with a spatial resolution of 30 m, the researchers collaborated with municipal health departments to investigate the relationship between satellite imagery and disease prevalence. The findings provided valuable insights into the suitability of the Maxent Algorithm and satellite data for identifying and characterizing disease hotspots, aiding in designing targeted interventions.

Study [6] utilized the Region-of-Interest tool provided by ENVI software and satellite data from the SPOT-5 satellite and ArcMap to analyze the malaria situation in Northwestern Burkina Faso from August to October 2008. The study aimed to identify and characterize malaria hotspots, which are critical for implementing effective prevention and control measures. Integrating satellite imagery and geospatial analysis tools enabled a comprehensive assessment of malaria risk factors, enhancing the understanding of disease dynamics in the region. The study in Côte d'Ivoire [7] utilized binary non-spatial models and Bayesian geostatistical logistic regression models to examine the correlation between satellite imagery and health outcomes. The researchers investigated the correlation between satellite imaging and health outcomes in the four regions of western Côte d'Ivoire by analyzing data from multiple sources, such as MODIS, WorldClime, FEWS NET, and STRM-WBD. This study incorporates satellite datasets to comprehend disease patterns and uncover risk factors.

In addition, the Study [8] utilized logistic regression and decision tree techniques with datasets obtained from the SPOT-5 satellite. The researchers utilized the geodatabase of the National Malaria Control Program (NMCP) to examine Swaziland specifically and investigate the correlation between satellite images and the prevalence of malaria. The research findings enhanced comprehension of the regional dispersion of malaria and furnished useful insights for directing interventions and allocating resources. A study in India utilized a digital elevation model (DEM) to employ an alternate approach for topography modeling [19]. A study conducted between 2005 and 2007 in Bihar, Jharkhand, West Bengal, and Uttar Pradesh examined the correlation between the geographical features, vegetation, and the occurrence of diseases. The researchers evaluated the influence of topographic and environmental parameters on disease transmission patterns by analyzing satellite data from SRTM (GTOP30) and MODIS.

A study conducted in the northeastern USA [10] demonstrated the application of remote sensing-based models in the management of natural environments, utilizing satellite data from Landsat's Multispectral Scanner (MSS), NOAA's Advanced Very High-Resolution Radiometer (AVHRR), and France's Système Pour l'Observation de la Terre (SPOT). For more than a decade, this study examined how environmental factors affect health results, emphasizing the significance of incorporating satellite imagery into wider ecological management frameworks. A study [11] analyzed the documented geographic spread of Ixodes scapularis and Ixodes pacificus ticks in the USA between 1982 and 2000 by utilizing satellite data from the National Oceanic and Atmospheric Administration (NOAA). The researchers obtained insights into the distribution patterns of these ticks and their correlation with environmental parameters by utilizing various standardized and point-to-point similarity measurements. This study showcased the significance of satellite images in comprehending the geographical arrangement of disease vectors and providing insights for preventive strategies.

Furthermore, the research [12] specifically examined climate-based prediction and employed health data (acquired from Arbovirus National Reference Centre) and meteorological data (obtained from ECMRWF). The researchers analyzed data from Saint-Georges de l'Oyapock municipal authorities in French Guiana from 1991 to 2006. Their objective was to examine the correlation between climate variables and the occurrence of particular diseases. The study emphasized using satellite-derived climate data to comprehend disease cycles and inform public health initiatives. A study conducted in Brazil [13] utilized the CART algorithm with satellite images acquired from the Landsat 5 Thematic Mapper (TM) to forecast health impacts in the regions of Parnaíba and Poti. The researchers attempted to determine the influence of environmental factors and demographics on disease prevalence by analyzing demographic census data and undertaking a study from 1991 to 2000.

The combination of demographic data and satellite imagery allowed for formulating predictive models, which enhanced comprehension of disease trends and their related determinants. Study [9] developed an SEIR transmission model in Kigali, Rwanda, which included various data sources such as transport data from the Rwanda Transport Development Authority, demographic data from Global Urban Footprint and Worldpop Rwanda, and influenza monitoring data. The researchers gained useful insights into disease transmission dynamics and the impact of population mobility on disease spread by merging multiple datasets. This study demonstrated the importance of satellite data and modeling techniques in offering useful insights for public health policies and resource management.

The research initiatives described in this section demonstrate the significant impact of satellite data and various methodologies on healthcare research. This study utilized satellite images and integrated many datasets to produce noteworthy findings about the relationship between meteorological characteristics and health outcomes. The results of this study can improve decision-making based on evidence, simplify the process of choosing drugs, and provide guidance for future research in the field of satellite-based healthcare applications. By integrating satellite data and analytical methodologies, public health programs can undergo a substantial metamorphosis, leading to enhanced health outcomes worldwide.

5 Key Observations

The critical observations in the healthcare sector have been tabulated in Table 2.

Table 2 Critical observations in satellite-based healthcare research

Observation	Description
Diverse applications	Versatile use of satellite imagery and modeling techniques across various health topics and global geographic locations
Global perspective	Research spans continents (Africa, Asia, North/South America, and Europe), highlighting the worldwide relevance of this field
Methodologies	Diverse methodologies include predictive modeling, logistic regression, spatial–temporal analysis, and more
Satellite data sources	Multiple satellite sources (Landsat, AVHRR, MODIS, SPOT, NOAA, etc.) showcase the wealth of available data for research
Integration of data	Integration of demographic, climate, topographic, and disease data enhances analysis comprehensiveness
Spatial and temporal analysis	Including spatial/temporal factors offers dynamic insights into disease patterns, trends, and hotspots
Targeted interventions	Findings inform practical public health interventions, aiding resource allocation and preventive measure design
Interdisciplinary research	Collaboration among remote sensing, epidemiology, geography, and public health experts strengthens research outcomes
Environmental impact	Investigations into environmental factors deepen understanding of their complex interplay with health outcomes
Predictive modelling	Frequent use of predictive models based on satellite data to forecast disease prevalence and identify risk factors
Resource allocation	Contribution to effective healthcare resource allocation and timely intervention implementation
Paradigm shift	Integration of satellite data and analytical methodologies can revolutionize global public health efforts

6 Conclusion and Future Scope

In conclusion, the wide range of papers evaluated in this table highlights the significant impact that satellite imagery and modeling may have on healthcare research, emphasizing its potential for transformation. This research has expanded our understanding of the complex relationship between satellite data and health outcomes, shedding light on illness patterns, factors contributing to risk, and the significant impact of environmental circumstances. By effectively employing satellite data, researchers have developed predictive models, identified areas of concentrated health concerns, and directed focused initiatives to improve health outcomes. Logistic regression models [6–8, 14, 16], predictive modeling [13, 15, 18], and spatial–temporal analysis [3, 12] have been fundamental methodological approaches utilized in this investigation, further validating the significance of satellite imagery in healthcare research.

Looking forward, the path for advancement in this particular domain is elucidated by several crucial directions. It is of utmost importance to prioritize ongoing research efforts to refine and improve current models, increase their accuracy in predicting outcomes, and broaden on refining and improving current models, increasing their accuracy in predicting outcomes and broadening their relevance in various healthcare settings. Furthermore, integrating sophisticated machine learning and artificial intelligence methodologies holds the potential to unveil novel perspectives in examining sophisticated machine learning and artificial intelligence methodologies to reveal novel views in exploring the potential to unveil novel perspectives in reviewing satellite data, facilitating detailed pattern identification and prediction capabilities. Furthermore, incorporating other data sources, such as demographic and socio-economic data, would provide a comprehensive viewpoint on the complex interplay between environmental determinants and health outcomes. Furthermore, by expanding the geographical range of research to encompass other regions and countries, valuable worldwide insights can be obtained in satellite-based healthcare applications. In conclusion, it is imperative to prioritize collaborative endeavors aimed at democratizing the accessibility of satellite data and analysis tools. This guarantees widespread accessibility and affordability for healthcare practitioners, policymakers, and researchers.

The combination of satellite imagery and healthcare research presents significant potential for promoting public health worldwide. Through the active pursuit of these various lines of study and the cultivation of joint endeavors, we can uncover novel frontiers in our ongoing quest for enhanced healthcare outcomes and overall wellbeing.

References

1. Fuentes MV (2006) Remote sensing and climate data as a key for understanding fasciolosis transmission in the Andes: review and update of an ongoing interdisciplinary project. Geospat Health 1(1):59. https://doi.org/10.4081/gh.2006.281
2. Dambach P, Machault V, Lacaux J-P, Vignolles C, Sié A, Sauerborn R (2012) Utilization of combined remote sensing techniques to detect environmental variables influencing malaria vector densities in rural West Africa. Int J Health Geogr 11(1):8. https://doi.org/10.1186/1476-072X-11-8
3. Anno S et al (2015) Space-time clustering characteristics of dengue based on ecological, socio-economic and demographic factors in northern Sri Lanka. Geospat Health 10(2). https://doi.org/10.4081/gh.2015.376
4. Cheng C, Wei Y, Sun X, Zhou Y (2013) Estimation of chlorophyll-a concentration in Turbid lake using spectral smoothing and derivative analysis. Int J Environ Res Public Health 10(7):2979–2994. https://doi.org/10.3390/ijerph10072979
5. Arboleda S, Jaramillo-O N, Peterson AT (2009) Mapping environmental dimensions of dengue fever transmission risk in the Aburrá Valley, Colombia. Int J Environ Res Public Health 6(12):3040–3055. https://doi.org/10.3390/ijerph6123040
6. Dambach P, Sié A, Lacaux J-P, Vignolles C, Machault V, Sauerborn R (2009) Using high spatial resolution remote sensing for risk mapping of malaria occurrence in the Nouna district, Burkina Faso. Glob Health Action 2(1):2094. https://doi.org/10.3402/gha.v2i0.2094
7. Assaré RK et al (2015) The spatial distribution of Schistosoma mansoni infection in four regions of western Côte d'Ivoire. Geospat Health 10(1). https://doi.org/10.4081/gh.2015.345

8. Dlamini SN, Franke J, Vounatsou P (2015) Assessing the relationship between environmental factors and malaria vector breeding sites in Swaziland using multi-scale remotely sensed data. Geospat Health 10(1). https://doi.org/10.4081/gh.2015.302
9. Randhawa N, Mailhot H, Lang DT, Martínez-López B, Gilardi K, Mazet JAK (2021) Fine scale infectious disease modeling using satellite-derived data. Sci Rep 11(1):6946. https://doi.org/10.1038/s41598-021-86124-2
10. Goetz SJ, Prince SD, Small J (2000) Advances in satellite remote sensing of environmental variables for epidemiological applications, pp 289–307. https://doi.org/10.1016/S0065-308X(00)47012-0
11. Estrada-Peña A (2002) Increasing habitat suitability in the United States for the tick that transmits Lyme disease: a remote sensing approach. Environ Health Perspect 110(7):635–640. https://doi.org/10.1289/ehp.110-1240908
12. Adde A et al (2016) Predicting dengue fever outbreaks in French Guiana using climate indicators. PLoS Negl Trop Dis 10(4):e0004681. https://doi.org/10.1371/journal.pntd.0004681
13. Almeida AS, Werneck GL (2014) Prediction of high-risk areas for visceral Leishmaniasis using socioeconomic indicators and remote sensing data. Int J Health Geogr 13(1):13. https://doi.org/10.1186/1476-072X-13-13
14. Bo Y-C, Song C, Wang J-F, Li X-W (2014) Using an autologistic regression model to identify spatial risk factors and spatial risk patterns of hand, foot and mouth disease (HFMD) in Mainland China. BMC Public Health 14(1):358. https://doi.org/10.1186/1471-2458-14-358
15. Buczak AL, Koshute PT, Babin SM, Feighner BH, Lewis SH (2012) A data-driven epidemiological prediction method for dengue outbreaks using local and remote sensing data. BMC Med Inform Decis Mak 12(1):124. https://doi.org/10.1186/1472-6947-12-124
16. Ageep TB et al (2009) Spatial and temporal distribution of the malaria mosquito Anopheles arabiensis in northern Sudan: influence of environmental factors and implications for vector control. Malar J 8(1):123. https://doi.org/10.1186/1475-2875-8-123
17. Adimi F, Soebiyanto RP, Safi N, Kiang R (2010) Towards malaria risk prediction in Afghanistan using remote sensing. Malar J 9(1):125. https://doi.org/10.1186/1475-2875-9-125
18. de Oliveira EC, dos Santos ES, Zeilhofer P, Souza-Santos R, Atanaka-Santos M (2013) Geographic information systems and logistic regression for high-resolution malaria risk mapping in a rural settlement of the southern Brazilian Amazon. Malar J 12(1):420. https://doi.org/10.1186/1475-2875-12-420
19. Bhunia GS, Kesari S, Jeyaram A, Kumar V, Das P (2010) Influence of topography on the endemicity of Kala-azar: a study based on remote sensing and geographical information system. Geospat Health 4(2):155. https://doi.org/10.4081/gh.2010.197

Achieving Sustainability in Supply Chain During Disruption Times: Role of Industry 4.0

Namit Shrivastava and Manoj K. Srivastava

Abstract The demand to digitalize the automotive sector, which entails linking manufacturers to a larger supply chain, is increasing. In the automobile industry, there is a higher requirement for sustainable growth due to rising supply disruption and frequent technology changes. Industry 4.0 can speed up manufacturing, increase customizability, and cut down on setup and lead times. It may result in innovation. The study focuses on establishing link between Industry 4.0 technologies and green supply chain practices, which will help in achieving sustainability during disruption times. It follows a qualitative survey approach to identify the prominent Industry 4.0 technologies and green supply chain practices using a fuzzy set analytical hierarchy process. The other section uses interpretive structural modelling with a multi-level hierarchical structure to study the cause-and-effect relationship between the final selected Industry 4.0 technologies and GSC practices. The study identifies a strong linkage between Industry 4.0 technologies and green supply chain practices to achieve overall sustainability in the supply chain. The future automotive supply chain should focus on driving Industry 4.0 technologies for effective implementation of green supply chain practices. Also in the Indian automotive sector, government regulation and policies and top management commitment are two key factors for driving sustainability in the Indian automotive supply chain.

Keywords Industry 4.0 · Green supply chain · Fuzzy set analytic hierarchy process

N. Shrivastava (✉)
Birla Institute of Technology and Science, Pilani, India
e-mail: f20201767@pilani.bits-pilani.ac.in; mks@mdi.ac.in

M. K. Srivastava
Management Development Institute, Gurgaon, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_19

1 Introduction

Over the past decade, organizations have been working continuously towards achieving a sustainable and resilient supply chain. Assuring a balance between economic, social, and environmental growth, sustainability is described as meeting the needs of the present without compromising those of future generations. However, with COVID-19 and issues like semiconductor shortages, sea container shortages, the Russia-Ukraine war, and natural disasters such as earthquakes and tsunamis, the automobile industry is currently facing many new challenges in managing their supply chain operations. Such supply chain disruptions result in shortages of critical components, resulting in a loss of production.

The supply chain in the automobile industry in India is under considerable pressure from the government of India and society to pursue a more sustainable model of growth. The automobile industry in India is going through tough times. The growth has slowed down due to a liquidity crunch in the market. COVID-19 has worsened the situation in the first half of the financial year 2020–21. The supply chain in automobiles is playing a crucial role and withstanding the pressure of changing regulatory norms of the government and political dynamics of India, especially with its neighbouring countries [1].

The structure schema of the paper is as follows: First, an introduction highlights the need for going towards sustainability practices and efforts in the supply chains especially in the disruption times. Second, a literature review is presented on Industry 4.0 and green supply chain practices. The third section describes the methodology, including a qualitative survey, a fuzzy set analytic hierarchy process, and interpretive structural modelling. The fourth section discusses the results and cause-and-effect relationship between these technologies and practices. Finally, conclusions, limitations, and future research are explained.

2 Literature Review

Several research articles on “sustainable supply chain” and “Industry 4.0” from relevant web, academic, and research sources are presented below.

2.1 Industry 4.0

Industry 4.0 technologies are considered as the disruptive ones, such as the Internet of things (IoT), 3D printing/additive manufacturing, cloud computing, blockchain, etc. come under the purview of Industry 4.0, which has a huge potential to bring drastic change in the way current manufacturing is done in India. Let us investigate these technologies in detail in a subsequent section.

Additive Manufacturing also known as 3D printing meets all essential requirements for bringing the Industry 4.0 revolution to the manufacturing sector. As the word “additive” suggests, it is the addition of material over its one layer to another in contrast to conventional manufacturing processes where material is chipped off to obtain required dimensions [2–4].

Internet of Things (IoT) is the connection between physical objects or machines and the Internet. It enables the implementation of smart connected products or embedded sensors, resulting in machine-to-machine connectivity through the Internet [5–8].

Blockchain is a type of distributed ledger technology with blocks of records that are linked securely through cryptography. The major benefit of blockchain technology is that it ensures secure data storage and transactions to happen in a transparent and secured manner, preventing any unauthorized interaction during whole process [9–11].

Big Data Analytics (BDA) is characterized by a large volume of data with a wide variety, which requires specific analytical methods to transform it into valuable data. Many organizations are spending a lot of money on training their employees to manage big data using BDA tools. It helps in taking decisions in a structured manner with reliable and real-time analysis [12, 13].

Cloud Computing (CC) refers to the idea that data can be stored, collected, and accessed from specialized shared data centres all over the world. Many organizations are now shifting to cloud-based data storage services such as Office 365, large database solutions, etc. [8].

2.2 Green Supply Chain Practices

Traditionally, the idea of incorporating sustainable supply chain operations is referred to as “sustainable” or “green supply chain” (GSC).

Green Purchasing (GP) is defined as purchasing the product or selecting a supplier that has a lesser effect on the environment or human health as compared to products serving the same purpose. It is also known as “environmentally preferable purchasing”. This includes sourcing recyclable products, reusable raw materials, and products that do not harm the environment [14].

Green Design. Achieving sustainability through a green design approach is better, as under this practice, at the product design stage itself, the design is optimized, taking into consideration the energy and material requirements for manufacturing the design into the final product [15].

Reverse Logistics (RL). It is a type of supply chain management in which the flow of goods occurs from the customer back to the seller or manufacturer. The purpose is to retrieve maximum value from products and material disposed.

Supplier and Customer Collaboration. Sustainability is not a one-time process; it is an act of continual improvement where an organization needs to work together with all its vendor partners to achieve overall excellence [15].

Government Regulation and Policies. Regardless of their form, regulations and policies are the primary drivers for companies to plan and execute sustainable practices of supply chain in their organizations [16].

Top Management Commitment. It is very essential that the top management of an organization is strongly committed to achieving sustainability. This should also be clearly visible in the company's vision and mission statements [16].

3 Research Methodology

The paper uses two stages of data collection through expert opinions and analysis. In the first stage, key Industry 4.0 technologies and green supply chain practices are identified, which are ranked through Fuzzy-Analytic Hierarchy Process (FAHP). Thereafter, the last ranked Industry 4.0 technology and GSC practices are removed, and balance factors are taken for further analysis. In the second stage, interpretive structural modelling (ISM) is used to identify different hierarchical levels. Later, a cross-impact matrix multiplication analysis (MICMAC) is done to find respective power of variables in two categories, namely dependence and driving. Finally, the cause-and-effect analysis is done on the structure obtained through ISM.

3.1 Expert's Profile and Data Collection

For creating contextual links between the variables in the interpretive structural modelling, the expert opinions are used. The selected experts have extensive experience in the automobile and manufacturing domains in India. They are automobile supply chain professionals and have good industry exposure related to the practical application of Industry 4.0 (I4.0) technologies and sustainable practices in supply chains. The experts' profile is highlighted in Table 1.

4 Results and Analysis

Table 2 suggests a list of alternative technologies and GSC practices as per the literature review.

Initially, the AHP matrix is constructed by taking inputs from various experts with large experience in the automotive industry in India. For selecting options among five experts' opinions, the average of all expert opinions is calculated, which is further rounded to the closest integer in the relative importance scale (1–7) to obtain the final AHP matrix. Tables 3, 4, 5, and 6 are showing various steps involved in Fuzzy-Analytic Hierarchy Process (AHP).

Table 1 Experts' profile

S. No	Expert	Experience	Industry	Domain	Designation
1	Expert 1	15	Automobile industry	Research and development	Deputy general manager
2	Expert 2	16	Automobile industry	Supply chain professional	Assistant general manager
3	Expert 3	23	Automobile industry	Supply chain professional	Vice president
4	Expert 4	21	Automobile industry	Product development	Assistant general manager
5	Expert 5	14	Manufacturing domain	Research & Development	Senior manager

Table 2 Set of alternative I4.0 technologies and GSC practices

	Alternative I4.0 technologies		Alternative GSC practices
IoT	Internet of things	GP	Green purchasing
3D	3D printing	SCC	Supplier/customer collaboration
BC	Blockchain	GD	Green design
BDA	Big data analytics	RL	Reverse logistics
CC	Cloud computing	GRP	Govt regulation and policies
		TMC	top management commitment

Table 3 Initial I4.0 technologies matrix (CR = 0.09 < 0.1)

I4.0 Technologies		IoT	3D	BC	BDA	CC
Internet of things	IoT	1.00	3.00	5.00	1.00	3.00
3D printing	3D	0.33	1.00	4.00	1.00	3.00
Blockchain	BC	0.20	0.25	1.00	0.20	0.20
Big data analytics	BDA	1.00	1.00	5.00	1.00	5.00
Cloud computing	CC	0.33	0.33	5.00	0.20	1.00

Table 4 Initial GSC practices matrix (CR = 0.094 < 0.1)

		GP	SCC	GD	RL	GRP	TMC
Green purchasing	GP	1.00	0.33	0.50	2.00	2.00	2.00
Supplier/customer collaboration	SCC	3.03	1.00	3.00	4.00	4.00	3.00
Green design	GD	2.00	0.33	1.00	2.00	2.00	1.00
Reverse logistics	RL	0.50	0.25	0.50	1.00	0.50	0.17
Govt regulation and policies	GRP	0.50	0.25	0.50	2.00	1.00	2.00
Top management commitment	TMC	0.50	0.33	1.00	6.00	0.50	1.00

Table 5 I4.0 technologies weighted fuzzy set matrix

I4.0 Technologies		IoT	3D	BC	BDA	CC	AHP weights (%)	Fuzzy AHP weights (%)	Ranking
Internet of things	IoT	(1,1,1)	(2,3,4)	(4,5,6)	(1,1,1)	(2,3,4)	34	34	I
3D printing	3D	(1/4,1/3,1/2)	(1,1,1)	(3,4,5)	(1,1,1)	(2,3,4)	21	21	III
Blockchain	BC	(1/6,1/5,1/4)	(1/5,1/4,1/3)	(1,1,1)	(1/6,1/5,1/4)	(1/6,1/5,1/4)	5	5	V
Big data analytics	BDA	(1,1,1)	(1,1,1)	(4,5,6)	(1,1,1)	(4,5,6)	30	30	II
Cloud computing	CC	(1/4,1/3,1/2)	(1/4,1/3,1/2)	(4,5,6)	(1/6,1/5,1/4)	(1,1,1)	11	11	IV

Table 6 GSC practices weighted fuzzy set matrix

Practices		GP	SCC	GD	RL	GRP	TMC	AHP weights (%)	Fuzzy AHP weights (%)	Ranking
Green purchasing	GP	(1,1,1)	(1/4,1/3,1/2)	(1/3,1/2,1)	(1,2,3)	(1,2,3)	(1,2,3)	15	15	III
Supplier/customer collaboration	SCC	(2,3,4)	(1,1,1)	(2,3,4)	(3,4,5)	(3,4,5)	(2,3,4)	37	37	I
Green design	GD	(1,2,3)	(1/4,1/3,1/2)	(1,1,1)	(1,2,3)	(1,2,3)	(1,1,1)	16	16	II
reverse logistics	RL	(1/3,1/2,1)	(1/5,1/4,1/3)	(1/3,1/2,1)	(1,1,1)	(1/3,1/2,1)	(1/7,1/6,1/5)	6	6	VI
Govt regulation and policies	GRP	(1/3,1/2,1)	(1/5,1/4,1/3)	(1/3,1/2,1)	(1,2,3)	(1,1,1)	(1,2,3)	11	12	V
Top management commitment	TMC	(1/3,1/2,1)	(1/4,1/3,1/2)	(1,1,1)	(5,6,7)	(1/3,1/2,1)	(1,1,1)	14	13	IV

There is a negligible difference in weights calculated from the normal AHP method and the fuzzy AHP method in both I4.0 technologies and GSC practices. It can be seen from Table 7 that the Internet of things (IoT), 3D printing, and others are the best choices among the alternative technologies for attaining supply chain sustainability. Blockchain is the least preferred choice among I4.0 technologies for achieving sustainability. Also, it can be seen from Table 8 that effective collaboration between supplier and customer is the most preferred choice to achieve sustainable automotive supply chain. However, reverse logistics is the least preferred choice among green supply chain practices.

Further below set of I4.0 technologies and GSC practices are taken to study the cause-and-effect diagram for achieving sustainability.

Table 7 Final selected I4.0 technologies and GSC practices for ISM modelling

	Alternative I4.0 technologies		Alternative GSC practices
IoT	Internet of things	SCC	Supplier/customer collaboration
3D	3D printing	GD	Green design
BDA	Big data analytics	GRP	Govt Regulation and policies
CC	Cloud computing	TMC	Top management commitment
GP	Green purchasing		

Table 8 Structural self-interaction matrix (SSIM)

Structural self-interaction matrix	Factors	3D	BDA	IoT	CC	SCC	GP	GD	TMC	GRP
3D printing	3D	–	A	A	A	X	X	X	A	A
Big data analytics	BDA		–	X	X	X	V	V	A	O
Internet of things	IoT			–	X	X	V	V	A	O
Cloud computing	CC				–	A	V	O	A	O
Supplier/ customer collaboration	SCC					–	X	X	A	A
Green purchasing	GP						–	X	A	A
Green design	GD							–	A	A
Top management commitment	TMC								–	A
Govt regulation and policies	GRP									–

4.1 *Interpretive Structural Modelling (ISM) for I4.0 Technologies and GSC Practices*

In ISM methodology, the use of expert opinions is recommended, as they give their response based on practical experience in the subject domain. For selecting options among five experts' opinions, options with a clear majority are taken, and any arbitration with an option without a clear majority is pursued with revisiting the experts. SSIM was created with input from experts in two prestigious domains: I4.0 and GSC, as below. Tables 8, 9, and 10 are showing various steps involved in ISM modelling.

In order to determine the levels of partitions, each factor's reachability, antecedent, and intersection set are found from the final reachability matrix. Following three iterations levels are obtained from final reachability matrix as shown in Table 10.

The final reachability matrix and levels of partitions are used to develop the ISM model, which incorporates selected I4.0 technologies and GSC practices from fuzzy

Table 9 Final reachability matrix

		1	2	3	4	5	6	7	8	9	
Final reachability matrix	Factors	3D	BDA	IoT	CC	SCC	GP	GD	TMC	GRP	Driving power
13D printing	3D	1	1*	1*	1*	1	1	1	0	0	7
2Big data analytics	BDA	1	1	1	1	1	1	1	0	0	7
3Internet of things	IoT	1	1	1	1	1	1	1	0	0	7
4Cloud computing	CC	1	1	1	1	1*	1	1*	0	0	7
5Supplier/ customer coll	SCC	1	1	1	1	1	1	1	0	0	7
6Green purchasing	GP	1	0	0	0	1	1	1	0	0	4
7Green design	GD	1	0	0	0	1	1	1	0	0	4
8Top management commitment	TMC	1	1	1	1	1	1	1	1	0	8
9Govt regulation and policies	GRP	1	1*	1*	1*	1	1	1	1	1	9
	Dep. Power	9	7	7	7	9	9	9	2	1	

Asterisks are representing transitivities in causal relationships. After incorporating these transitivities, the final reachability matrix is achieved

Table 10 Levels of partitions

Factors	Reachability set	Antecedent set	Intersection set	Level
3D printing	1,2,3,4,5,6,7	1,2,3,4,5,6,7,8,9	1,2,3,4,5,6,7	I
Big data analytics	1,2,3,4,5,6,7	1,2,3,4,5,8,9	1,2,3,4,5	II
Internet of things	1,2,3,4,5,6,7	1,2,3,4,5,8,9	1,2,3,4,5	II
Cloud computing	2,3,4,5,6,7	1,2,3,4,5,8,9	2,3,4,5	II
Supplier/customer collaboration	1,2,3,4,5,6,7	1,2,3,4,5,6,7,8,9	1,2,3,4,5,6,7	I
Green purchasing	1,5,6,7	1,2,3,4,5,6,7,8,9	1,5,6,7	I
Green design	1,5,6,7	1,2,3,4,5,6,7,8,9	1,5,6,7	I
Top management commitment	1,2,3,4,5,6,7,8	8,9	8	III
Govt regulation and policies	1,2,3,4,5,6,7,8,9	9	9	III

AHP technique to ensure sustainability in the automotive supply chain as shown in Fig. 1.

The above ISM model depicts that for achieving sustainability in the automotive supply chain in India, government regulation and policies and top management commitment towards sustainability are the most essential factors. It is followed by the Internet of things, big data analytics, and cloud computing, which are directly or indirectly driving green supply chain practices to achieve overall sustainability. These factors are causing supplier and customer collaboration to be achieved in an effective manner.

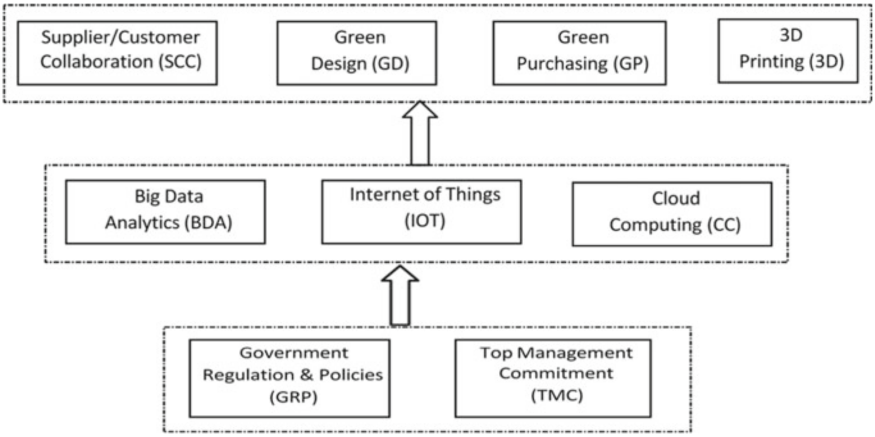


Fig. 1 ISM depicting levels of I4.0 technologies and GSC practices for achieving sustainability. (*Source* Prepared by authors)

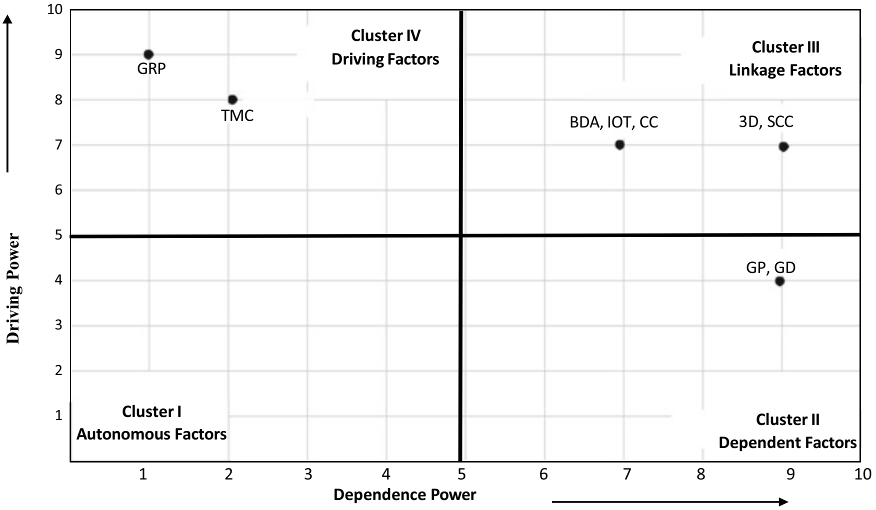


Fig. 2 Factor classification in MICMAC analysis (*Source* Prepared by authors)

4.2 Factor Classification Using MICMAC Analysis

The driving power and dependence power of factors are further examined using a cross-impact matrix multiplication analysis (MICMAC). It is done to pinpoint the primary driving forces behind the system across different domains. There are four sorts of elements based on driving and dependence power: autonomous factors (being less connected to the system and having weak driving & dependence power), linkage factors (having both strong driving and dependence power), dependent factors (having weak driving and strong dependence power) and driving factors (having strong driving power but weak dependence power). The results of the MICMAC analysis are shown in Fig. 2.

There are no autonomous factors present in the MICMAC analysis, as shown in Fig. 2; hence, all the Industry 4.0 technologies, considered, and green supply chain practices are important. Top management commitment and government regulation and policies are driving factors that affect sustainability's performance because they are the driving dependent factors. Among them, government regulation and policies are key factors with the highest driving power.

Big data analytics, the Internet of things, cloud computing, and supplier/customer collaboration are linkage factors. They may have an impact on other system components for achieving sustainability in the supply chain. These factors hold both strong driving and dependent power. Dependent factors include green design and green purchasing, which have high dependence power. Thus, it can be interpreted from the analysis that industry technologies are essential to implementing GSC practices in the automotive supply chain.

5 Conclusion

There is a greater need to digitalize the automobile industry, which involves integrating manufacturers with extended and broad supply chain. The research aims to achieve the linkage between I4.0 technologies and green supply chain practices to achieve sustainability in the automotive supply chain. The study found that I4.0 technologies have a direct or indirect linkage with green supply chain practices.

The study uses a two-stage process wherein, in the first stage, prominent I4.0 technologies and green supply chain practices are selected using a fuzzy set analytical hierarchy process. The expert opinions are validated using the AHP method consistency ratio, which is obtained within the range of 0–10%. The uncertainty in the expert's value preferences is further eliminated using a hesitant fuzzy AHP method to find the final set of I4.0 technologies and GSC practices for further cause-and-effect study. Thereafter, in the second stage, through interpretive structural modelling, the interrelationship between I4.0 technologies and GSC practices is studied for cause-and-effect analysis. The three-level hierarchical structure depicts that government regulation and policies and top management commitment are two essential factors for driving sustainability in the automotive supply chain in India. Further MICMAC analysis supports the driving and dependence power matrix with no autonomous factors in selected variables. Also, it states that major I4.0 technologies such as the Internet of Things, big data analytics, cloud computing are linkage factors that could have an impact on other factors of the system to achieve supply chain sustainability. Thus, it can be interpreted from the analysis that in the automotive supply chain, I4.0 technologies are essential to strengthening GSC practices.

Further investigation can explore the specific challenges and barriers faced in implementing I4.0 technologies and green supply chain practices in the automotive sector. The study focuses on the Indian automotive supply chain, to examine the applicability of the findings in different geographical contexts. The paper highlights the importance of government regulations and top management commitment, but future research can delve deeper into the specific policies and strategies that can drive sustainability in the supply chain.

References

1. Tseng ML, Tan RR, Chiu ASF, Chien CF, Kuo TC (2018) Circular economy meets industry 4.0: Can big data drive industrial symbiosis? Resour Conserv Recycl 131:146–147. https://doi.org/10.1016/j.resconrec.2017.12.028
2. Frazier WE (2014) Metal additive manufacturing: a review. J Mater Eng Perform 23(6):1917–1928. https://doi.org/10.1007/s11665-014-0958-z
3. Chang C-Y, Pan W, Howard R (2017) Impact of building information modeling implementation on the acceptance of integrated delivery systems: structural equation modeling analysis. J Constr Eng Manag 143(8):4017044. https://doi.org/10.1061/(asce)co.1943-7862.0001335

4. Holmström J, Liotta G, Chaudhuri A (2017) Sustainability outcomes through direct digital manufacturing-based operational practices: a design theory approach. J Clean Prod 167:951–961. https://doi.org/10.1016/j.jclepro.2017.03.092
5. Mastos TD, Nizamis A, Vafeiadis T, Alexopoulos N, Ntinas C, Gkortzis D, Tzovaras D (2020) Industry 4.0 sustainable supply chains: an application of an IoT enabled scrap metal management solution. J Cleaner Product 269:122, 377. https://doi.org/10.1016/j.jclepro.2020.122377
6. Rebelo RML, Pereira SCF, Queiroz MM (2022) The interplay between the Internet of things and supply chain management: challenges and opportunities based on a systematic literature review. Benchmarking 29(2):683–711. https://doi.org/10.1108/BIJ-02-2021-0085
7. De Vass T, Shee H, Miah SJ (2021) IoT in supply chain management: opportunities and challenges for businesses in early industry 4.0 context. Operat Suppl Chain Manage: An Int J 14(2):148–161. https://doi.org/10.31387/oscm0450293
8. Huang F, Chen J, Sun L, Zhang Y, Yao S (2020) Value-based contract for smart operation and maintenance service based on equitable entropy. Int J Prod Res 58(4):1271–1284. https://doi.org/10.1080/00207543.2019.1617450
9. Agrawal P, Narain R (2023) Analysis of enablers for the digitalization of supply chain using an interpretive structural modelling approach. Int J Product Perform Manag 72(2):410–439. https://doi.org/10.1108/IJPPM-09-2020-0481
10. Esmaeilian B, Sarkis J, Lewis K, Behdad S (2020) Blockchain for the future of sustainable supply chain management in Industry 4.0. Resources, Conservation and Recycling, 163:64, 105. https://doi.org/10.1016/j.resconrec.2020.105064
11. Jabbar S, Lloyd H, Hammoudeh M, Adebisi B, Raza U (2021) Blockchain-enabled supply chain: analysis, challenges, and future directions. Multimedia Syst 27(4):787–806. https://doi.org/10.1007/s00530-020-00687-0
12. Kache F, Seuring S (2017) Challenges and opportunities of digital information at the intersection of big data analytics and supply chain management. Int J Oper Prod Manag 37(1):10–36. https://doi.org/10.1108/IJOPM-02-2015-0078
13. Gravili G, Benvenuto M, Avram A, Viola C (2018) The influence of the digital divide on big data generation within supply chain management. Int J Logist Manag 29(2):592–628. https://doi.org/10.1108/IJLM-06-2017-0175
14. Min H, Galle WP (2001) Green purchasing practices of US firms. Int J Oper Prod Manag 21(9):1222–1238. https://doi.org/10.1108/EUM0000000005923
15. Vachon S, Klassen RD (2008) Environmental management and manufacturing performance: the role of collaboration in the supply chain. Int J Prod Econ 111(2):299–315. https://doi.org/10.1016/j.ijpe.2006.11.030
16. Centobelli P, Cerchione R, Chiaroni D, Del Vecchio P, Urbinati A (2020) Designing business models in circular economy: a systematic literature review and research agenda. Bus Strateg Environ 29(4):1734–1749. https://doi.org/10.1002/bse.2466

Emotionally Engaged Neurosymbolic AI for Usable Password Generation

Sumitra Biswal

Abstract Password-based authentication remains essential despite the advent of Multi-factor Authentication (MFA). A significant challenge is encouraging users to create strong, memorable passwords, as weak or reused passwords pose considerable security risks. This research introduces the Emotionally Engaged Neurosymbolic AI (EENAI) system, a novel approach for generating usable passwords. It combines neurosymbolic AI and emotional engagement, utilizing valence and arousal in emotionally engaged scenarios. Neurosymbolic AI combines neural network learning with classical AI's symbolic reasoning, ideal for generating context-aware passwords. By integrating valence and arousal, EENAI generates secure, memorable passwords. This paper details EENAI principles, experimental procedures, and observations. Results suggest that EENAI-generated passwords balance security, memorability, and usability, potentially revolutionizing password creation practices. The passwords are further evaluated using a standard password strength estimation tool, yielding promising results. The paper concludes with an EENAI impact assessment and future work recommendations.

Keywords Neurosymbolic AI · Password-based authentication · Emotional intelligence

1 Introduction

In the digital era, the need for secure, user-friendly authentication mechanisms are ever-growing. Despite advances in biometric and Multi-factor Authentication (MFA), password-based authentication remains dominant. A survey shows that only 60% of organizations have adopted password-less methods for accessing IT infrastructure [1]. The challenge lies in the user tendency to opt for easily memorable passwords, often at the expense of security. Data breaches due to weak passwords are alarming;

S. Biswal (✉)
Bosch Global Software Technologies (BGSW), Bosch, India
e-mail: Sumitra.Biswal@in.bosch.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_20

30% of Internet users have fallen victim to such breaches, and 13% of Americans reuse the same password across all accounts [1].

Even with the advent of MFA, credential stuffing remains pervasive. Credentials are implicated in 61% of all breaches, either stolen via social engineering or brute-forced [2]. This suggests that the enduring prevalence of password-based authentication is due to its inherent convenience. To improve upon this method's strengths and weaknesses, this paper introduces a novel approach for generating passwords that are both secure and memorable. The Emotionally Engaged Neurosymbolic AI (EENAI) Password Recommender System is at the crossroads of Artificial Intelligence (AI) and cognitive psychology. It leverages emotional attributes, particularly valence and arousal (VA), to offer personalized password suggestions.

Although prior studies have investigated the effects of emotional attributes on memorability and the impact of emotions on password strength [3, 4], there is a noticeable gap in the existing literature concerning their utilization in the creation of secure passwords. Additionally, while some researches have investigated into the use of random images and the rehearsal of their memorability for password creation [5, 6], these approaches have their own set of limitations, which are elaborated upon in the subsequent "Related Work" section. However, there has been minimal research conducted to associate emotional attributes, such as VA, with different usage domains for the personalization of passwords. The proposed work aims to fill this notable gap in the literature by considering these emotional attributes.

It is essential to note that due to the novel nature of this proposed method, a direct comparative analysis with existing algorithms is not feasible within the scope of this paper. Nonetheless, the research addresses a significant gap in the literature to balance the trade-off between memorability and security of passwords, thereby enhancing their usability. The remainder of this paper is structured as—Sect. 2 reviews related work in the fields of password security, memorability, and emotional attributes. Section 3 explains the principles underlying EENAI. Sections 4 and 5 cover the experimentation process and its results. Finally, Sect. 6 offers conclusions and identifies future avenues for research.

2 Related Work

The body of research associated with password creation, emotional attributes, and their subsequent effects on memorability is vast. This section provides a brief review of these studies, highlighting the significance of each to this research.

Emotional attributes, such as VA, play a crucial role in memory recall. Multiple studies have illustrated the influence of emotions on memory processes [7, 8]. The emotional enhancement effect, where emotionally engaged events are remembered more effectively than neutral ones, is well-documented in these research work. This suggests that integrating emotional attributes into the password creation process may inherently enhance password memorability. Significant investigations have also been made to address the trade-off between password security and usability [9–11]. Strong passwords (i.e., those that are long, complex, and unique) are extremely difficult for users to remember, leading to usability issues. Conversely, passwords that are easy to remember often lack the complexity required for strong security. This trade-off has led to the proposal of several password creation strategies, such as the use of password managers [12] or graphical passwords [13]. In addition to focusing on usability, significant research efforts have been dedicated to enhancing the security aspects of password creation strategies [14–16]. However, few have explored the application of emotional attributes in secure password generation.

While individual studies have explored the roles of emotional attributes in memory, there is a dearth of research combining these elements to create secure and memorable passwords. Some limited research works have connected emotions with passwords, but these do not take into consideration the emotional attributes (VA) of a user's perception toward the use of a certain domain-specific web page login account, the attributes of the page, operational relevance, and purpose of accessing the account of this particular domain-based web page. The domains include but are not limited to Banking, Health Care, E-Commerce, and Education. One notable approach toward enhancing password memorability and strength is the Person-Action-Object (PAO) visual and textual mnemonics model [5, 6]. This approach entails users committing to memory narratives built around PAO, which are crafted by linking machine-created, random pairs of actions and objects with well-known individuals and scenarios. Although this approach showed promising results, it also observed an interference effect, suggesting users found it challenging to memorize many stories at once, likely due to user fatigue [6]. This observation suggests limitations to its scalability and user-friendliness, particularly when users are required to memorize multiple PAO stories.

However, the proposed approach in this research paper aims to bridge this gap by introducing an innovative approach that integrates emotional attributes, specifically VA, into a neurosymbolic AI system for secure password creation. This integration of elements aims to generate passwords that are not only secure and unique but also naturally easier for users to remember due to their emotional associations. Moreover, because it personalizes the memorability and robustness of the password creation process for individual users and domains, this novel proposal can effectively address the interference effect faced in the PAO model.

Additionally, it is pertinent to highlight that the current phase of the proposed work, which involves conceptualization and initial experimentation, has not included participant involvement, unlike other existing research works. However, as the work progresses, participants will be actively involved to provide a more thorough validation of the proposed approach.

3 EENAI: Principle and Mechanism

Neurosymbolic AI (NSAI) has emerged as a powerful tool, capable of harmonizing the inherent strengths of symbolic reasoning from classical AI with the flexible learning capabilities of neural networks. Its ability to adapt to individual users' habits and preferences allows NSAI to facilitate the generation of personalized, secure, yet memorable passwords.

Critical to this methodology are the dimensions of VA in the psychology of emotions. Extensive research indicates that emotional experiences, especially those with high VA, can considerably enhance memory recall. In this context, the generation and recall of passwords are associated with emotionally engaged scenarios that are unique to each user. By capitalizing on the emotional enhancement of memory, passwords are generated that are not only secure but also naturally memorable.

Mnemonic cues are central to the proposed approach. The ideal mnemonic cue should be distinct, emotionally significant, and should have a clear association with the password. Uniqueness is a crucial aspect as it facilitates specific recall of the associated password. Emotional significance leverages the power of emotional experiences to enhance memorability. A clear association between the cue and the password simplifies recall, making the password more user-friendly.

The concept of the memorability index (MI) is introduced as a metric to quantitatively assess the effectiveness of the generated passwords and mnemonic cues. It encapsulates various factors such as the emotional charge of the mnemonic cue, the complexity and length of the password, and the user's personal password preferences and habits. By optimizing the MI during the password generation process, a balance between security and memorability is ensured.

3.1 Modules of EENAI

The EENAI system comprises several modules, each contributing to different stages of the password generation process:

Emotionally Engaged Scenario Generation. This module combines neural network-based image analysis and symbolic AI rules. By leveraging neural networks, the module analyzes the user's input domain, such as a bank's webpage, to extract color scheme, logos, symbols, and other distinguishing features. These extracted colors, along with relevant keywords from the user's input, are then utilized in conjunction with symbolic AI rules to generate personalized and emotionally engaged scenarios. This integrated approach enhances password memorability and security, providing users with a more intuitive and memorable authentication experience.

Password Creation. Once the emotionally engaged scenario is generated in the preceding module, this module employs a blend of symbolic AI and neural networks to assemble a list of passwords, incorporating keywords from the scenario and user input. The symbolic AI applies pre-established password generation rules to these keywords and user input, whereas the neural network calculates the strength of each generated password using a Markov model. Additionally, this module considers the impact of the VA values associated with the keywords on the strength of each password.

Mnemonic Cue Generation. Once a user selects a password, the module generates mnemonic cues by utilizing symbolic AI and mapping techniques such as reverse leetspeak to associate the chosen password, user's personal information, and the emotionally engaged scenario. Subsequently, the efficacy of each cue is assessed by a neural network that estimates its recall ease based on human memory patterns. The module then calculates the MI of each cue, considering various factors such as association strength, emotional intensity, sequence order, cue length, and personalization. These elements collectively influence the overall impact of VA on the effectiveness of the mnemonic and the memorization of the password. The MI formula validates the necessity of each component in evaluating the impact of VA on the memorability of the cue.

$$\mathrm{MI} = \frac{(\text{Association Strength} * \text{Emotional Intensity} * \text{Sequence Factor})}{(\text{Length Factor} * \text{Personalization Factor})} \tag{1}$$

where

1. **Association Strength**: Represents the strength of the relationship between the cue and the password. It is determined by the number of common characters present in both the cue and the password, which reflects their shared characteristics, pertinence, and significance.
2. **Emotional Intensity**: Captures the emotional impact of the cue, influenced by VA, enhancing memorability.
3. **Sequence Factor**: Considers the role of the order of words in the mnemonic cue, in relation to the password, on both memorization and the creation of strong associations.
4. **Length Factor**: Accounts for the influence of cue length on memorability, with shorter cues being easier to remember.
5. **Personalization Factor**: Considers personal connections, experiences, and preferences to enhance memorability.

By combining these factors in the memorability index formula, the impact of VA on mnemonic effectiveness is assessed. EENAI facilitates the creation of passwords that are secure and easily memorable for the user. By integrating neural and symbolic AI, it personalizes passwords based on the user's preferences, ensuring a balance between security and usability. The high-level architecture of EENAI is available in this section (see Fig. 1).

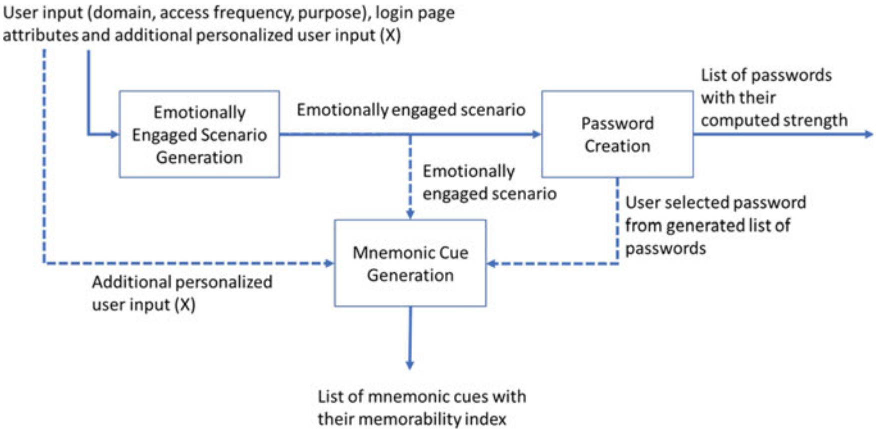

User input (domain, access frequency, purpose), login page attributes and additional personalized user input (X)
Emotionally Engaged Scenario Generation
Emotionally engaged scenario
Password Creation
List of passwords with their computed strength
Emotionally engaged scenario
User selected password from generated list of passwords
Mnemonic Cue Generation
Additional personalized user input (X)
List of mnemonic cues with their memorability index

Fig. 1 Architectural diagram of EENAI

3.2 Comparison with Related Work

While PAO stories [5, 6] also aim to make passwords more memorable, they often lack the multi-dimensional approach that EENAI considers. PAO-based approach generally relies on pre-established imagery and actions to aid in memorization but does not offer the adaptability and personalization inherent to EENAI. Most notably, EENAI considers a range of factors—like VA, user input, and preferences, and web page characteristics—to generate not only a memorable but also a secure password. The integration of symbolic and neural AI enables EENAI to create personalized, emotionally charged mnemonic cues with quantifiable measures of effectiveness (MI), setting it apart from traditional methods like PAO. Therefore, EENAI presents a more balanced, optimized, and personalized approach to password generation and recall.

4 Experimentation

This section of the research is dedicated to the implementation and validation of the EENAI system taking an use case into account, with the primary aim of assessing the impact of emotional attributes, specifically VA, on the generation of secure and memorable passwords. The experiment involves the implementation of different EENAI system modules and evaluating the passwords and cues generated. The process encompasses the following steps:

4.1 Emotionally Engaged Scenario Generation

This subsection details the experimental method employed to integrate neural network analysis and symbolic AI rules to create emotionally engaged scenarios. The process begins with extracting dominant colors from the user's input domain, for example, a bank's webpage, along with key information based on the domain's purpose, access frequency, and usage criticality. This gathered information is then merged with symbolic AI rules to craft emotionally engaging narratives. For example, if the dominant colors identified from a bank's webpage are Floral white, Sandy brown, and Dark slate gray, and the user's purpose for using the banking services includes managing savings and maintaining a salary account, the rules in this module symbolize "Savings" as "A person holding a piggy bank" and "Salary" as "A person counting salary." The symbolic rule for the color gray, for instance, is depicted as "Wrapping a long gray muffler around the neck." The user also inputs additional information, such as the year of birth (1982) and the bank's name (Barclays), which this module incorporates to construct the emotionally engaged scenario—"A person holding a piggy bank. A person counting salary. A person is wrapping a long gray muffler around the neck. The person has entered 1982 and Barclays."

4.2 Password Creation

This module employs symbolic AI rules to craft passwords using the previously generated scenario and user-supplied data. Subsequently, Markov Model calculates the strengths of these passwords. For instance, passwords created from keywords (bearing high VA) derived from the generated scenario, combined with user-provided information, and demonstrating high password strength, include but are not limited to "WRn9SAryNEckGR4y," "BAy5NAm3SAryPE0n," "BAnkBAy5MU3rCOn9," and "SAryNAm3COn9ARnd."

4.3 Mnemonic Cue Generation

This module entails generating mnemonic cues by taking input information such as emotionally engaged scenario, user-selected password, and additional data provided by user during scenario generation. This module maps the input information using techniques like reverse leetspeak. Post mapping, it integrates the keywords derived from the scenario to generate mnemonic cues and simultaneously computes the MI for each cue. For instance, mnemonic cues having high MI derived from a user-selected password (BAnkBAy5MU3rCOn9) include but are not limited to, "Bank Name Muffler Count" with an MI of 0.5998252881975799, "Neck Bank Shopping

Piggy” with an MI of 0.5633858994040093, and “Piggy Name Neck Salary” with an MI of 0.5498422921934262.

5 Observations and Discussion

This section presents the findings from the experiments conducted to assess the impact of VA on password strength and mnemonic memorability, using the EENAI system. The results mentioned here are derived from the experimentation conducted on the use case mentioned in previous section. The analysis includes an examination of the generated passwords using both the internal measures and the widely accepted **zxcvbn** password strength estimation tool [17] which has also been used in the research that studies the effect of negative emotions such as anger on choice of password [3]. This comprehensive approach ensures a robust assessment of the effectiveness of the EENAI system in generating secure and memorable passwords.

In the first graph, the influence of VA of keywords used to generate the password was analyzed. The results showed that higher VA values significantly contributed to increased password strength. When VA were high, the password exhibited greater strength (see Fig. 2).

In the second graph, the impact of VA of keywords in mnemonic cues on mnemonic memorability was investigated. The findings revealed that cues with higher memorability indices had a stronger impact when keywords with high VA values were incorporated. Cues containing keywords with low VA had a comparatively weaker impact on memorability (see Fig. 3).

To corroborate these findings, the **zxcvbn** tool was employed, which provided additional insights into the real-world applicability of the generated passwords. Table 1 comprises the results that shows how the passwords generated with higher

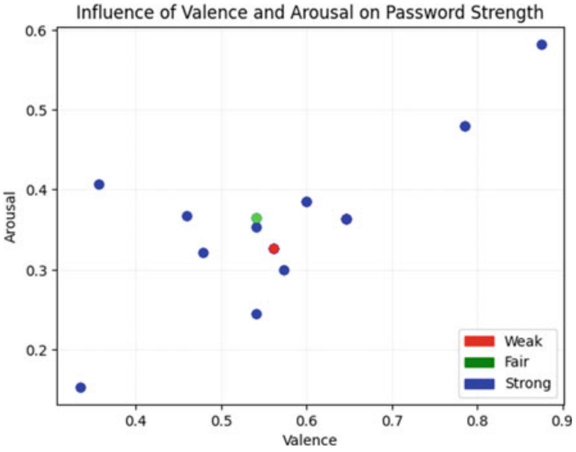


Fig. 2 Impact of valence and arousal on password strengths

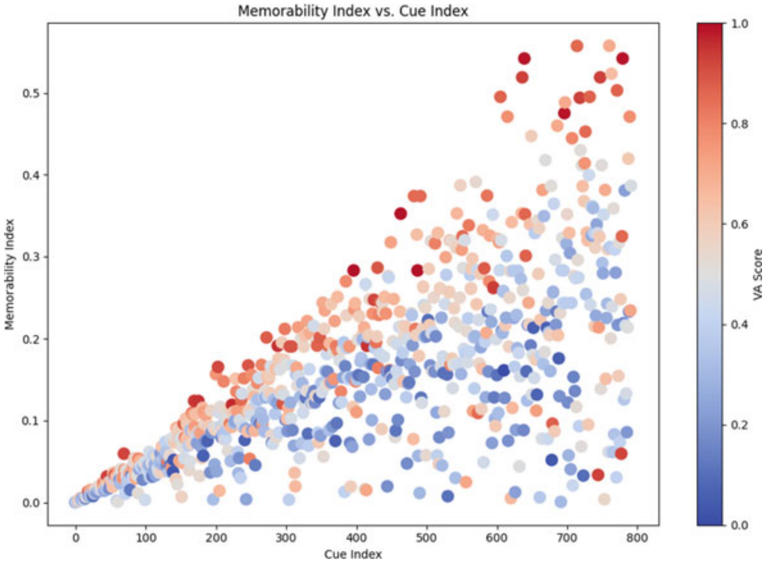


Fig. 3 Impact of valence and arousal on memorability Index

VA and having high MI exhibited greater strength as indicated by higher zxcvbn scores.

All the listed passwords have a zxcvbn score of 4, which is the highest possible score and indicates that the passwords have "Strong protection from offline slow-hash scenario" and are considered "Very unguessable." This is a positive indication of the strength of passwords generated by the EENAI system. Additionally, the guesses_log10 values for all the passwords are relatively high, ranging from approximately 14.90 to 16.00. This indicates that the estimated number of guesses needed to crack these passwords is between $10^{14.90}$ and 10^{16}, which translates to a very large number of guesses, reinforcing the strength of the passwords. The Guess times further break down the estimated time required to crack the password under different attack scenarios:

1. **Throttled online attack (100 attempts per hour)**: For all the listed passwords, the estimated crack time is "Centuries," which indicates an extremely strong resistance to online attacks with rate limiting.
2. **Un-throttled online attack (10 attempts per second)**: Similar to the throttled scenario, the estimated crack time for all passwords is "Centuries," showcasing strong resistance to online attacks without rate limiting.
3. **Offline attack with slow hash and many cores (10 k attempts per second)**: Again, all the listed passwords exhibit a crack time of "Centuries," demonstrating strong resistance to offline attacks with slow hashing algorithms and multiple cores.

Table 1 EENAI generated password assessment using zxcvbn tool

Password	guesses_log10	Score	Guess times
WRn9SAryNEckGR4y	14.90526	4	100/h: centuries (throttled online attack) 10/sec: centuries (unthrottled online attack) 10 k/second: centuries (offline attack, slow hash, many cores) 10B/sec: 22 h (offline attack, fast hash, many cores)
BAy5NAm3SAryPE0n	15.53737	4	100/h: centuries (throttled online attack) 10/sec: centuries (unthrottled online attack) 10 k/sec: centuries (offline attack, slow hash, many cores) 10B /sec: 4 days (offline attack, fast hash, many cores)
3@y$@lP!99Y$#!Rt	16	4	100/h: centuries (throttled online attack) 10/sec: centuries (unthrottled online attack) 10 k/sec: centuries (offline attack, slow hash, many cores) 10B/sec: 12 days (offline attack, fast hash, many cores)
BAnkBAy5MU3rCOn9	15.99913	4	100/h: centuries (throttled online attack) 10/sec: centuries (unthrottled online attack) 10 k/sec: centuries (offline attack, slow hash, many cores) 10B/sec: 12 days (offline attack, fast hash, many cores)
SAryNAm3COn9ARnd	15.58161	4	100/h: centuries (throttled online attack) 10/sec: centuries (unthrottled online attack) 10 k/sec: centuries (offline attack, slow hash, many cores) 10B/sec: 4 days (offline attack, fast hash, many cores)

4. **Offline attack with fast hash and many cores (10 billion attempts per second)**: This is the most aggressive attack scenario. The estimated crack times for the passwords range from "22 h" to "12 days." Although this is a significantly shorter time compared to the other scenarios, it still represents a high level of security as this attack scenario is highly resource-intensive and unlikely to be encountered in most real-world situations.

The passwords produced by the EENAI system are characterized by high MI and robust resistance to a variety of attack scenarios, as demonstrated by their impressive zxcvbn scores, elevated guesses_log10 values, and estimated crack times across different situations. This underscores the EENAI system's ability in crafting secure passwords. Crucially, this analysis elucidates the interplay between VA, password strength, and mnemonic memorability, underscoring the pivotal role of emotional dimensions in both the formulation and recollection of passwords.

Remarkably, the integration of emotionally resonant keywords not only amplified the mnemonic recall of the cues but also facilitated the construction of passwords that satisfied widely recognized benchmarks of password strength, as affirmed by the zxcvbn assessment. This innovative approach, leveraging emotional cues to craft passwords, stands out as a unique strategy that yields both memorable and robust passwords. Moreover, the highly personalized mnemonic cues generated can be employed to create distinct, usable passwords for various domains and login accounts, thereby addressing the interference effect highlighted in prior research and aiding in the mitigation of attacks such as credential stuffing.

6 Conclusion and Future Work

This research study investigated into the association of emotionally engaged scenarios and mnemonic cues in the crafting of secure and memorable passwords. The experimental findings underscored the salutary impact of VA on both the robustness and recall of passwords. The confluence of symbolic AI rules, neural network scrutiny, and Markov Models generated a holistic framework for password creation and assessment. The outcomes emphasized the necessity of factoring in emotional dimensions during password formulation, as elevated VA indices markedly bolstered password robustness. Moreover, the recall of mnemonic cues was augmented when keywords imbued with high VA were harnessed. These insights highlight the viability of capitalizing on emotional associations to engineer more robust and memorable passwords.

Subsequent research in this domain could traverse numerous pathways for refinement and broadening. Primarily, the inclusion of a more expansive lexicon of emotionally engaging words and a wider spectrum of VA indices could furnish a more thorough investigation toward improvement of password robustness and recall. Additionally, probing the effects of individual variances in emotional reactions and personal affiliations to scenarios could amplify the customization facet of mnemonic cues. Furthermore, examining disparate symbolic AI rule sets, neural network configurations, and machine learning algorithms could enhance the password creation process and refine the precision of robustness evaluations. Assessing the user-friendliness and acceptance of the generated passwords and mnemonic cues through user studies and involving participants would yield invaluable insights for practical applications.

Moreover, a critical examination of potential security vulnerabilities and addressing any privacy apprehensions linked to the gathering and analysis of user data would be paramount for the real-world deployment of these techniques. Lastly, the incorporation of supplementary elements, such as contextual data and behavioral biometrics, could bolster password security and user-friendliness. Collectively, this research paves the way for harnessing emotional associations and mnemonic cues in password creation, presenting promising avenues for strengthening password robustness and recall. By addressing the aforementioned areas for future exploration, this approach harbors the potential to wield a transformative influence on the domain of password security.

Acknowledgements The NRC Valence, Arousal, and Dominance Lexicon (Non-Commercial Use Only—Research or Educational) obtained from Saif Mohammad's website (https://saifmohammad.com/WebPages/nrc-vad.html) [18] was used as a valuable resource for analyzing the emotional attributes of the textual data in this research. Its inclusion greatly contributed to the experimentations conducted.

References

1. Howarth J (2023) 50+ Password statistics: the state of password security in 2023 (2023), https://explodingtopics.com/blog/password-stats. Last accessed 03 Sept 2023
2. Jones C (2023) 50 Identity And Access Security Stats You Should Know In 2023 (2023), https://expertinsights.com/insights/50-identity-and-access-security-stats-you-should-know/. Last accessed 03 Sept 2023
3. Khan L, Coopamootoo KPL, Ng M (2020) Not annoying the user for better password choice: effect of incidental anger emotion on password choice. In: Moallem A (eds) HCI for Cybersecurity, privacy and trust. HCII 2020. Lecture Notes in Computer Science(), vol 12210. Springer, pp 143–161
4. Coopamootoo KPL (2020) Empathy as a response to frustration in password choice. In: Berhard M et al (2020) Financial cryptography and data security. FC 2020. Lecture Notes in Computer Science, vol 12063. Springer, pp 177–191
5. Blocki J, Blum M, Datta A (2013) Naturally rehearsing passwords. In: Sako K, Sarkar P (eds) Advances in cryptology—ASIACRYPT 2013. ASIACRYPT 2013. Lecture Notes in Computer Science, vol 8270. Springer, Berlin, Heidelberg, pp 361–380
6. Blocki J, Komanduri S, Cranor L, Datta A (2014) Spaced repetition and mnemonics enable recall of multiple strong passwords. arXiv preprint arXiv:1410.1490 [cs.CR]
7. Custodio J, Justel N (2023) Stress and Novelty: two interventions to modulate emotional memory in adolescents. J Cogn Enhanc 7:39–50
8. Hou TY, Cai WP (2022) What emotion dimensions can affect working memory performance in healthy adults? A review. World J Clin Cases. 10(2):401–411
9. Di Nocera F, Tempestini G (2022) Getting rid of the usability/security trade-off: a behavioral approach. J Cybersecur Privacy 2(2):245–256
10. Bhana B, Flowerday SV (2022) Usability of the login authentication process: passphrases and passwords. Inf Comput Secur 30(2):280–305
11. Rodriguez JJ, Zibran MF, Eishita FZ (2022) Finding the middle ground: measuring passwords for security and memorability. In: IEEE/ACIS 20th international conference on software engineering research, management and applications (SERA). IEEE, Las Vegas, NV, USA, pp 77–82

12. Wang L, Li Y, Sun K (2016) Amnesia: a bilateral generative password manager. In: IEEE 36th international conference on distributed computing systems (ICDCS). IEEE, Nara, Japan, pp 313–322
13. Andriotis P, Kirby M, Takasu A (2023) Bu-Dash: a universal and dynamic graphical password scheme (extended version). Int J Inf Secur 22:381–401
14. Segreti SM, Melicher W, Komanduri S, Melicher D, Shay R, Ur B et al (2017) Diversify to survive: making passwords stronger with adaptive policies. In: Proceedings of the thirteenth USENIX conference on usable privacy and security (SOUPS 2017), USENIX Association, USA, pp 1–12
15. Habib H, Naeini PE, Devlin S, Oates M, Swoopes C, Bauer L et al (2018) User behaviors and attitudes under password expiration policies. In: Proceedings of the fourteenth USENIX conference on usable privacy and security (SOUPS 2018), USENIX Association, USA, pp 13–30
16. Davis DK, Chowdhury MM, Rifat N (2022) Password security: what are we doing wrong? In: 2022 IEEE international conference on electro information technology (eIT), Mankato, MN, USA, pp 562–567
17. Sturge AJ (2022) Zxcvbn Password Strength Estimator. https://infosecwriteups.com/implementing-zxcvbn-a-password-strength-estimator-96192af9800a. Last accessed 03 Sept 2023
18. Mohammad SM (2023) The NRC Valence, Arousal, and Dominance (NRC-VAD) Lexicon. https://saifmohammad.com/WebPages/nrc-vad.html. Last accessed 03 Sept 2023

Exploring the Deep Learning Techniques in Plant Disease Detection: A Review of Recent Advances

Saurabh Singh and Rahul Katarya

Abstract In agriculture, protecting crop yield is one of the most critical aspects of avoiding crop waste and ensuring food security around the world. One of the most critical aspects of preserving yield is protecting it from pests and plant diseases. With the advancement in the field of Artificial Intelligence (AI), it has been applied to different domains, and one such field is agriculture, where we can incorporate AI. Deep learning (DL), which is a subset of Artificial Intelligence, has gained lots of attention toward plant disease detection in the present day because of its better accuracy and performance in comparison with other techniques like machine learning (ML), etc. In this paper, we provide a comprehensive review of the current research work by utilizing deep learning for plant disease detection. We study the different models and architectures proposed by different authors and try to identify the pros and cons of the proposed methodology. We also discuss the various datasets that have been used in research work for detecting plant diseases. Finally, we describe the possible challenges in implementing deep learning models and discuss the future roadmap that can be followed by trying to identify the research gaps.

Keywords Agriculture · Plant diseases · Artificial Intelligence (AI) · Deep learning (DL) · Machine learning · Plant disease detection

1 Introduction

The agricultural sector holds significant importance in the economy, serving as the primary means of sustenance for a significant portion of the population and playing a vital role in livelihoods worldwide. Here's a brief overview of agriculture in India and the world.

S. Singh · R. Katarya (✉)
Delhi Technological University, New Delhi, India
e-mail: rahuldtu@gmail.com

S. Singh
e-mail: saurabh_2k22afi20@dtu.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_21

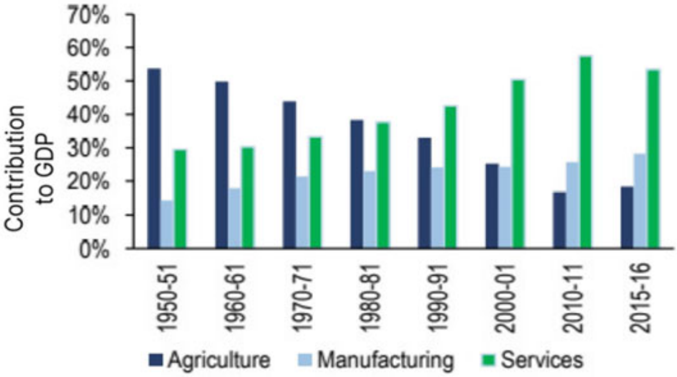


Fig. 1 Sector-wise contribution to GDP of India [2]

Agriculture in the World: Over 26% of the world's workforce is employed in the agriculture sector, which is an important one. The agricultural sector only made up 4.00% of the global GDP, while in low-income nations it accounts for an average of 30.00% of the GDP. International trade in agriculture contributes to meeting the various demands of nations and the availability of food [1].

Agriculture in India: In India, agriculture employs about 50% of the population and generates close to 17.5% of the GDP. Figure 1 shows the contribution of different sectors to the GDP of India over the years. It plays a crucial role in the country's development, alleviation of poverty, and food security. India is a major producer of several goods, including rice, wheat, pulses, oil seeds, fruits, and vegetables [2].

We can see agriculture holds great significance on a global scale and any disruption in agriculture can have significant impacts on various aspects, including food security, economy, environment, and social well-being. Agriculture faces various challenges, including fragmented landholdings, dependence on monsoons, crop damage due to plant diseases, water scarcity, inadequate infrastructure, post-harvest losses, farmer indebtedness, and market volatility. One such biggest challenge faced in agriculture is plant diseases. Plant diseases can lead to significant economic losses by reducing crop yields, quality, and market value. They can affect both food crops and cash crops, impacting farmers' livelihoods and global food supplies.

The five main crops grown across the world include wheat, rice, maize, soybeans, and potatoes—contributing roughly 18.3, 18.9, 5.4, 3.3, and 2.2 percent of the calories consumed worldwide, respectively. Each of these crops has an estimated 21.5, 30.0, 22.6, 21.4%, and 17.2% loss worldwide as a result of illnesses and pests infecting these plants [3]. Figure 2 shows the percentage of crop loss caused by different plant pathogens.

At the current time deep learning [5] has spread across different domains and areas with advancements in hardware and technologies and the same can be employed in the field of agriculture. In this paper, we try to review different plant disease detection

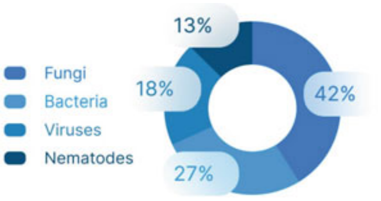


Fig. 2 Crop loss by plants pathogens [4]

techniques based on deep learning. This study's key contribution can be summed up as follows:

- To provide a comprehensive review of the current research work by utilizing deep learning for plant disease detection.
- Study and discuss the pros and cons of different models and architectures proposed for plant disease detection.
- Identify the challenges and limitations for future research direction.

The paper is organized with Sect. 2 giving background about plant diseases and deep learning and its components. Section 3 provides information on different plant disease datasets available and the latest research work done utilizing deep learning for plant disease detection. Section 4 describes the various limitations and research gaps that need to be addressed in the future. Section 5 gives the conclusion of this study work.

2 Background

2.1 *Plant Diseases*

Plant diseases refer to the abnormal conditions or disorders that affect plants, leading to a decline in their health, growth, and productivity. These diseases can be caused by various factors, including pathogens, parasitic plants, and abiotic factors (such as nutrient deficiencies, extreme temperatures, and pollution). Here, we discuss different types of plant diseases [6]:

- **Fungal Diseases**: Fungi can impact various parts of plants including leaves, stems, fruits, and roots. Some common types of plant fungal diseases include powdery mildew, rust, downy mildew, smuts, leaf spots, and root rots.
- **Bacterial Diseases**: Bacteria can infect plants and cause various diseases like bacterial blight, bacterial canker, crown gall, and bacterial wilt. Wilting, leaf spots, cankers, or galls are some examples of the symptoms of bacterial infections.

- **Viral Diseases**: Viruses, such as mosaic viruses, leaf curl viruses, and necrotic ring spot viruses, are contagious agents that can harm plants. Symptoms of viral infections frequently include stunted or deformed development, yellowing, mosaic patterns, and leaf mottling.
- **Nematode Diseases**: Microscopic worms that can parasitize plant roots are the cause of nematode infections, which include cyst nematodes and root-knot nematodes. Plants that are impacted could exhibit signs of growth stunting, root galling, or nutritional shortages.
- **Parasitic Plant Diseases**: Diseases caused by parasitic plants include dodder and witchweed, which physically attach to their host plants and siphon off nutrients and water.
- **Abiotic Diseases**: Various abiotic factors like extreme temperatures, nutrient imbalances or deficiencies, water stress, chemical toxicity, or air pollution can cause plant illnesses. These illnesses can cause tissue death (necrosis), chlorosis (yellowing), or an overall decline in plant health.

It's important to note that the specific types and names of plant diseases can vary across different plant species, regions, and environmental conditions. Figure 3 shows sample images of some of the plant diseases. Proper diagnosis and identification of plant diseases are crucial for implementing effective disease management strategies.

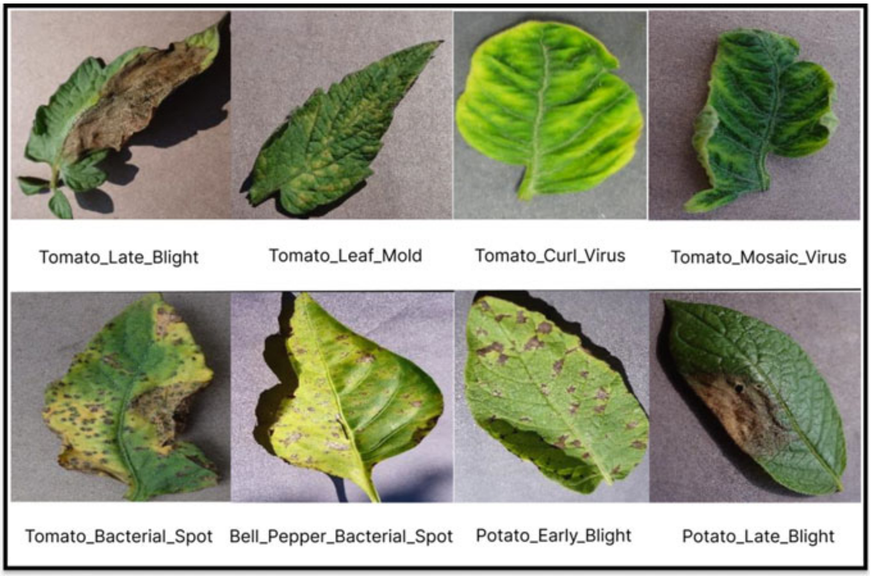


Fig. 3 Sample images of plant diseases

2.2 Deep Learning

Deep learning is a subfield of machine learning that focuses on building artificial neural networks that can learn from complicated data and make predictions based on that data. These networks are modeled after the structure and operation of the neural network in the human brain. The key components of a deep learning architecture are as follows:

- **Input Layer**: It is the first layer in the model that receives the input data or features that are fed into the deep learning model. A feature or property of the input data is represented by each neuron in the input layer. The input layer is connected to hidden layers.
- **Hidden Layers**: Deep learning models have one or more hidden layers associated with them. The hidden layer is composed of multiple neurons that are connected to one another. The weighted sum of the inputs received from the last layer is performed for each neuron in the hidden layer, which is followed by applying the activation function to finally produce an output. The hidden layer helps the network learn complex information and extract hierarchical features from the input data.
- **Activation Functions**: Activation functions are used to add nonlinearity to the network. Some of the commonly used activation functions are sigmoid, hyperbolic tanh, Rectified Linear Unit, etc. This nonlinearity helps the network learn the complex patterns and relationships from the data.
- **Weight Parameters**: The connections between neurons in different layers are associated with weight parameters. These weights are adjusted to minimize the loss function during the training of the model, using techniques like gradient descent and backpropagation.
- **Output Layer**: It is the final layer of the model which produces the final output or predictions based upon learned knowledge from the previous layers. The activation function and neurons in the final layer depend upon the nature of the task being performed.
- **Loss Function**: The difference between the actual output and the expected output is measured by the loss function. During model training, the weights are adjusted using the gradients of the loss function to enhance the model's ability to predict outcomes.
- **Optimization Algorithm**: Optimization algorithms are used to adjust the weights and bias associated with the neurons using the gradients of the loss function. Some commonly used optimization algorithms are stochastic gradient descent, Adam, AdaGrad, etc. These algorithms decide how much weight should be updated and in what direction to iteratively boost the model's performance.

Deep learning architectures can vary depending on the specific task or application. Well-known architectures encompass convolutional neural networks (CNNs) [7] that excel in image and video processing, recurrent neural networks (RNNs) [7] which are ideal for handling sequential data, and transformer models that are widely used in

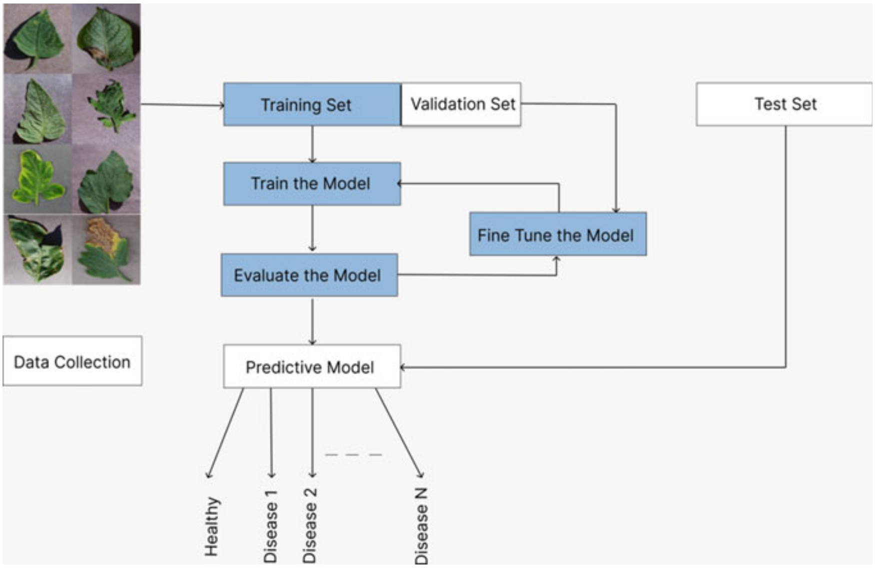


Fig. 4 General deep learning framework for plant disease detection

natural language processing tasks. These architectures incorporate specialized layers and components to handle the unique characteristics and challenges of different data types and tasks. Figure 4 shows a general deep learning framework for plant disease detection.

3 Related Work

3.1 Literature Survey

In recent years, extensive work has been done using deep learning toward plant disease detection. This literature review seeks to give an overview of the major advancements and developments in deep learning-based plant disease detection. The literature survey highlights the rapid advancements and significant contributions of deep learning techniques in plant disease detection. Deep learning-based models tend to show better accuracy and performance in comparison with other methods provided quality data is available to train and evaluate the model. Some of the recent works are discussed below:

Gehlot et al. [8] proposed an architecture named EffiNet-TS, which consists of two classifiers and one decoder. The proposed model is based on state-of-the-art Teacher/Student architecture built around EfficientNetV2. The suggested model highlights the important feature for classifying plant disease, which improves classification and offers a clearer visual representation of specific plant disease symptoms. Many authors have used state-of-the-art object detection algorithms with some internal modifications to improve the performance. Li et al. [9] proposed an integrated model that combines single-stage and two-stage target detection networks. The single-stage network is built upon the YOLO with internal structure optimization. The two-stage network, on the other hand, is based on the Faster R-CNN (Region Convolutional Neural Network). Initially, the target frames are clustered using clustering techniques followed by the integration of two models to perform the disease detection task. Another author Mahum et al. [10] proposed a model based on DenseNet-201 for classifying potato leaves into five different categories.

Some of the authors have proposed a novel deep learning model for plant disease detection like Yu et al. [11] and Ramamoorthy et al. [12] both the authors proposed a novel deep learning-based model with the former proposing a model based on inception convolution and vision transformer and the latter based on MobileNet V1 architecture. Wang et al. [13] proposed a lightweight model which is based on state-of-the-start YOLOV5 architecture for plant disease detection which has better accuracy and performance in comparison with other state-of-the-art techniques. Another author Elaraby et al. [14] proposed a model based on AlexNet for classifying the diseases in plants and use of Particle Swarm Optimization (PSO) for feature selection which helped in optimizing the performance of the overall model.

Saleem et al. [15] proposed a model named region-based fully convolution network (RFCN). The author also studied the use of different data augmentation techniques and the effect of hyperparameter tuning on the performance of the model. Shah et al. [16] proposed a model named ResTS (Residual Teacher/ Student) which is based on CNN architecture. It consists of two classifiers and a decoder. The proposed model is capable of finer visualization for disease detection. Panchal et al. [17] performed a comparative study on four models, namely Inception-v3, ResNet50, VGG16, and VGG19. The author also performed parameter tuning on these models to obtain better results and gave insights for each of these models.

The studies discussed demonstrate the effectiveness of deep learning models, such as CNNs, in accurately identifying and classifying plant diseases across various crops and imaging modalities. In Table 1, we highlight the key findings of our literature review and perform a comparative study of the different models proposed for plant disease detection. The various deep learning approaches that have been studied can prove to be highly beneficial to the farmers in monitoring and identifying the different types of diseases and taking appropriate action.

3.2 Datasets

Datasets play a crucial role in deep learning-based models, and they tend to show better accuracy and performance in comparison with other methods provided quality data is available to train and evaluate the model. Some of the commonly used datasets are discussed in brief and mentioned in Table 2.

Table 1 Literature survey of recent deep learning-based plant disease detection

Year, References	Model used	Performance	Dataset	Pros	Cons
2023, [8]	EfficientNetV2	F1 score: 0.989, Accuracy: 0.990	PlantVillage dataset	Better performance in comparison with the state-of-the-art architecture of ResTS	The dataset does not represent real-world crop scenarios Focuses only on the leaf organ of plants. Diseases are associated with other organs like fruits, stems, etc.
2023, [9]	YOLO + RCNN	Accuracy 85.2%	A self-made dataset consisting of 7199 images belonging to 6 species namely peach, pepper, potato, squash, tomato, and strawberry	Computationally fast and efficient	The model accuracy is improved but at the cost of operation speed The quality and quantity of the dataset is questionable
2023, [10]	Efficient DenseNet	Accuracy 97.2%	PlantVillage dataset + manually gathered data	Computationally fast and efficient	Focuses only on potato leaf disease detection The quality of the dataset is questionable
2023, [11]	CNN	Accuracy (PlantVillage): 99.94, Accuracy (ibean): 99.22, Accuracy(AI2018):86.89, Accuracy (PlantDoc):77.54	PlantVillage dataset + ibean leaf image dataset + AI2018 + PlantDoc dataset	Better performance in comparison with the state-of-the-art models	Focuses only on the leaf organ of plants. Diseases are associated with other organs like fruits, stems, etc.

(continued)

Table 1 (continued)

Year, References	Model used	Performance	Dataset	Pros	Cons
2023, [12]	CNN	Accuracy of 95%	PlantVillage dataset	Simple and efficient	Focuses only on the leaf organ of plants. Diseases are associated with other organs like fruits, stems, etc.
2022, [13]	Optimized YOLOv5	Precision 93.73, Recall 92.94, Accuracy (PlantDoc): 90.26% Accuracy (Peanut Rust): 92.57%	PlantVillage dataset, PlantDoc, and self-made Peanut Rust dataset	The proposed model, i.e., optimized YOLOv5 is efficient in terms of memory and operation time in comparison with other networks	The dataset does not represent real-world crop scenarios Focuses only on the leaf organ of plants. Diseases are associated with other organs like fruits, stems, etc.
2022, [14]	AlexNet + PSO	Accuracy 98.83	The dataset consists of nearly 13,000 images of 5 crop species	Significant improvement in accuracy of AlexNet from 95.6 to 98.83 with PSO	Room for improvement in the quality of the dataset for plant disease detection
2022, [15]	Region-based fully convolutional network (RFCN)	mAP 93.8%	NZDLPlantDisease-v1	Focuses on other organs of plants like fruits, stems, etc., instead of just leaves	Disease detection for multiple organs is not considered in all plant species The dataset is limited to fields from New Zealand

(continued)

Table 1 (continued)

Year, References	Model used	Performance	Dataset	Pros	Cons
2021, [16]	CNN	F1 score of 0.991	PlantVillage dataset	Better performance in comparison with the state-of-the-art Teacher/Student architecture	Requires more memory and operation time in comparison with benchmark Teacher/Student architecture Focuses only on the leaf organ of plants. Diseases are associated with other organs like fruits, stems, etc.
2021, [17]	CNN	Accuracy of 93.5%	PlantVillage dataset	Simple and efficient model	More training data could have helped further improve the accuracy of the model Focuses only on the leaf organ of plants. Diseases are associated with other organs like fruits, stems, etc.

Table 2 Datasets used in plant disease detection

Dataset name	No. of images	Year released
Plant pathology 2021—FGVC8 [18]	23,249	2021
DiaMOS plant dataset [19]	3505	2021
PlantDoc [20]	2598	2019
RoCoLe dataset [21]	1560	2019
BRACOL dataset [22]	1747	2019
PlantVillage dataset [23]	54,303	2015

The Plant Pathology 2021—FGVC8 [18] is a dataset for apple foliar disease detection in apple plants released in the year 2021. Another dataset released in the year 2021 is the DiaMOS [19] plant dataset belonging to pear fruit which consists of 3505 images of leaves belonging to four different classes of disease.

The PlantDoc [20] dataset consists of 2598 images of 13 plant species belonging to 17 different classes of disease released in the year 2019. RoCoLe [21] dataset was also released in the year 2019 focusing on the Robusta Coffee Leaf image divided into 6 classes. BRACOL [22] dataset was also released in the year 2019 consisting of 1747 images focusing on Arabica coffee leaves affected by different diseases belonging to five different classes.

PlantVillage [23] is one of the most extensively used datasets in different studies. The PlantVillage dataset consists of high-quality images of diseased and healthy plant leaves captured under controlled conditions. It consists of a total of 54,303 images spanning across 14 crop species. The dataset contains images depicting a wide range of plant diseases caused by fungi, bacteria, viruses, and other pathogens.

4 Research Gap

From the above literature review in Table 1, we identify various limitations that can be addressed in the future. Following are the research gaps we could identify from this study:

- **Generalization**: Most of the research work has been performed using the PlantVillage dataset but it may not represent a practical real-world scenario as it was developed under a controlled environment. The generality of the algorithms is impacted by this issue, making them unsuitable for real-world deployment.
- **Limited scope**: Most of the study focus only on plant leaf disease detection but diseases may also be associated with other organs of plant like stem, fruits, etc. Hence there is a need for comprehensive plant disease detection beyond the leaf organ of plants.

- **Recognition efficiency at the cost of inference efficiency**: The vast majority of current studies concentrate on recognition efficiency while ignoring inference efficiency, which restricts their practical real-world application.
- **Small datasets**: Dataset is very crucial for deep learning-based models for better performance and accuracy. There are limited and small datasets for plant disease detection which needs to be addressed.
- **Multiple disease detection**: Furthermore, it hasn't been done to simultaneously detect several illnesses in a single plant organ. Also, the effectiveness of the same optimized/modified model has not been studied in complicated horticulture settings consisting of different crops.

5 Conclusion

This paper aims to provide a comprehensive review of the latest research work by utilizing deep learning for plant disease detection. Large datasets and advances in deep learning architectures have allowed researchers to detect plant diseases with high accuracy rates, outperforming earlier techniques and enabling quick and accurate diagnosis. Deep learning is advantageous for detecting plant diseases because it has the potential for real-time monitoring and is scalable for use in large-scale agricultural applications. Deep learning algorithms can be used to develop intelligent systems that can identify diseases early on, assisting with timely disease management decisions. The various deep learning approaches that have been studied can prove to be highly beneficial to the farmers in monitoring and identifying the different types of diseases and taking appropriate action, which would ultimately prevent crop wastage and financial loss to the farmer. After a thorough analysis and study, we could identify some limitations and research gaps that could be addressed in the future work. Future research directions might also involve combining deep learning with cutting-edge technologies like drones and the Internet of Things (IoT).

In conclusion, deep learning-based plant disease detection has enormous potential to transform crop protection and disease management techniques. Deep learning techniques will continue to evolve, along with the incorporation of complementing technology, opening the door for more precise, effective, and sustainable agricultural practices that will ultimately improve crop health and contribute to global food security.

References

1. Anik R, Asif, SR, Sarker JR (2020) Five decades of productivity and efficiency changes in world agriculture (1969–2013). Agriculture 10(6):200
2. Deshpande T (2017) State of agriculture in India. PRS Legislative Res 53(8):6–7
3. Savary S, Willocquet L, Pethybridge SJ, Esker P, McRoberts N, Nelson A (2019) The global burden of pathogens and pests on major food crops. Nat Ecol Evolut 3(3):430–439

4. Khan MR, Sharma RK (2020) Fusarium-nematode wilt disease complexes, etiology and mechanism of development. Ind Phytopathol 73(4):615–628
5. Hao X, Zhang G, Ma S (2016) Deep learning. Int J Semant Comput 10(03):417–439
6. Singh RS (2018) Plant diseases. Oxford and IBH Publishing
7. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8:1–74
8. Gehlot M, Gandhi GC (2023) EffiNet-TS: a deep interpretable architecture using EfficientNet for plant disease detection and visualization. J Plant Diseases Protect 130(2):413–430
9. Li M, Cheng S, Cui J, Li C, Li Z, Zhou C, Lv C (2023) High-performance plant pest and disease detection based on model ensemble with inception module and cluster algorithm. Plants 12(1):200
10. Mahum R, Munir H, Mughal Z-U-N, Awais M, Khan FS, Saqlain M, Mahamad S, Tlili I (2023) A novel framework for potato leaf disease detection using an efficient deep learning model. Human and Ecological Risk Assessment: Int J 29(2):303–326
11. Yu S, Xie Li, Huang Q (2023) Inception convolutional vision transformers for plant disease identification. Internet of Things 21:100650
12. Ramamoorthy R, Saravana Kumar E, Naidu RCA, Shruthi K (2023) Reliable and accurate plant leaf disease detection with treatment suggestions using enhanced deep learning techniques. SN Comput Sci 4(2):158
13. Wang H, Shang S, Wang D, He X, Feng K, Zhu H (2022) Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. Agriculture 12(7):931
14. Elaraby A, Hamdy W, Alruwaili M (2022) Optimization of deep learning model for plant disease detection using particle swarm optimizer. Comput Mater Cont 71(2)
15. Saleem MH, Potgieter J, Mahmood Arif K (2022) A performance-optimized deep learning-based plant disease detection approach for horticultural crops of New Zealand. IEEE Access 10:89798–89822
16. Shah D, Trivedi V, Sheth V, Shah A, Chauhan U (2022) ResTS: residual deep interpretable architecture for plant disease detection. Inf Proc Agricul 9(2):212–223
17. Panchal AV, Patel SC, Bagyalakshmi K, Kumar P, Khan IR, Soni M (2023) Image-based plant diseases detection using deep learning. Mater Today: Proc 80:3500–3506
18. Thapa R, Zhang K, Snavely N, Belongie S, Khan A (2020) The plant pathology challenge 2020 data set to classify foliar disease of apples. Appl Plant Sci 8(9):e11390
19. Fenu G, Malloci FM (2021) DiaMOS plant: a dataset for diagnosis and monitoring plant disease. Agronomy 11(11):2107
20. Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N (2020) PlantDoc: a dataset for visual plant disease detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp 249–253
21. Parraga-Alava J, Cusme K, Loor A, Santander E (2019) RoCoLe: a robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition. Data Brief 25:104414
22. Krohling RA, Esgario J, Ventura JA (2019) BRACOL–a Brazilian Arabica coffee leaf images dataset to identification and quantification of coffee diseases and pests. Mendeley Data 1
23. Hughes D, Salathe M (2015) An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing

A Robust Driver Distraction Estimation Technique for ADAS Applications

Sriman Sathish, S. Ashwin, S. Manish, Nishanth S. Shukapuri, Mayur S. Gowda, and Viswanath Talasila

Abstract Road accidents account for significant economic and personal costs, and the cognitive state of the driver is one of its major causes. We propose a novel approach for a driver monitoring system (DMS) to detect the cognitive state of the driver in real time using object detection. For this, we have used data from 19 different drivers in diverse traffic conditions in Bengaluru with over 7 h of driving time. We have chosen the YOLOv5 algorithm for our classification model with three categories, namely focused, sleepy (eyes closed or yawning), and distracted (looking away from the road). A magnetometer integrated with this model effectively categorizes distracted head turns while discerning them from deliberate and desirable head movements; this improves the robustness of classification. Velocity is computed using an onboard GPS unit, and the readings are used to determine if the vehicle is stationary in which case the detections are ignored. Experiments conducted on 19 people showed that our system has an average accuracy of 96.90% for this three-class classification model.

Keywords Driver monitoring system · Magnetometer · Haversine formula

S. Sathish · S. Ashwin · S. Manish (✉) · N. S. Shukapuri · V. Talasila
Department of Electronics and Telecommunication Engineering, Ramaiah Institute of Technology, Bengaluru, India
e-mail: manishshashi26@gmail.com

S. Sathish
e-mail: srimansathish@gmail.com

S. Ashwin
e-mail: ashwins02102001@gmail.com

N. S. Shukapuri
e-mail: nishanth.shukapuri@gmail.com

M. S. Gowda · V. Talasila
Centre for Imaging Technologies, Ramaiah Institute of Technology, Bengaluru, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_22

1 Introduction

Road traffic accidents are a major cause of death and disability, with over 1.3 million fatalities [1], underscoring the urgent need for improved road safety measures. According to a report by the National Highway Traffic Safety Administration (NHTSA) in 2019, drowsy driving was a factor in an estimated 96,000 crashes, resulting in over 800 deaths and 52,000 injuries in the USA [2]. A study conducted by NHTSA in 2019 [3] showed that distracted driving was the cause of one-third of all accidents. Driver behavior plays a significant role in road safety, and driver monitoring systems (DMS) are emerging as a promising technology for improving driver safety with other ADAS systems as well. The system uses various sensors and algorithms to monitor the driver's behavior and detect signs of fatigue, distraction, or inattention. When the system detects such behavior, it can issue an alert to the driver, warning them to take corrective action or pull over if necessary.

The focus of this work is to determine the cognitive state of drivers using object detection while ensuring that false estimates are significantly reduced. Specifically, generating a diverse dataset of driving data of multiple drivers in different traffic conditions, then training a machine learning model using the YOLOv5 algorithm. The novelty in our approach is the GPS and compass sensors being incorporated into the driver monitoring system (DMS) which enable false detections to be significantly reduced. The remaining sections of the paper are structured as follows. Section 2 provides a review of the existing literature on driver state monitoring. Section 3 includes details about the data collection process, the selection of participants or subjects, and the devices employed for data acquisition. Building upon this foundation, Sect. 4 introduces the methodology, techniques, and algorithms that are employed to monitor the state of drivers effectively. Sections 5 and 6 will delve into a discussion of the results obtained and the classification accuracies.

2 Literature Survey

The automotive industry today uses proprietary technologies for driver monitoring [4]. Attention Assist is a system developed by Mercedes-Benz that recognizes steering patterns of the driver at the beginning of the journey and then monitors the braking and acceleration to analyze the driver's concentration [5]. Toyota built an advanced driver assistance system (ADAS) that uses a camera affixed to the steering wheel which detects eye and head movement to determine if the driver is looking forward [6]. We have taken a similar approach, by mounting a camera on the driver-side sun visor to monitor the driver's face. Eyesight driver assist technology is a driver drowsiness detection system developed by Subaru [7]. It has a camera unobtrusively placed near the rear-view mirror that scans the road, monitors traffic movement, and warns if drivers sway outside the lane.

Francisco Vicente et al. [8] proposed an inexpensive vision-based system to detect eyes off the road (EOR). The system comprises three components: facial feature tracking, head pose estimation, and eye gaze estimation. This model achieved an overall accuracy of 90%. This system does not require any driver-dependent calibration or manual initialization.

Kashevnik et al. [9] proposed a system that utilizes smartphone sensors for detecting dangerous states for a driver in a vehicle. The smartphone camera is used to detect the driver's face. This information captured, along with other sensor data, is used to detect drowsiness, distraction, and aggressive driving that can lead to road accidents. Along with this, a cloud-based architecture is used to capture driver metrics and personalize the smartphone application for the driver.

A potential drawback with current literature is that false detections of a cognitive state may be high during certain driving conditions (such as taking turns). Here, the driver needs to constantly shift their focus in a large field of view, and existing algorithms may classify this as distracted driving. Thus, classification accuracy may be reduced. In this work, an elementary sensor fusion was used to significantly reduce such classification errors and thereby improve classification robustness.

YOLOv5 belongs to the you only look once (YOLO) [10] family of computer vision models. Specifically designed for object detection, YOLOv5 generates features from input images, which are subsequently processed through a prediction system. This system draws bounding boxes around objects and predicts their respective classes. YOLOv5 facilitates accurate detections at a high framerate compared to other more traditional object detection models and older versions of YOLO, which is imperative as it allows us to gauge the duration of drivers' blinks in units of fractions of a second to judge if they are fatigued.

For the default batch size of 16, YOLOv5 achieves 140 fps while YOLOv4 achieves 50 fps. A Tesla p100 GPU was used to obtain these results. Even with a sizable increase in fps, YOLOv5 maintains a mean average precision similar to the previous version. Lastly, the weights for a YOLOv5 model are 90 percent smaller than the weights of a YOLOv4 model. For the aforementioned reasons, YOLOv5 was chosen.

3 Experimental Setup with Dataset Description

19 drivers participated in this experiment; data was collected with a camera affixed to the driver-side sun visor as shown in Fig. 1. Video data was recorded at 60fps 1080p for training using GoPro Hero (2018)—testing was carried out in real time using Intel RealSense d435i, and the videos collected using the GoPro were passed through the model for testing. To ensure that the classification model is robust across data collected from different cameras, data from two different cameras was collected.

A brief description of the dataset is given below

- Driving Hours Recorded—7 h.

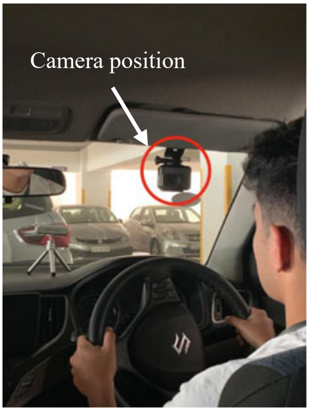


Fig. 1 Experimental arrangement indicating camera placement

Table 1 Definition of the three classes

Focused	When the driver pays attention to the road and does not look in any other direction
Sleepy	When they either yawn or close their eyes for more than 2 s
Distracted	When a driver's visual attention is diverted from the road, either by looking away or by their head being turned or tilted at an angle equal to or exceeding 45° away from the road

- Number of drivers—16 male and 3 female.
- Age of drivers—between 21 and 52.
- Traffic conditions—usual Bengaluru traffic and highway travel.
- Approximate distance traveled—185 km.
- Size of total data collected in GB—3.08.
- Cameras used: GoPro Hero (2018), Intel RealSense d435i

Table 1 gives the definition for each of the three classes, and Fig. 2 shows samples of collected images for each class (focused, sleepy, and distracted).

4 Methodology

The pipeline of our approach is shown in Fig. 3. The first step is generating a diverse dataset of multiple people driving in different traffic conditions. Data was collected at different times of the day from early mornings to late evenings. We chose this particular position (see Fig. 1) because it gives a direct view of the driver's face which allows us to easily track their head position and eye movement to accurately gauge the attentiveness of the driver.

The data is recorded in video format at 60 fps, 1080p resolution using GoPro Hero (2018). Frames were extracted from these videos and annotated using VoTT [8] as

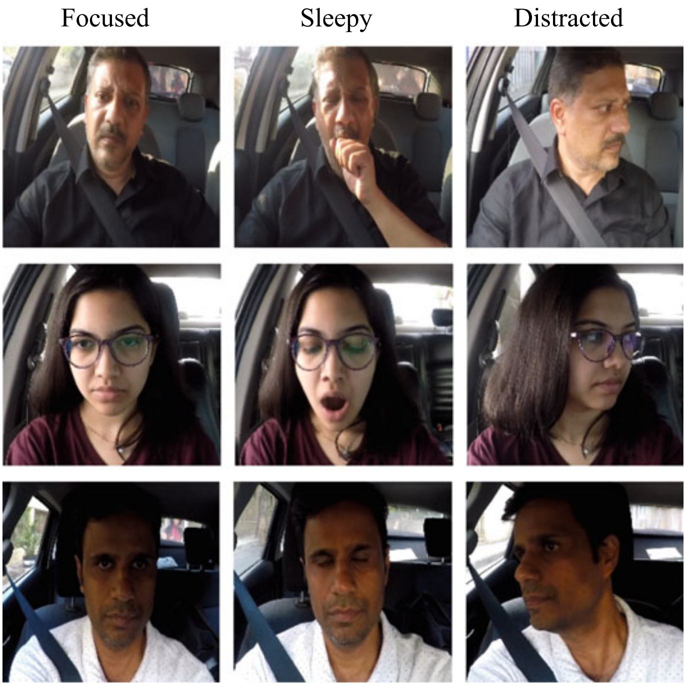


Fig. 2 Experimental arrangement showing camera FOV and showing examples of the three different classes for classification

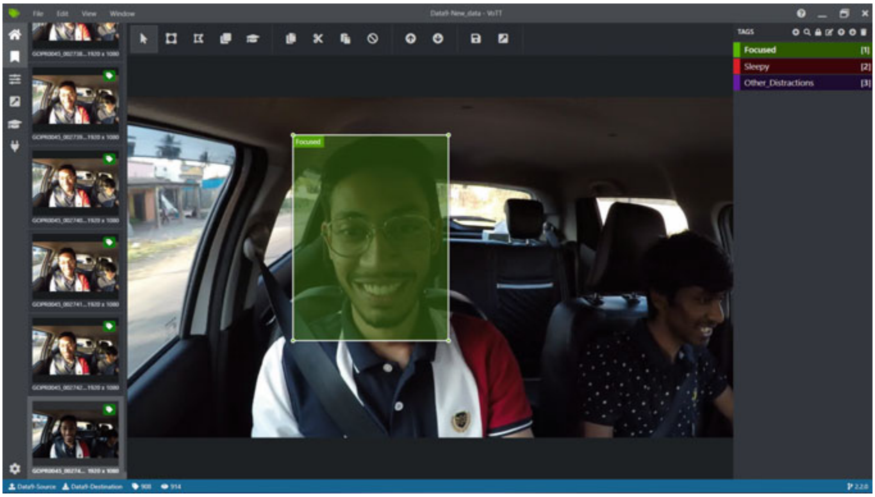

File
Edit
View
Window
Data9 New_data - VoTT
TAGS
Focused [1]
Sleepy [2]
Other_Distractions [3]
Focused
GOPR0045_002738...1920 x 1080
GOPR0045_002739...1920 x 1080
GOPR0045_002740...1920 x 1080
GOPR0045_002741...1920 x 1080
GOPR0045_002742...1920 x 1080
GOPR0045_00274... 1920 x 1080
Data9-Source
Data9-Destination
908
914
2.2.0

Fig. 3 Snapshot of the VoTT annotation process

shown in Fig. 3. The annotated data is then exported as JSON files and then passed through Roboflow along with their respective frames. Roboflow returns the data in a format compatible with YOLOv5's darknet architecture with 90 percent allocated as training and 10% as validation data. Once the model is trained, we test its accuracy in real-world driving conditions.

Figure 4 shown below describes the process of the driver monitoring system beginning from capturing the drivers face while simultaneously initializing the different modules including magnetometer and GPS. The model is set up or started only if the vehicle is moving at a speed greater than 5 km/h. As soon as the velocity drops below the threshold, the detections are disregarded.

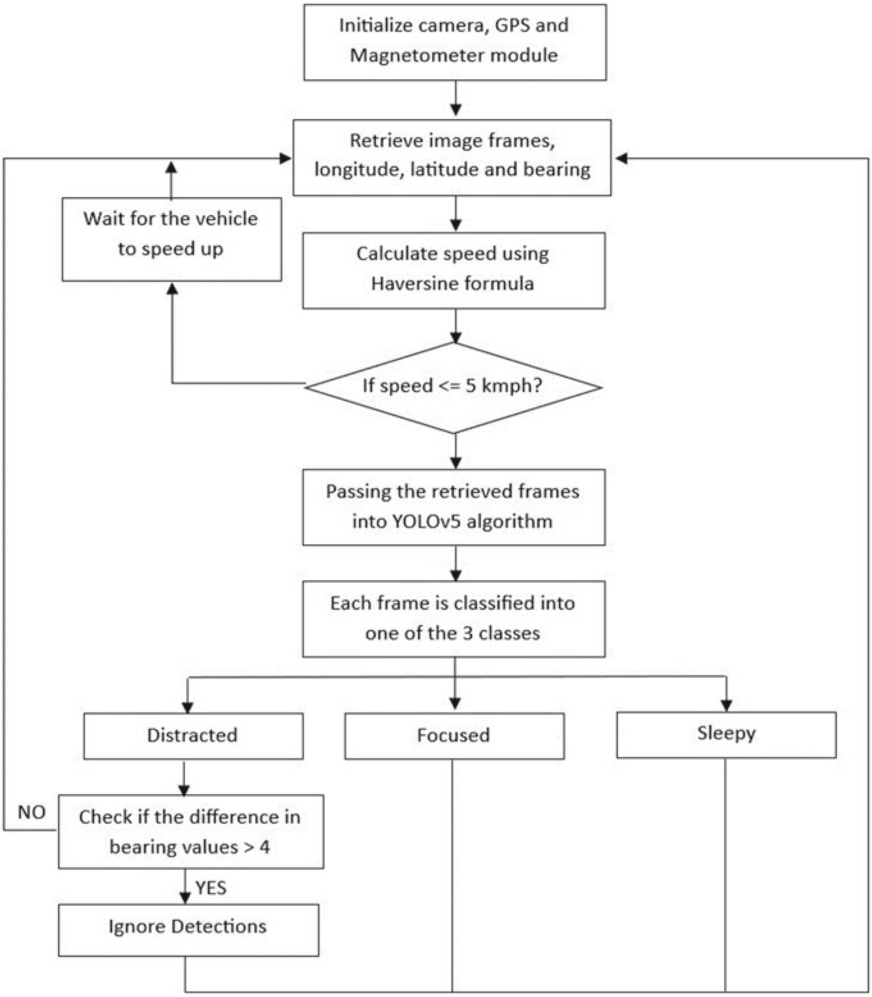

Initialize camera, GPS and Magnetometer module
Retrieve image frames, longitude, latitude and bearing
Wait for the vehicle to speed up
Calculate speed using Haversine formula
If speed <= 5 kmph?
Passing the retrieved frames into YOLOv5 algorithm
Each frame is classified into one of the 3 classes
Distracted
Focused
Sleepy
NO
Check if the difference in bearing values > 4
YES
Ignore Detections

Fig. 4 Flowchart of the model

4.1 Driver Distraction Analysis with Magnetometer-Based Turn Detection

To effectively analyze driver distractions and minimize false positives, it is crucial to detect when the car is turning to determine if the driver is focusing on the road or distracted by other factors. To achieve this, we extract bearing values from the real-time magnetometer data, which measures the magnetic field strength along the x, y, and z axes around the sensor.

Initially, we collect the three magnetic field values along the three axes (μx, μy, and μz) from the magnetometer. By applying the arctangent function to the μx and μy values, we convert them into radians, resulting in a single bearing value. The following formula illustrates this process:

$$\text{bearing} = a \tan \tan\left(\frac{\mu_y}{\mu_x}\right) \tag{1}$$

The next step involves storing two consecutive bearing values and calculating their difference. The magnetometer sensor returns eight magnetic field values every second. The threshold of the bearing value difference for turning was set to 3°. This value was chosen after testing the sensor multiple times in real time when the car is taking a turn. If the difference exceeds a threshold of 3°, it indicates that the car is undergoing a turn.

During a turning period, the system can then disregard or filter out any detections that occur, enabling a more precise evaluation of driver distractions. This filtering mechanism helps eliminate false-positive detections that might arise due to routine turning maneuvers.

Figure 5 indicates how the change in bearing values (delta) varies when the vehicle is moving in a straight path. As can be seen, the bearing values are fairly close to zero, and in other data samples, it was observed that the bearing may go close to 2°. Hence, the threshold for detecting a turn has been fixed at 3°.

Figure 6 shows the change in bearing values with time when a car is taking a turn. Figure 6 is more spread out compared to Fig. 5 indicating the delta value starts to increase more during a turn than a straight path. As can be seen, the change in the bearing values is significantly higher than 3°, and this has been observed across all datasets.

4.2 Speed Calculation Using Haversine Formula

It is expected that drivers may naturally look around their surroundings or interact with the infotainment system when the vehicle is at a standstill or moving at a slow pace without compromising road safety. Measuring the vehicle's speed to check if its stationary or not, we have extracted latitude and longitude values from the

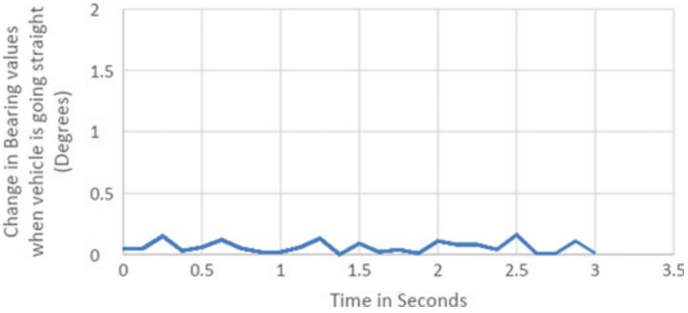


Fig. 5 Change in bearing values with respect to time when the vehicle is following a straight path (data shown above is one sample from a driving experiment, similar results were observed for all samples in which the vehicle was moving on a straight road)

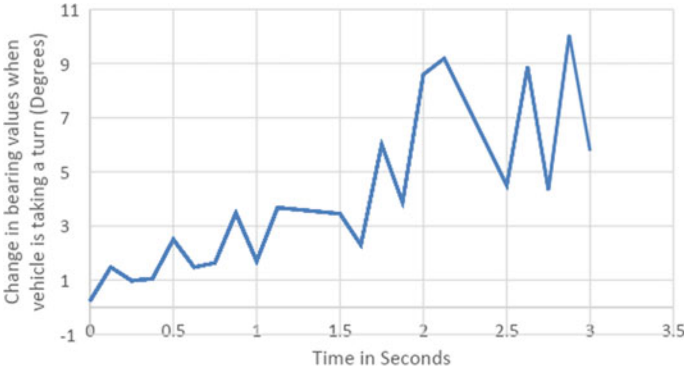


Fig. 6 Change in bearing values with respect to time when the vehicle is taking a turn (data shown above is one sample from a driving experiment, similar results were observed for all samples in which the vehicle was taking a turn)

GPS sensor of the phone to calculate the distance traveled by the car within two consecutive seconds. By using the Haversine formula [11], the distance between two points along the surface area of a sphere (earth in our case) is calculated using the latitude and longitude of the object. The Haversine formula is given below:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos lat1 * \cos \mathrm{lat2} * \sin^2\left(\frac{\Delta \mathrm{lon}}{2}\right) \tag{2}$$

$$c = 2 * a \tan \tan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \tag{3}$$

$$d = R * c \tag{4}$$

where Δlat is the difference between the two latitudes at two consecutive seconds and Δlon is the difference between two longitudes for the same time stamps. 'R' denotes the radius of the earth and 'd' is the distance traveled by the vehicle in that time period. These calculations might not be accurate for very long distances or in regions near the poles because they assume a simplified spherical model of the Earth.

The threshold to judge if the vehicle is moving or stationary is set to 5 km/h. Thus, only when the value of 'd' is equal or lesser than 5, the car is determined to be stationary and the model can skip these detections for that duration and wait until the vehicle starts moving again.

5 Experiments and Results

After collecting data of about 7 h of driving time for 19 different people and recorded at 60 fps, we obtained approximately 1.5 million frames. We observed that consecutive frames of a high fps video had no significant differences; hence, we opted to consider only one out of every ten frames for our model, bringing it down to 15 fps. Around 82,000 frames were used to train the model, and around 7000 frames were used for validation. The code processes each frame at 35 ms.

Figure 7 illustrates the fluctuation in the number of frames in which detections were observed the vehicle's velocity increases in 10 km/h increments. The red hollow square markers represent the detections recorded when the vehicle's speed was below 5 km/h, which are disregarded. The blue solid diamond markers indicate the frames that are detected when the vehicle is moving at a satisfactory speed. An assumption was made that during a turn, a vehicle speed is typically less than or equal to 5 km/h; this was based on all the data collected. This is of course not true in general, and in future work, we aim to combine velocity data with bearing angle data to decide when to disregard frames (for classifying distraction).

5.1 Relation Between Bearing Angles and Driver Distraction via Detecting Vehicle Turns

Figure 8 illustrates the frequency of frames in which distractions were detected for various changes in compass values during a 30-min driving session. The data points marked in blue solid diamonds, which occur within the 3° threshold, represent the number of frames observed when the driver was actually distracted (i.e., driver was not taking a turn and thus should not have been distracted). The data points marked in red hollow squares, which occur beyond a 3° threshold, represent false detections made by the model, accounting for approximately one-third of the total distracted frames. Here, the driver was making an actual turn and naturally swivel their head across their field of view, and this is not considered as distracted driving. This issue

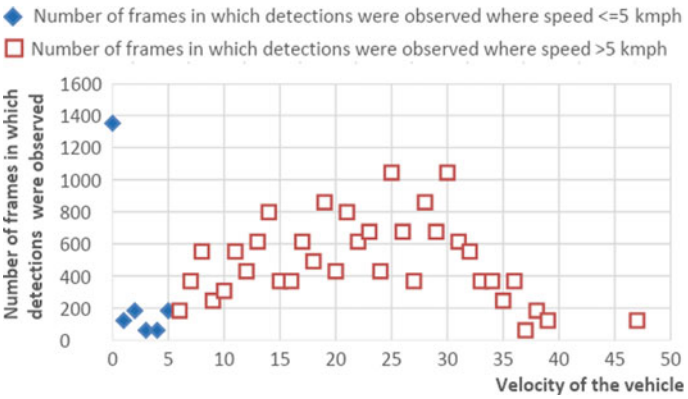


Fig. 7 Velocity of the vehicle recorded corresponding to the number of frames in which detections were observed at that instance

is mitigated by incorporating a magnetometer module into the system. From the data collected in this experiment, it was observed that the change in bearing angles is greater than 3° on average (based on a 8 Hz data rate magnetometer that was used in this experiment) during any turn. Thus, 3° was fixed as an upper threshold to detect a turn. Additional driving experiments and different road geometries may lead to slightly different threshold for the turn detection.

Figure 9 illustrates the correlation between bearing values (in degrees) and the number of frames recorded when the driver was distracted at those specific bearing

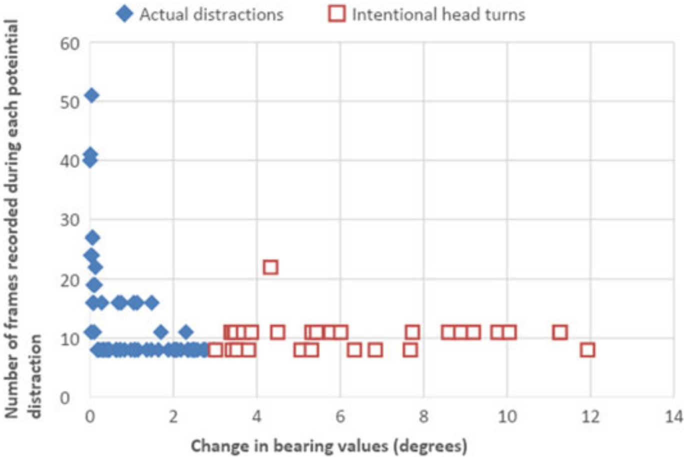


Fig. 8 The vertical and horizontal axes represent number of frames recorded during each potential distraction and change in bearing values, respectively. The solid blue diamonds and the hollow red squares differentiate between actual distractions and intentional head turns (observed when the driver intentionally looks both ways when taking a turn)

values. The red highlighted portion (Solid outline) in the graph represents instances where the vehicle maintains a constant direction while the driver is distracted, indicating actual distractions. Additionally, Fig. 9b visually displays the trajectory of the vehicle during such instances. The green highlighted portion (Dashed outline) signifies situations where the driver is taking a turn and their head is turning intentionally during the process, indicating intentional head turns. Figure 9c provides a corresponding trajectory representation for this scenario. The spikes observed in Fig. 9a indicate instances of actual distractions, while the concentrated points correspond to intentional head turns.

5.2 Results Observed in Varying Lighting Conditions

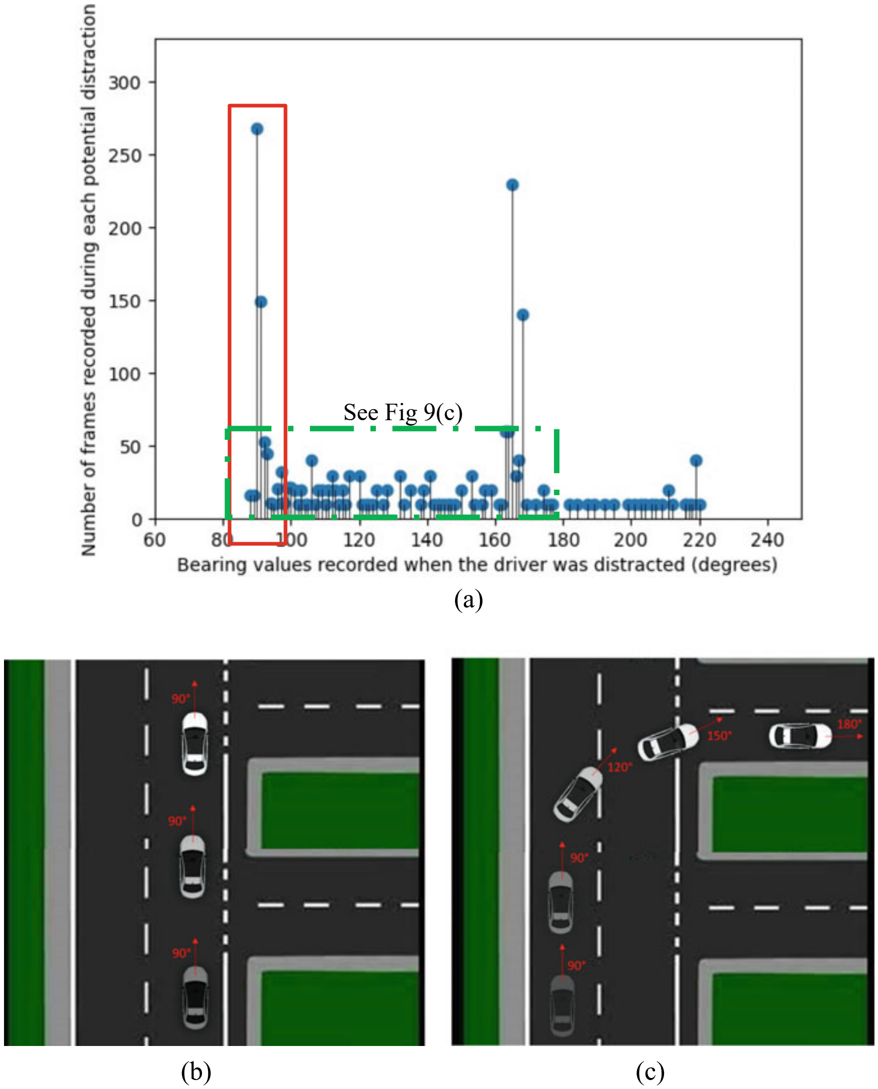
5.3 Classification Accuracies

The evaluation process for the model's accuracy involved recording data in multiple lighting conditions with varying luminescence. These recorded videos were then used to test the model, and the mean accuracy for each of the classes was calculated based on the model's performance.

During the testing phase, several scenarios were observed where the system could exhibit false-positive detections. First, when the driver is wearing sunglasses or shades, the camera will face difficulty in accurately identifying if the driver's eyes are open or closed; hence, it becomes challenging to detect whether the driver is sleepy or focused. Second, in night-time or low-light conditions, the camera may struggle to identify the driver's face due to insufficient illumination. As a result, it becomes challenging to detect the driver's face, leading to no detections. Third, when another person sitting beside the driver enters the camera's field of view, their face may be detected instead of the driver's. To mitigate this, we use the azimuth elevation and range to capture only the driver's face and hence obtain better results.

Figure 10 gives an insight into the effects of different lighting conditions on the ability of the model to detect the driver's face. The luminosity of the frames in the last row is very low, and hence, there are no detections. Figure 11. illustrates three different states of the driver collected at different luminosity values.

Three state-of-the-art methods, [12, 13], including MCNN, Gaze Detection, FastRCNN [12], are compared with our results. The Gaze Detection approach utilized a video processing framework, while the MCNN algorithm achieved an accuracy of 98% when trained on 10,300 images captured on smooth terrains. However, the accuracy dropped to 90% when the vehicle encountered uneven or rough roads due to vibrations. Our model has the capability to differentiate intentional head turns and actual distractions which the Gaze Detection approach is unable to do. The mean average precision (mAP) values for different models were compared with SSD achieving 72.39%, FastRCNN achieving 77.65%, and MobileNetV3 achieving 65.86% [13], and our model achieved an impressive 96.90%. Overall, our model



(a)

(b)

(c)

Fig. 9 **a** Number of frames recorded during each potential distraction was detected versus bearing values recorded at that instant. **b** Top view of a vehicle going on a straight path correlating to the highlighted part in red (solid outline) in the above graph (a) indicating actual distractions. **c** Top view of a vehicle taking a 90° turn correlating to the highlighted part in green (Dashed outline) in the above graph (a) indicating intentional head turns observed during a turn

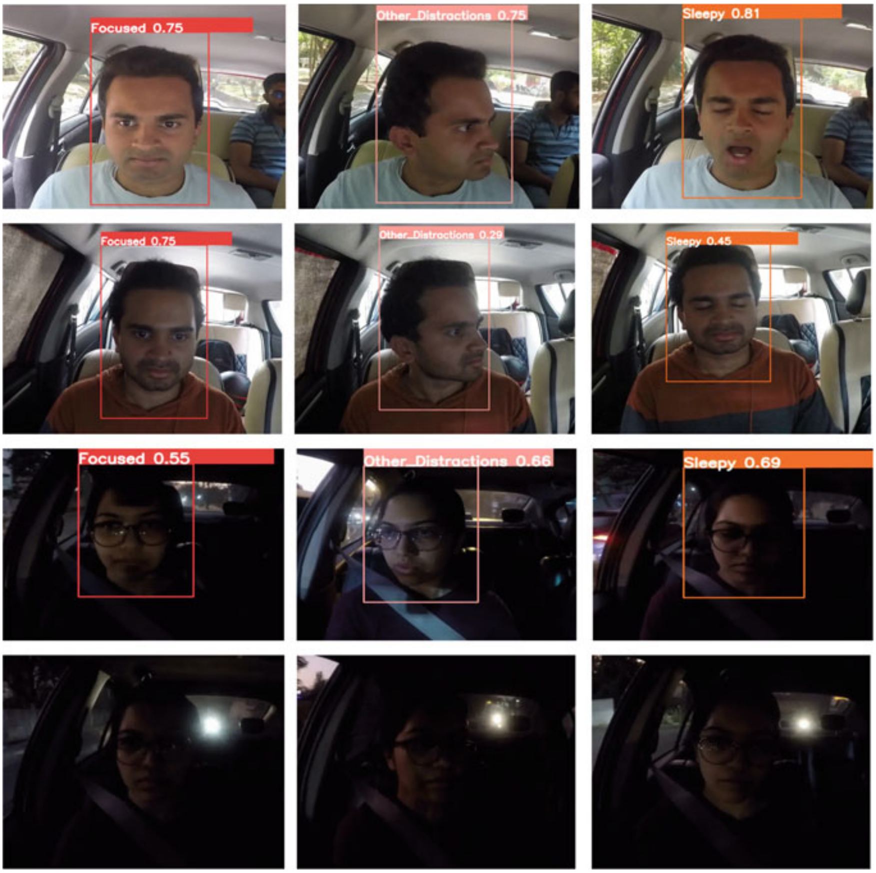


Fig. 10 Examples of testing run at varying lighting conditions with the probability of the detected class. Rows (top to bottom): high, medium, low and 0 lm in that order. Columns (left to right): Focused, distracted, and sleepy in that order

showcased superior performance in terms of mAP, while the MCNN algorithm's accuracy was influenced by road conditions and vehicle vibrations.

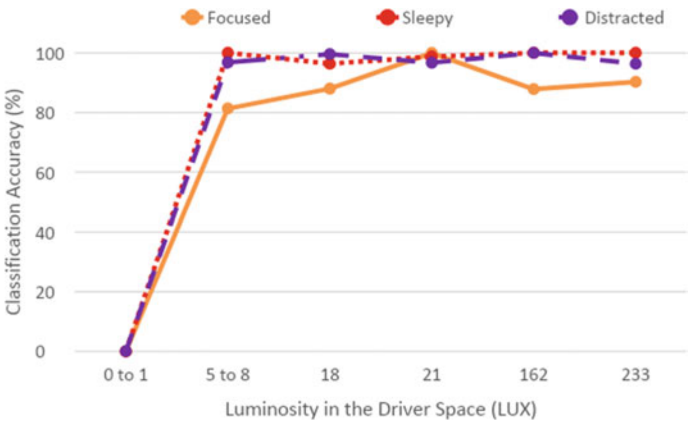


Fig. 11 Luminosity plot for the three cognitive states versus classification accuracy

6 Conclusion

The proposed driver behavior analysis system demonstrated good accuracy in detecting and categorizing driver states during the recorded video tests with minimal false detections (thus increasing the robustness of the classification accuracy). The use of the magnetometer and GPS has significantly improved both the accuracy and the robustness by being able to distinguish between turns (which create large movements of the head and eye) and going on a relatively straight trajectory. The MAP is around 96.9%. These results indicate the system's ability to differentiate between different driver behaviors. However, it is important to acknowledge certain limitations, such as the model's inability to detect driver eye movements when wearing sunglasses, difficulty in identifying faces in low-light conditions, and potential interference from other individuals in the camera's field of view. Addressing these limitations and refining the system's capabilities in various scenarios will be crucial for further improving its overall accuracy and reliability.

References

1. NHTSA Estimates for 2022 Show Roadway Fatalities Remain Flat After Two Years of Dramatic Increases. https://www.nhtsa.gov/press-releases/traffic-crash-death-estimates-2022. Last accessed 07 June 2023
2. NHTSA Drowsy Driving. https://www.nhtsa.gov/book/countermeasures/countermeasures-work/drowsy-driving. Last accessed 27 June 2023
3. NHTSA Reminds Drivers to Avoid Distractions, Launches Distracted Driving Campaign. https://www.nhtsa.gov/press-releases/nhtsa-reminds-drivers-avoid-distractions-launches-distracted-driving-campaign. last accessed 07 June 2023
4. Doudou M, Bouabdallah A, Berge-Cherfaoui V (2020) Driver drowsiness measurement technologies: current research, market solutions, and challenges. Int J ITS Res 18:297–319

5. Taylor M (2023) No doze: Mercedes E-class alerts drowsy drivers. https://www.autoweek.com/news/a2032716/no-doze-mercedes-e-class-alerts-drowsy-drivers/. Last accessed 10 Apr 2023
6. Toyota (2023) T-mate driving assistance—DRIVER MONITOR Camera. https://www.toyota-europe.com/brands-and-services/toyota/t-mate-driving-assistance. Last accessed 10 Apr 2023
7. EyeSight: EyeSight driver assist technology. Subaru. https://www.subaru.com/vehicle-info/eyesight.html. Last accessed 10 Apr 2023
8. Vicente F, Huang Z, Xiong X, De la Torre F, Zhang W, Levi D (2015) Driver gaze tracking and eyes off the road detection system. IEEE Trans Intell Transp Syst 16(4):2014–2027. https://doi.org/10.1109/TITS.2015.2396031
9. Lashkov KI, Ponomarev A, Teslya N, Gurtov A (2020) Cloud-based driver monitoring system using a smartphone. IEEE Sens J 20(12):6701–6715. https://doi.org/10.1109/JSEN.2020.2975382
10. Nelson J, Solawetz J (2023) YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS. https://blog.roboflow.com/yolov5-is-here. Last accessed 15 Feb 2023
11. SimonKettle: “Distance on a sphere: The Haversine Formula”. https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128. 29 Apr 2023
12. Poon Y-S, Lin C-C, Liu Y-H, Fan C-P (2022) YOLO-based deep learning design for In-cabin monitoring system with fisheye-lens camera. In: 2022 IEEE international conference on consumer electronics (ICCE), Las Vegas, NV, USA, pp 1–4. https://doi.org/10.1109/ICCE53296.2022.9730235
13. Guo Z, Wang G, Zhou M, Li G (2020) Monitoring and detection of driver fatigue from monocular cameras based on Yolo v5. In: 2022 6th CAA international conference on vehicular control and intelligence (CVCI), Nanjing, China, pp 1–6. https://doi.org/10.1109/CVCI56766.2022.9964752

A Hybrid Approach for Depression Detection Using Word Embedding, Naive Bayes and Bi-LSTM Models

Jyoti Singh, Ishan Mangotra, Minni Jain, and Amita Jain

Abstract Depression is a serious illness that negatively affects health and well-being. A large population suffers from depression and they do not want to talk about the mental illness. The stigma associated with mental illness may discourage people from getting treatment thus leading to serious issues, such as social isolation, discrimination and self-harm. The high use of social media enables people to express their feeling and thoughts easily. The objective of this research is the diagnosis of depression in a person from his/her social media behaviour. The novel approach of the proposed model is to ensemble the Gaussian Naive Bayes classifier and Bi-LSTM to find contextual semantics of the text using Part-of-Speech (POS) tagging and Word Embedding. The experimental result shows the proposed model outperforms the state-of–the-art method and shows an accuracy of 83% on the benchmark dataset.

Keywords Depression detection · Mental health · Social media · Word embedding · Ensemble method

J. Singh (✉) · I. Mangotra · A. Jain
Netaji Subhas University of Technology, Delhi, India
e-mail: jyotigaharwarsingh@gmail.com

I. Mangotra
e-mail: ishanmangotra25@gmail.com

A. Jain
e-mail: amita.jain@nsut.ac.in

M. Jain
Delhi Technological University, Delhi, India
e-mail: minnijain@dtu.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_23

1 Introduction

Depression is a very serious psychological disorder which affects a normal state of mind resulting in loss of interest in various kinds of work for the duration during which the patient is undiagnosed. It is different from regular mood swings in our everyday life. Depression alters each and every aspect of life, including relationships with near and dear ones [1]. An estimated 3.8% of the crowd suffering from depression includes 5.7% of people older than 60 years and 5% of adults (4% among men and 6% among women) [2]. More than 10% of pregnant women who just become mothers suffer from depression around the world [3]. Many serious illnesses such as heart disease, diabetes are developed due to depression. The most common cause of serious illness is depression [4, 5]. Millions of suicides occur in the world every year in which half of them are due to depressive disorder [5]. Among the age group of 15–29 years, the major cause of death is suicide. Long-term depression is caused by several reasons like harsh childhood, physical and mental harassment, alcohol addiction, work pressure, etc. [6]. Figure 1 depicts that seven out of ten teens says that anxiety and depression are major problem for their peers regardless of their gender, race and socio-economic lines. There are some other problems from which the youth suffer like bullying, drug addiction, poverty, etc., but anxiety and depression are the significant issues [7].

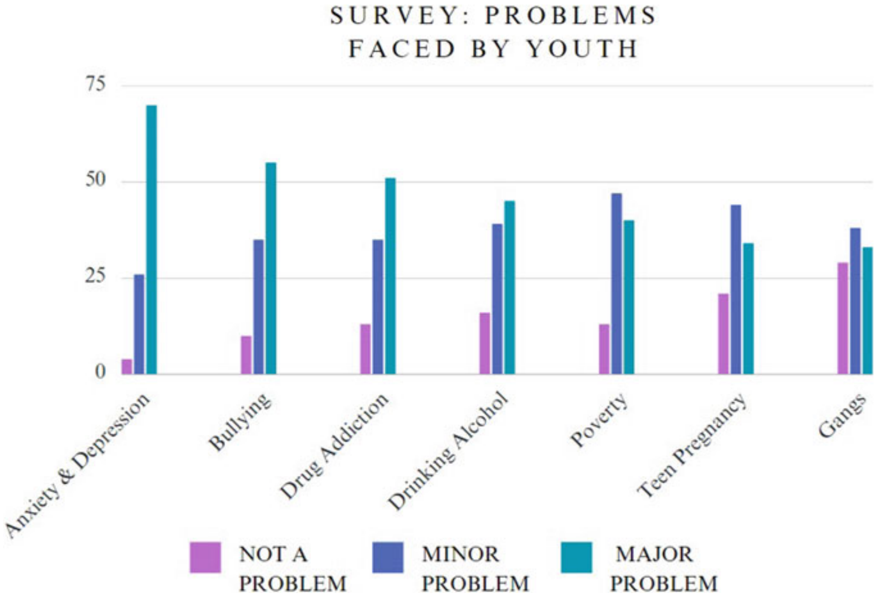


Fig. 1 Percentage of teens thinks each of the following is a problem among the peers in which anxiety and depression are the major problem

Twitter produces about 6000 tweets every second or 200 billion tweets on average per year. Twitter is a platform that provided open-source data [8]. The increased use of social media among youth gives flexibility to researchers to gather and analyse shared content on social platforms [9, 10]. Lots of crowds suppose depression a taboo, to express their sentiments to their friends and family and feel free to share it on social media.

NLP is an artificial intelligence technique for making computers effectively comprehend human language. Deep learning techniques like RNN and CNN with word embedding show significant accuracy in classifying text data. Various deep learning (DL) and machine learning (ML) methods are used to detect depressive disorder.

This study deployed an ensemble model of the Gaussian Naive Bayes classifier and the Bi-LSTM model. Word embedding and POS tagging to find the contextual meaning of the text have been used in this work. Twitter datasets are used to detect whether an individual suffers from depression or not.

There were the following sections in this study. The related research on depression detection is covered in Sect. 2. Section 3 provides the description of the methods and techniques used in this study. The datasets, different preprocessing methods, feature extraction and selection procedures are covered in Sect. 4. The model utilised in this investigation is discussed in Sect. 5. The method of experimentation and outcome are described in Sect. 6. Section 7 brings the study to a conclusion.

2 Related Work

Various machine learning algorithms were used by Priya et al. [11] to detect the mental disorder in which Naive Bayes classifier performs best. Choudhury et al. [12] collected data from 935 undergraduate students in Bangladesh to identify depression. After data preprocessing, they apply a machine learning algorithm to only 577 students. They obtain the highest accuracy with the Random Forest classifier. Hiraga et al. [13] detect mental disorders from Japanese blogs. They used a variety of machine learning algorithms like multinomial Logistic Regression, Naive Bayes and Linear SVMs. Uddin et al. [14] used Long Short-Term Memory (LSTM) and deep recurrent network to identify depression on online data available in Bangla Language. Wu et al. [15] utilised various machine learning algorithms like Decision Trees, Logistic Regression, SVM, XG Boost and Random Forest to predict job burnout. Fatima et al. [16] detect postpartum depression from social media texts by using different machine learning techniques like SVMs, Multilayer perceptron neural networks and Logistic Regression. Zulfiker et al. [17] detect depressive disorder using different machine learning classifiers. To achieve higher accuracy, various feature selection techniques were used in the model. The AdaBoost classifier using the SelectK-Best feature selection method stands out over all other methods. Kour et al. [18] proposed a depression prediction model from users' tweets using a hybrid of two deep learning methods, i.e. Convolution Neural Network (CNN) and Bi-directional Long Short-Term Memory (Bi-LSTM). Islam et al. [19] use Facebook posts and comments to predict whether the person is suffering from depression or not. They employ a variety of machine learning methods in which Decision Tree has performed well. Sau et al. [20] utilised various machine learning approaches to find the presence of anxiety and depression among seafarers. Here Cat-Boost classifier performs best. Ansari et al. [21] proposed two models; one is hybrid which combine different lexicons with Logistic Regression (LR) and other is ensemble model which is combination of deep learning model with hybrid Lexicon-Based LR model to detect depression. Here, ensemble model outperforms with an accuracy of 75%.

3 Proposed Methodology

This study proposed a model for depression detection which classifies a text as depressive or non-depressive. It comprises the following steps: (1) Data collection which is generated by online users. (2) Preprocessing of data that involves handling of missing and null values, conversion of texts into tokens, filtration of punctuation marks and stop words, POS tagging, etc. (3) Feature extraction is performed on the preprocessed data using word embedding. Word embedding converts text data into a real-valued numerical representation. (4) The extracted features are subjected to feature selection in order to choose pertinent characteristics and eliminate irrelevant features. (5) To build a model using the Gaussian Naive Bayes classifier and Bi-LSTM and finally ensemble these to detect depression. Figure 2 describes the methodology used in this study.

4 Dataset and Extraction & Selection of Features

4.1 Dataset Description

Twitter is an open-source online platform where one can access data easily. Many users share their thoughts and feelings without any hesitation. Generally, researchers follow two approaches for collecting online data, i.e. either using an existing dataset that is publicly shared by others or searching on social media websites like Facebook, Twitter, Reddit, etc., to collect the data.

In this framework, Twitter dataset is used.

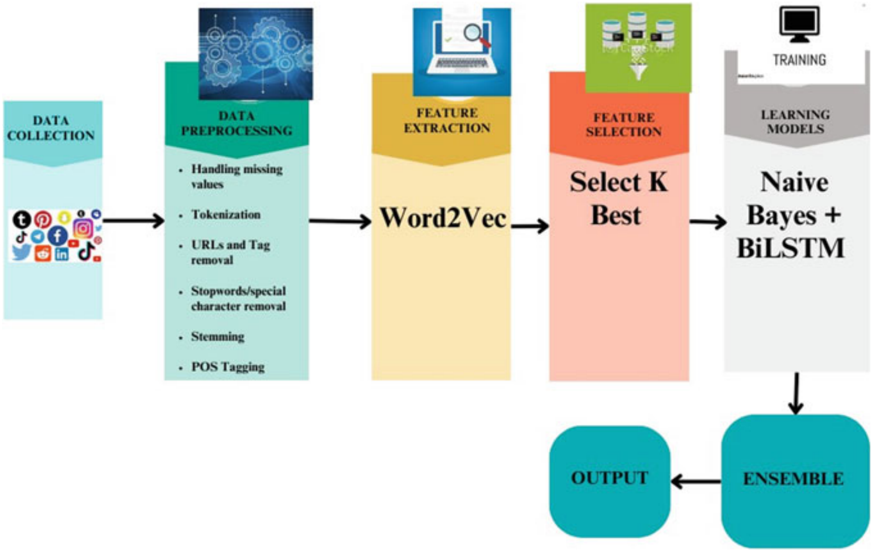


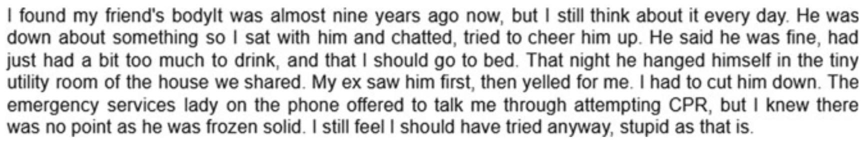
Fig. 2 Diagram of proposed method

4.2 Data Preprocessing

A vital stage in data mining jobs is data preprocessing [22]. The information contained in real-world data, which is gathered from numerous sources using diverse techniques, is partial, unstructured and erroneous. These kinds of facts produce ineffective outcomes. In this model, various preprocessing techniques have been employed. The first technique involves taking out the user handles (@username), hashtags, URLs, symbols, NAN-filled rows, duplicate rows, etc.

Stop words (like is, are, am, etc.) do not play any role in the context of sentences. The next step is to remove the stop words. The NLTK [23] package is used to eliminate the stop words. After this, stemming is performed. Converting a word to its base form is called stemming [24]. Once the data is cleaned, the cleaned data is tokenized using different tokenization functions.

Tokenizer is used to convert the string into tokens by breaking large textual data into small lines or words [25]. POS tagging will be done in the next stage. Each word is given its part of speech through a procedure known as POS tagging. POS tagging helps the model to understand the grammatical structure of the sentence and relationships between words such as Subject–Verb–object relationships can provide insights into the sentiments expressed in the text. For example: in the sentence, I feel extremely sad and hopeless, POS tag for ‘extremely’—Adverb, ‘sad’—Adjective indicates the intensity and negativity of a sentence.



I found my friend's bodyIt was almost nine years ago now, but I still think about it every day. He was down about something so I sat with him and chatted, tried to cheer him up. He said he was fine, had just had a bit too much to drink, and that I should go to bed. That night he hanged himself in the tiny utility room of the house we shared. My ex saw him first, then yelled for me. I had to cut him down. The emergency services lady on the phone offered to talk me through attempting CPR, but I knew there was no point as he was frozen solid. I still feel I should have tried anyway, stupid as that is.

Fig. 3 Snapshot of one example from the datasets which shows that the contextual meaning of the word 'hanged' is determined by the given surrounding words of the text

4.3 Feature Extraction

A new set of features that are appropriate for the model are extracted using a feature extraction technique. Different techniques, such as Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA), word embedding, etc., are employed for feature extraction. In this framework, word embedding is used for extracting features. Word embedding is a powerful NLP tool that performs various tasks like semantic analysis (SA), information retrieval (IR), dependency parsing (DP), question answering (QA), etc. [26]. It is used to extract the feature vector of a word by using semantic and syntactic meanings of a text. There are several word embedding methods such as Word2Vec, GloVe, fastText. Word2Vec method includes two approaches; one is Continuous-Bag-of-Words (C-BOW) and the second is skip-gram [27]. In this framework, C-BOW word embedding has used which predicts the target word from its surrounding contexts. For example, in Fig. 3 C-BOW predicts the target word 'hanged' and finds its contextual meaning based on its surrounding words.

4.4 Feature Selection

Feature selection is used to select the best features and remove irrelevant data. In this model, the SelectK-best feature selection technique using Scikit-learn (sklearn) is applied. SelectK-best operates on each feature independently and ranks them according to a specific statistical metric because it is based on univariate statistical tests. SelectK-best method chooses the feature which has the highest score value.

5 Proposed Models

5.1 Naive Bayes Classifier

Naive Bayes is a Bayesian network probabilistic classifier which is based on the Bayes theorem. Equation 1 represents the Bayes theorem. Given the class labels, it assumes that the characteristics are conditionally independent of one another.

$$P(x_1|x_2) = \frac{P(x_2|x_1)P(x_1)}{P(x_2)}, \tag{1}$$

where

- $P(x_1|x_2)$ is the posterior probability of event c given event x.
- $P(x_2|x_1)$ is the likelihood which is the probability of event x given event c.
- $P(x_1)$ is the prior probability of event c.
- $P(x_2)$ is the prior probability of event x.

5.2 Bi-directional Long Short-Term Memory (Bi-LSTM)

Although the recurrent neural network is effective at making time-series predictions, it has two drawbacks: (1) gradients that disappear and explode. (2) Future data are not taken into account [28]. To solve the vanishing gradient issue, LSTM is implemented.

Three gates—the input, output and forget gates—operate the LSTM. The quantity of information sent to the following layer is determined by the forget gate. The data addition and subtraction utilising different gates are shown in Eqs. 2, 3 and 4, where $\hat{I}$ stands for input, $\hat{O}$ for output and $\dot{F}$ for forget gate. Equations 5, 6 and 7 are used to determine the values of several cells, including candidate, output and cell state.

$$\hat{I}_{\dot{t}} = \sigma\left(\hat{W}_{\hat{I}}\left[\hat{H}_{\dot{t}-1,}X'_{\dot{t}}\right] + \mathcal{B}_{\hat{I}}\right), \tag{2}$$

$$\hat{O}_{\dot{t}} = \sigma\left(\hat{W}_{\hat{O}}\left[\hat{H}_{\dot{t}-1}, X'_{\dot{t}}\right] + \mathcal{B}_{\hat{O}}\right), \tag{3}$$

$$\dot{F}_{\dot{t}} = \sigma\left(\hat{W}_{\dot{F}}\left[\hat{H}_{\dot{t}-1}, X'_{\dot{t}}\right] + s_{\dot{F}}\right), \tag{4}$$

where $\widetilde{\hat{C}}_{\dot{t}}$: candidate for cell; σ: sigmoid function; $\hat{H}_{\dot{t}-1}$: output; $\mathcal{B}_{x\prime}$: bias; $w'_{x'}$: gate (x′) weight; $X'_{\dot{t}}$: input; $\hat{C}_{\dot{t}}$: cell memory;

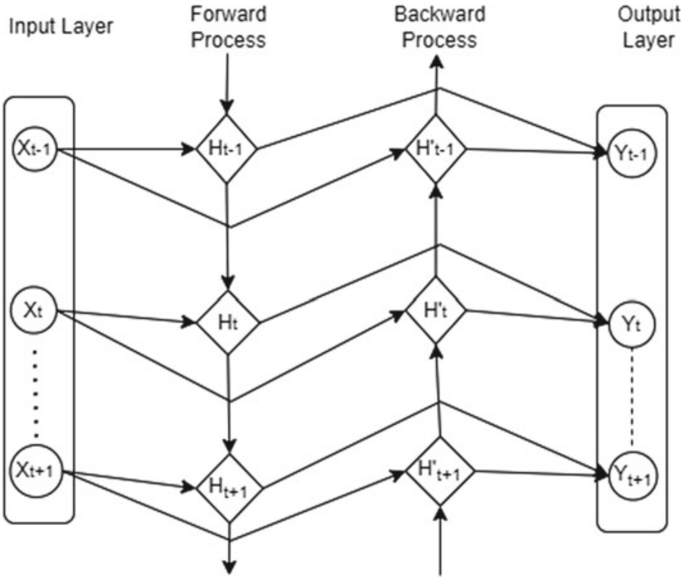


Fig. 4 Structure of Bi-LSTM model. The input data X_m is processed by both layers, i.e. forward and backward layers and finally hidden state H_m obtained and hidden states are fused to obtain output Y_m

$$\tilde{\hat{C}}_t = \tanh\left(\hat{W}_{\hat{c}}\left[\hat{H}_{t-1}, X'_t\right] + \mathcal{B}_{\hat{c}}\right), \tag{5}$$

$$\hat{C}_t = F'_t * \hat{C}_{t-1} + \hat{I}_t * \tilde{\hat{C}}_t, \tag{6}$$

$$\hat{H}_t = \hat{O}_t * \tanh\left(\hat{C}_t\right). \tag{7}$$

But LSTM does not resolve the second problem of RNN. As a result of which, Bi-LSTM is developed which solves both the problems of RNN. A Bi-LSTM is made up of two LSTM models, one of which accepts input forward and the other backward. Figure 4 [29] depicts the Bi-LSTM model's structure.

5.3 Ensemble

To make the final prediction, several models' predictions are combined using the ensemble. There are various ensembling techniques, e.g. Voting Ensembles, Bagging Ensemble, Boosting Ensemble and Stacking Ensemble. In this model, the Gradient Boosting Classifier as an ensemble is used.

6 Experiments and Results

6.1 Experimental Setup

Google collaboratory was chosen to implement the methodology which uses Python 3.10.12 version.

A deep neural network of Bi-LSTM was ensembled, using a Gradient Boosting Classifier, with a Gaussian Naive Bayes classifier for a better classification approach. TensorFlow's version 2.12.0[1] was used to build the Sequential Bi-LSTM model. To maintain the simplicity of the model, the number of hidden layers was confined to 3. Bi-LSTM layers were added with a recurrent dropout of 0.2 as it reduces the tendency of the model to memorise certain sequences which helps to develop the generalisation capacity of the model and make it robust to variations in the input data. In the end, two dense layers were added with 'tanh' and 'softmax' activation functions, respectively, to enhance the model's performance as well as maintain nonlinearity.

The loss was compiled using sparse_categorical_entropy, while the optimizer 'adam' was utilised and metrics of 'accuracy' was used to verify the training and validation accuracy.

For the second method, the Naive Bayes Classifier from the Scikit-learn library was trained on the dataset. The Scikit-learn library provides the SelectK-best feature selector which was used to obtain the selected features. By experimentation it was concluded that if the number of features was increased or decreased from 20, then the accuracy of the classifier began to degrade; therefore, 20 features were chosen from the vectorised dataset to train the Naive Bayes classifier. The hyperparameters of the model were experimented on, but in the end, the standard hyperparameters were chosen to train the classifier. When both the models were trained on the training dataset and tested on the validation dataset, the Gradient Boosting Classifier provided by the Scikit-learn library was put in use to carry out the ensembling procedure.

6.2 Result Analysis

The Scikit-learn library also provides the metrics of the 'Classification's Report' which comprises the following evaluations: F1~score, Recall and Precision. Table 1 shows the performance metrics of the proposed ensemble model and its comparison to Naïve Bayes and Bi-LSTM models. The ensemble approach outperforms and is achieved 83% accuracy with 83% of F1~score, 83% of Recall and 83% of Precision.

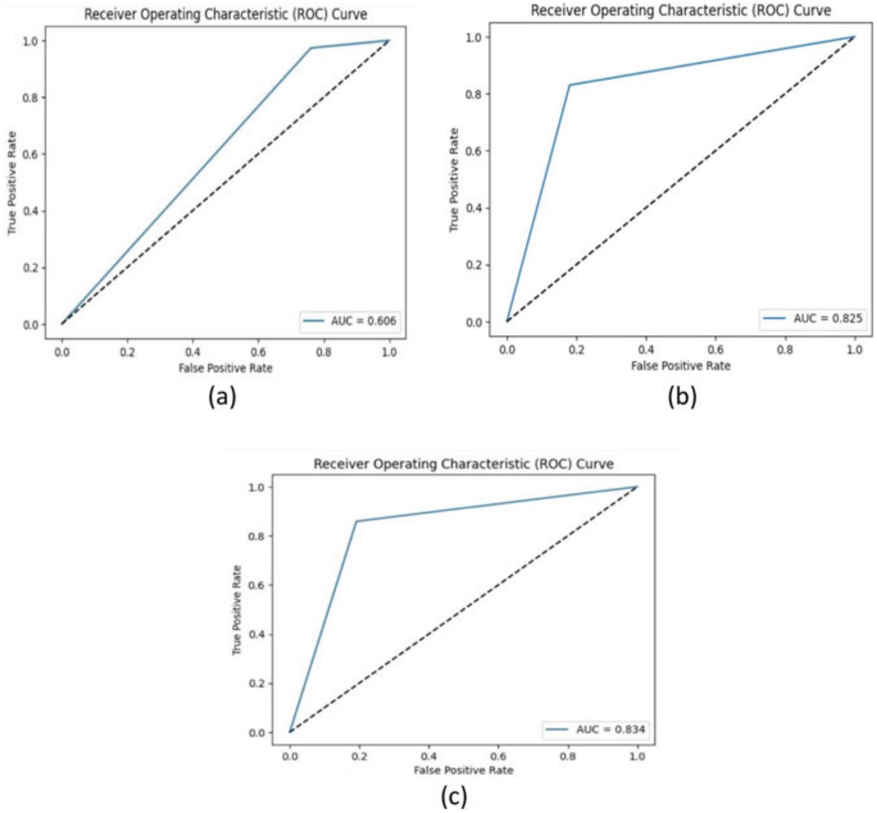
For visualising the results of the used approach, the Scikit-learn library was utilised to fetch the Receiver Operating Characteristic (ROC) Curve and the roc_auc_score. The x-label has denoted the 'false-positive rate' and the y-label has represented the 'true-positive rate'. This evaluation procedure of classification report and the ROC curve was applied to all the three models: Bi-LSTM, Naive Bayes and the Voting

Table 1 Performance metrics of the proposed ensemble model and comparison of the ensemble model with Naïve Bayes and Bi-LSTM models

Performance metrics	Naive Bayes Classifier	Bi-LSTM	Ensemble (Naïve Bayes + Bi-LSTM)
Accuracy	60	82	83
Precision	73	82	83
Recall	61	83	83
F1-score	54	83	83

Classifier. Figure 5 represents the ROC curve of the Naive Bayes, Bi-LSTM and Gradient Boosting Classifier.

Figure 5 depicts the ROC curves for the three models proposed in the current work. The first ROC curve is for Naive Bayes whose AUC comes out to be 60.6. The second curve shows results of the Bi-LSTM deep learning model with a good



(a)

(b)

(c)

Fig. 5 **a–c** represents the ROC curve of Naive Bayes classifier, Bi-LSTM and Gradient Boosting Classifier (ensemble), respectively

AUC of 82.5. Bi-LSTM performs much better than the traditional Naive Bayes ML approach which can be observed by its high AUC. The last curve is for the ensemble model of both Naive Bayes and Bi-LSTM. The ensemble model reaches an AUC of 83.4, showcasing its superiority to the other two models. AUC is chosen as a metric as it is quite useful in comparison of various classification models. Higher the AUC, better will be the model's ability to distinguish between the different classes.

7 Conclusion

The primary aim of this research is to identify depression in a person via social media texts. In this study, the Twitter dataset is used having two labels, i.e. '1' and '0'. '1' represents depressed data, while '0' represents non-depressed data. Different approaches like Gaussian Naive Bayes, Bi-LSTM and the ensemble of these two models are applied to the dataset to find out the depressive disorder. To improve the performance of the model, word embedding and POS tagging are used. The ensemble performed well with an accuracy of 83%.

This work can be studied further by using a pre-trained model like Bi-directional Encoder Representation from Transformers (BERTs) with word sense disambiguation.

Acknowledgements The authors would like to express their sincere gratitude to the IIT Mandi iHub & HCI Foundation under the Ministry of Science & Technology, Government of India, for their invaluable support to this research work.

References

1. "Depressive disorder (depression)." https://www.who.int/news-room/fact-sheets/detail/depression. Accessed 21 Aug 2023
2. "VizHub - GBD Results." https://vizhub.healthdata.org/gbd-results/. Accessed 21 Aug 2023
3. Woody CA, Ferrari AJ, Siskind DJ, Whiteford HA, Harris MG (2017) A systematic review and meta-regression of the prevalence and incidence of perinatal depression. J Affect Disord 219:86–92. https://doi.org/10.1016/J.JAD.2017.05.003
4. Whooley MA, Wong JM (2013) Depression and cardiovascular disorders 9:327–354. https://doi.org/10.1146/ANNUREV-CLINPSY-050212-185526
5. Otte C et al (2016) Major depressive disorder. Nat Rev Dis Primers 2:1:1–20. https://doi.org/10.1038/nrdp.2016.65
6. Oquendo MA, Ellis SP, Greenwald S, Malone KM, Weissman MM, Mann JJ (2001) Ethnic and sex differences in suicide rates relative to major depression in the United States. Am J Psychiatry 158(10):1652–1658. https://doi.org/10.1176/APPI.AJP.158.10.1652/ASSET/IMAGES/LARGE/J719T6.JPEG
7. "Most U.S. Teens See Anxiety, Depression as Major Problems | Pew Research Center." https://www.pewresearch.org/social-trends/2019/02/20/most-u-s-teens-see-anxiety-and-depression-as-a-major-problem-among-their-peers/. Accessed 21 Aug 2023

8. Arora P, Arora P (2019) Mining twitter data for depression detection. In: 2019 International conference on signal processing and communication, ICSC 2019, pp 186–189. https://doi.org/10.1109/ICSC45622.2019.8938353
9. Almeida H, Briand A, Meurs M-J (2023) Detecting early risk of depression from social media user-generated content. Accessed: 21 Aug 2023. [Online]. Available: http://psychpage.com/learning/library/assess/feelings.html
10. Biradar A, Totad SG (2019) Detecting depression in social media posts using machine learning. Commun Comput Inf Sci 1037:716–725. https://doi.org/10.1007/978-981-13-9187-3_64/COVER
11. Priya A, Garg S, Tigga NP (2020) Predicting anxiety, depression and stress in modern life using machine learning algorithms. Proc Comput Sci 167:1258–1267. https://doi.org/10.1016/J.PROCS.2020.03.442
12. Choudhury AA, Khan MRH, Nahim NZ, Tulon SR, Islam S, Chakrabarty A (2019) Predicting depression in Bangladeshi undergraduates using machine learning. In: Proceedings of 2019 IEEE Region 10 Symposium, TENSYMP 2019, pp 789–794. https://doi.org/10.1109/TENSYMP46218.2019.8971369
13. Hiraga M Predicting depression for Japanese blog text 107–113. https://doi.org/10.18653/v1/P17-3018
14. Uddin AH, Bapery D, Arif ASM (2019) Depression analysis from social media data in Bangla language using long short term memory (LSTM) recurrent neural network technique. In: 5th international conference on computer, communication, chemical, materials and electronic engineering, IC4ME2 2019. https://doi.org/10.1109/IC4ME247184.2019.9036528
15. Wu Y et al (2016) Google's neural machine translation system: bridging the gap between human and machine translation. Accessed: 21 Aug 2023. [Online]. Available: https://arxiv.org/abs/1609.08144v2
16. Fatima I, Abbasi BUD, Khan S, Al-Saeed M, Ahmad HF, Mumtaz R (2019) Prediction of postpartum depression using machine learning techniques from social media text. Expert Syst 36(4):e12409. https://doi.org/10.1111/EXSY.12409
17. Zulfiker MS, Kabir N, Biswas AA, Nazneen T, Uddin MS (2021) An in-depth analysis of machine learning approaches to predict depression. Current Res Behav Sci 2. https://doi.org/10.1016/j.crbeha.2021.100044
18. Kour H, Gupta MK (2022) An hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM. Multimed Tools Appl 81(17):23649–23685. https://doi.org/10.1007/S11042-022-12648-Y/FIGURES/20
19. Islam MR, Kabir MA, Ahmed A, Kamal ARM, Wang H, Ulhaq A (2018) Depression detection from social network data using machine learning techniques. Health Inf Sci Syst 6(1):1–12. https://doi.org/10.1007/S13755-018-0046-0/METRICS
20. Sau A, Bhakta I (2019) Screening of anxiety and depression among seafarers using machine learning technology. Inform Med Unlocked 16:100228. https://doi.org/10.1016/J.IMU.2019.100228
21. Ansari L, Ji S, Chen Q, Cambria E (2023) Ensemble hybrid learning methods for automated depression detection. IEEE Trans Comput Soc Syst 10(1):211–219. https://doi.org/10.1109/TCSS.2022.3154442
22. Naseem U, Razzak I, Khushi M, Eklund PW, Kim J (2021) COVIDSenti: a large-scale benchmark twitter data set for COVID-19 sentiment analysis. IEEE Trans Comput Soc Syst 8(4):976–988. https://doi.org/10.1109/TCSS.2021.3051189
23. Bird S, Loper E (2023) NLTK: the natural language toolkit. Accessed: 21 Aug 2023. [Online]. Available: www.python.org
24. Alshaer HN, Otair MA, Abualigah L, Alshinwan M, Khasawneh AM (2021) Feature selection method using improved CHI square on Arabic text classifiers: analysis and application. Multimed Tools Appl 80(7):10373–10390. https://doi.org/10.1007/S11042-020-10074-6/METRICS
25. Shen G et al Depression detection via harvesting social media: a multimodal dictionary learning solution. IJCAI Int Joint Conf Artif Intell 0:3838–3844.https://doi.org/10.24963/IJCAI.2017/536

26. Wang B, Wang A, Chen F, Wang Y, Kuo CCJ (2019) Evaluating word embedding models: methods and experimental results. APSIPA Trans Signal Inf Process 8. https://doi.org/10.1017/ATSIP.2019.12
27. Mikolov T, Chen K, Corrado G, Dean J (2023) Efficient estimation of word representations in vector space. Accessed: 21 Aug 2023. [Online]. Available: http://ronan.collobert.com/senna/
28. Kosana V, Teeparthi K, Madasthu S (2022) Hybrid convolutional Bi-LSTM autoencoder framework for short-term wind speed prediction. Neural Comput Appl 34(15):12653–12662. https://doi.org/10.1007/S00521-022-07125-4/METRICS
29. Wang P, Qian Y, Soong FK, He L, Zhao H (2023) A unified tagging solution: bidirectional LSTM recurrent neural network with word embedding," Nov. 2015, Accessed: 21 Aug 2023. [Online]. Available: http://arxiv.org/abs/1511.00215

A Deep Learning Approach to Computer-Aided Screening and Early Diagnosis of Middle Ear Disease

Ankit Kumar Singh, Ajay Singh Raghuvanshi, Anmol Gupta, and Harsh Dewangan

Abstract This article introduces a deep learning approach to computer-aided screening and early diagnosis of middle ear diseases such as earwax, otitis externa, tympanosclerosis, and ear ventilation tubes. The timely detection of middle ear conditions is crucial for effective treatment and prevention of complications. The proposed system utilizes a deep neural network trained on a large dataset of middle ear images obtained through advanced diagnostic imaging techniques. The system automatically analyzes these images by leveraging deep learning to provide accurate and efficient screening and diagnostic support. The proposed system aims to assist healthcare professionals in accurate and efficient early diagnosis and screening of critical conditions, leading to improved patient outcomes and optimized treatment plans. The proposed model presents a deep learning 2D-CNN model for binary and multi-class classification of ear diseases in medical healthcare. The results demonstrate its effectiveness and superiority compared with the traditional machine learning approaches.

Keywords 2D-convolutional neural networks · Tympanic membrane · Otitis externa · Image classifications · Otoscopic images

1 Introduction

Ear and mastoid disease is common and can easily be treated with early medical care. However, if one does not get prompt detection and the right care, it could have consequences like hearing loss. The initial stage in evaluating ear and mastoid disorders in a clinic involves a physical examination using traditional otoscopy and asking about

A. K. Singh (✉) · A. S. Raghuvanshi · A. Gupta · H. Dewangan
Department of Electronics and Communication, NIT Raipur, Raipur, India
e-mail: aksingh.phd2022.etc@nitrr.ac.in

A. S. Raghuvanshi
e-mail: asraghuvanshi.etc@nitrr.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_24

a patient's history. However, non-otolaryngologists employing otoscopy for diagnosis are prone to making mistakes. As per the research till now, there are machine learning systems for automated diagnosis of ear disease using otoscopic images, which divided the tympanic membrane into five groups based on the presence of cerumen impaction, acute otitis media, otitis media with effusion, otitis media with perforation, and normal eardrum. This work uses deep learning for otoscopy images of the eardrum and the external auditory canal (EAC) to provide a trustworthy diagnosis of acute otitis media [1], chronic suppurative otitis media, earwax, otitis externa [2], tympanosclerosis, and ear ventilation tube. For clinical use, most ear ailments that can be identified by otoendoscopy in clinics fall under these categories. For this, we present an ensemble classifier established on the top of deep neural networks as determined by ear image analysis. Ear diseases significantly impact individuals' quality of life and well-being, affecting their hearing ability and overall health. It's been a matter of concern to have an accurate and timely diagnosis of ear diseases for appropriate medical intervention and treatment planning. In recent years, deep learning has been introduced in various healthcare sectors. Most of these studies utilize the well-known supervised deep learning concept of 2D-convolutional neural network (CNN) with transfer learning, which shows promising results in various medical applications [3], including image classifications. This research paper investigates the application of transfer learning for classifying ear diseases from medical images, aiming to improve the efficiency and accuracy of diagnosis in otolaryngology.

Transfer learning is reusing and fine-tuning public CNN models already trained on raw images for a particular application. Transfer learning leverages the pre-trained models on large datasets from a source domain to solve problems in a target domain with limited data. By utilizing the knowledge learned from convolutional neural networks, transfer learning enables the efficient training of deep neural networks on smaller datasets, thereby addressing the limitations of insufficient labeled data in medical imaging. A key element of transfer learning is pre-trained models. These models are typically trained on large-scale datasets, such as ImageNet or COCO [4], for tasks like image classification, object detection, and natural language processing. The pre-trained models learn high-level features and representations from the source domain data, capturing rich patterns and semantics. The introduction of endoscopy in otoneurology clinics and ear surgeries has revolutionized the examination and treatment of middle ear pathologies. The tympanic membrane (TM), a vital middle ear component, plays a crucial role in sound transmission and can be affected by various pathologies. Structural changes in the tympanic membrane due to infections like acute otitis media, otitis media with effusion (OME) [5], and chronic otitis media (COM), as well as perforations caused by trauma, can lead to compromised sound transmission and conductive hearing loss. Among the factors affecting sound transmission, the thickness of the tympanic membrane has been identified as a significant contributor. Otoscopic examination is an essential diagnostic tool that allows for the accurate and efficient observation of the external auditory canal (EAC) [6], TM, and middle ear conditions. It has been a valuable technique for evaluating and diagnosing various ear pathologies. Since the 1990s, endoscopy has been introduced in otoneurology clinics and used in simple ear surgeries. Compared with microscopy, endoscopy provides a broader and higher quality field of view, making it particularly advantageous in pediatric patients [7]. Its accessibility through the EAC simplifies the examination process. In most cases, the external auditory canal (EAC) was found between the ages of 8 and 18 years old; in addition to the cases, 21,235 matched controls were included.

The mortality in cases and controls were similar; 1472 cases (13.9%) and 2791 controls (13.1%) died during the diagnosis [8].

2 Methodology

2.1 Dataset Collection

Patient selection and data acquisition are crucial steps in researching ear diseases, especially when obtaining images for analysis. In this paper, patient selection and data acquisition were carried out at Dicle University Medical School, Turkey. The dataset acquired is 769 otoscopic, segmented, and labeled middle ear images, of which 650 are normal, 119 are AOM-infected images, and 63 are CSOM-infected images [21]. A systematic approach was followed to ensure appropriate patient selection. Patients with suspected ear ailments, including acute otitis media, chronic otitis media (COM), otitis externa, tympanosclerosis, and ear ventilation tubes were considered for inclusion in this study. The selection criteria included patients of different age groups, both pediatric and adult, to capture a diverse range of cases and provide a comprehensive dataset. A set of otoscopic tests and endoscopy were used to collect the images. The test is suitable for an external auditory canal (EAC) and tympanic membrane (TM). An endoscopy was performed using specialized equipment for broader visualization and documentation. The endoscope provided a broader field of view and higher quality imaging than traditional microscopy [19]. It allows for clear visualization of the tympanic membrane, middle ear, and associated pathologies and facilitates accurate diagnosis and classification of ear diseases.

Ethical considerations were ensured during the data acquisition, and patient confidentiality was maintained. Informed consent was obtained from the patients or their guardians before including them in the study. The data acquisition process adhered to the guidelines and protocols set by the institutional ethics committee to protect the rights and well-being of the patients. Overall, patient selection and data acquisition at AIIMS, Raipur followed a systematic approach encompassing a diverse group of patients with suspected ear diseases. High-quality images were acquired by employing otoscopic examination and endoscopy, enabling accurate classification of ear diseases for the research paper.

2.2 Data Classification and Labeling

To conduct research on ear diseases for the abovementioned paper, a crucial step involved labeling the acquired images with appropriate annotations. AIIMS Raipur's team of qualified experts, including otolaryngologists and audiologists, meticulously labeled the images using a standardized annotation protocol. The labeling process involved categorizing the images into classes based on the specific ear disease being investigated. The identified classes for labeling included acute otitis media (AOM), chronic suppurative otitis media (CSOM), earwax, otitis externa, tympanosclerosis, and ear ventilation tube [9]. Medical experts test each image to identify the presence of the specific ear disease. For example, in cases of AOM and CSOM, the characteristic signs such as fluid accumulation, inflammation, and perforations were identified and labeled. The labeled images serve as a valuable dataset for training and evaluating the deep learning model proposed in the paper. By accurately annotating the images with specific ear diseases, the dataset provides a foundation for developing a robust computer-aided screening and diagnosis system. As shown in Fig. 1, accurately labeling the images with AOM, CSOM, earwax, otitis externa, tympanosclerosis, and ear ventilation tube classifications enhances the research's validity and reliability. It enables the deep learning model [10] to learn and recognize patterns associated with each ear disease, facilitating accurate classification and diagnosis.

A convolutional neural network (CNN) model was trained using a diverse dataset comprising labeled images representing AOM, CSOM, earwax, otitis externa, tympanosclerosis, and ear ventilation tube. The dataset encompasses visual representations of these specific ear diseases as described in Fig. 2a, b. Training the CNN model on this dataset allowed it to classify and distinguish between different ear conditions accurately. This trained neural network is a powerful tool for automated

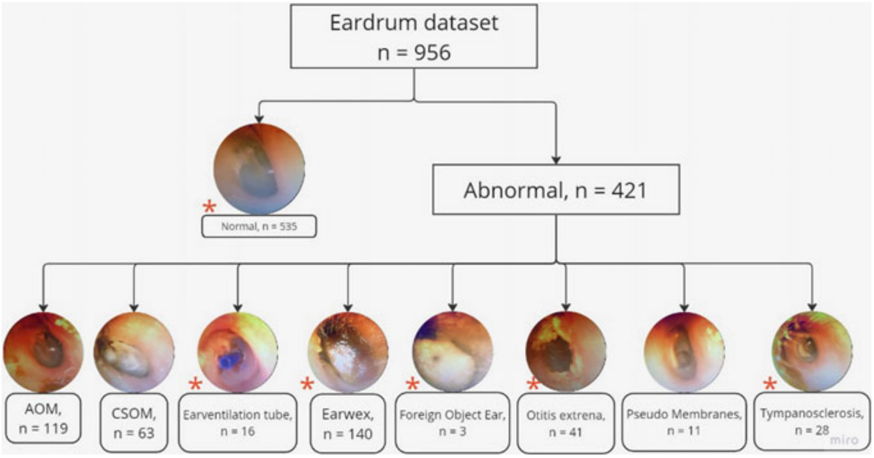


Fig. 1 Decision tree for labeling of otoendoscopy images and eight diagnostic classes that were used [21]

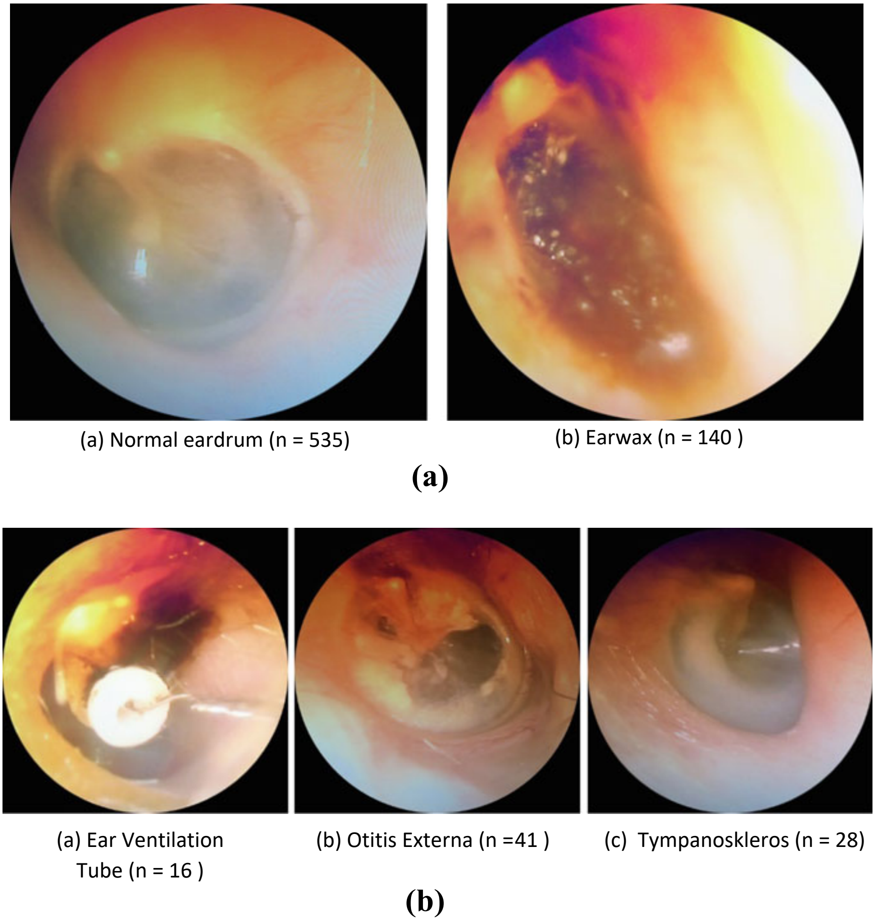

(a) Normal eardrum (n = 535)
(b) Earwax (n = 140)
(a)
(a) Ear Ventilation Tube (n = 16)
(b) Otitis Externa (n =41)
(c) Tympanoskleros (n = 28)
(b)

Fig. 2 **a** Binary classification [21]. **b** Multi classification [21]

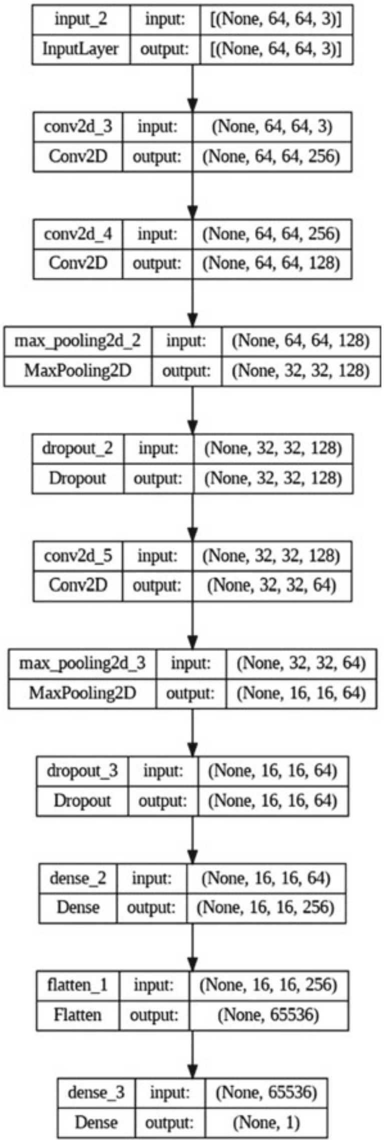
identification and diagnosis of ear diseases, contributing to improved healthcare practices and aiding in timely and effective interventions.

2.3 Training 2-D Convolution Neural Network Model

Ear diseases, such as acute otitis media (AOM), chronic suppurative otitis media (CSOM) [10], earwax impaction, otitis externa, tympanosclerosis, and ear ventilation tube placement can significantly impact auditory health. Traditional diagnostic methods for these conditions rely on visual examination by trained experts, which can be subjective, time-consuming, and prone to human error. In recent years, deep learning approaches, particularly convolutional neural networks (CNNs), have shown great promise in automating the diagnosis of medical conditions using image data. In this paper, we propose a deep learning CNN model (as discussed in Fig. 3) for the automated diagnosis of ear diseases. One common preprocessing step is resizing the images to a common shape, such as 64 × 64 pixels, which facilitates efficient processing and reduces computational requirements. Additionally, pixel values are normalized to a range of 0–1 to mitigate the influence of varying intensity levels [11]. This normalization step ensures that all images are on a similar scale, allowing the model to learn effectively from the data. Moreover, the dataset is further split into training and validation sets to prevent bias during training. Convolutional layers apply convolution operations to the input images, capturing relevant features such as edges, textures, and patterns. These layers leverage filter kernels to convolve across the input, extracting low-level and high-level features.

Data augmentation techniques are applied during training to enhance the model's performance and prevent overfitting. Data augmentation involves creating additional training data by applying various transformations to the existing images. The Image Data Generator class from the Keras library is employed for this purpose. Augmentation techniques include shear range, zoom range, and horizontal flip. Applying these transformations expands the training dataset, increasing its diversity and allowing the model to learn more robust and generalized features. This data augmentation process [12] helps to overcome limited training data and improves the model's ability to handle variations and noise in real-world ear images. Once the model architecture and data preprocessing steps are defined, the model is trained using the augmented dataset. During training, the model iterates over the dataset for several epochs. Each epoch represents a complete pass through the entire dataset, during which the model updates its parameters based on the optimization algorithm. The Adam optimizer, a popular optimization algorithm, is used for training deep learning models due to its adaptive learning rate and efficient convergence properties. The model is optimized by minimizing the categorical cross-entropy [13] loss function, which measures the dissimilarity between predicted and true class probabilities. The model learns to accurately classify ear disease images through the iterative training process, adjusting its internal parameters to minimize the training loss and improve its predictive capabilities. After training, the model's performance is evaluated using a separate validation set not used during training. The validation set provides an unbiased assessment of the model's generalization ability to unseen data. Accuracy and loss metrics are calculated to gauge the model's diagnostic accuracy and convergence. A high validation accuracy and low validation loss indicate that the model has learned to classify ear disease images accurately and performs well on unseen data. Evaluating the model's performance on the validation set helps to identify potential issues such as underfitting or overfitting.

The test findings highlight the model's performance inappropriately categorizing distinct circumstances about ear disorders. A proper evaluation of confusion matrix through its validation testing. These metrics comprehensively assess the model's diagnostic capabilities and highlight its proficiency in distinguishing between different ear diseases. For example, high precision indicates a low false



input_2	InputLayer	input:	[(None, 64, 64, 3)]
		output:	[(None, 64, 64, 3)]
conv2d_3	Conv2D	input:	(None, 64, 64, 3)
		output:	(None, 64, 64, 256)
conv2d_4	Conv2D	input:	(None, 64, 64, 256)
		output:	(None, 64, 64, 128)
max_pooling2d_2	MaxPooling2D	input:	(None, 64, 64, 128)
		output:	(None, 32, 32, 128)
dropout_2	Dropout	input:	(None, 32, 32, 128)
		output:	(None, 32, 32, 128)
conv2d_5	Conv2D	input:	(None, 32, 32, 128)
		output:	(None, 32, 32, 64)
max_pooling2d_3	MaxPooling2D	input:	(None, 32, 32, 64)
		output:	(None, 16, 16, 64)
dropout_3	Dropout	input:	(None, 16, 16, 64)
		output:	(None, 16, 16, 64)
dense_2	Dense	input:	(None, 16, 16, 64)
		output:	(None, 16, 16, 256)
flatten_1	Flatten	input:	(None, 16, 16, 256)
		output:	(None, 65536)
dense_3	Dense	input:	(None, 65536)
		output:	(None, 1)

Fig. 3 Convolution 2-D layer CNN model

positive rate, while high recall indicates a low false negative rate. By analyzing these metrics across different classes, we have identified its strength and limitation and their potential to revolutionize the healthcare sector.

2.4 Training ResNet101 Transfer Learning Model

Transfer learning is a deep learning approach that enables the information acquired from solving a certain problem to be transferred to another that is unrelated yet challenging. It involves leveraging pre-trained models, which have been trained on large-scale datasets, to extract useful features and patterns from images. These pre-trained models have learned to recognize various visual concepts and have captured rich data representations [14]. By transferring this knowledge to a new task, transfer learning enables effective learning even with limited labeled data. In transfer learning, the idea is to use a pre-trained model as a feature extractor by removing the final classification layer and connecting a new classifier specific to the new task. This approach avoids training a model from scratch, which can be computationally expensive and time-consuming. Instead, the pre-trained model is fine-tuned on the new dataset, adjusting the weights to adapt to the task. It involves several steps, which are mentioned in detail.

First, a pre-trained model is selected based on its performance and compatibility with its target task. Several models, like VGG, ResNet, and Inception [15] have been trained on large-scale image classification, such as ImageNet. Once a pre-trained model is chosen, the next step is to remove the top layers, including the final classification layer. These top layers are task-specific and must be replaced with new layers suited to the new problem. The lower layers, responsible for learning low-level features like edges and textures, are retained as they capture general useful image representations across various tasks. After removing the top layers, the remaining layers are frozen, meaning their weights are not updated during training. This ensures that the pre-trained weights, which have learned valuable features, are preserved. The most significant advantage of transfer learning is that it allows effective learning even with limited labeled data. Instead of starting the learning process from scratch, the pre-trained model already knows various visual concepts. This knowledge is transferred to the new task, enabling the model to make better predictions with fewer training samples. Transfer learning is particularly useful when collecting large labeled datasets which may be time-consuming or expensive. The objective is to minimize the loss function, typically cross-entropy loss, by adjusting the network layers' weights and biases. Training ResNet101 is computationally intensive and requires access to significant computational resources, such as graphics processing units (GPUs).

The ResNet101 model to detect ear disease starts by importing Keras applications. ResNet module: It is initialized with the input shape parameter corresponding to the desired image size and three channels (RGB). The weights argument is set to 'image Net,' which loads the pre-trained weights from the ImageNet dataset. The top parameter is false, excluding the fully connected layers at the network's top. To preserve the pre-trained weights [16], all layers in the ResNet101 model are set to non-trainable using a loop. This ensures that only the newly added layers will be trained. Custom layers are added to the ResNet101 model for classification purposes. The output of the ResNet101 model is flattened using the flatten layer, followed by a fully connected dense layer with 224 units and a ReLU activation function. Finally, a dense layer with the number of classes obtained from the len(folders) and a softmax activation function is added to generate class predictions. The complete model is created by specifying the input and output layers using the model class from Keras. The inputs parameter is set to res. Input represents the input layer of the ResNet101 model [17], and the outputs parameter is set to prediction, representing the output layer defined earlier. After that, the model is compiled by specifying the loss function as categorical cross-entropy, the optimizer as Adam, and the accuracy metric. This step prepares the model for training.

3 Result

The comparative analysis of eardrum disease image classification using convolutional neural networks (CNNs) revealed significant findings. Two models were evaluated for performance: a custom three-layer convolutional 2D model and the ResNet101 model. The results are summarized in Table 1 along with Figs. 4 and 5 for the binary classification task of distinguishing between normal and earwax; our custom model achieved an impressive validation accuracy of 93%, outperforming ResNet101 with an accuracy of 89%. The custom model demonstrated superior performance, requiring fewer parameters (249,537) than ResNet101 (65,137,698) for this task. The learning rate was set at 0.001, and both models were trained for 150 epochs.

In the multi-class classification task, where the dataset contained four labeled categories, including normal, ear ventilation tube, otitis externa, and tympanoscleros, our custom model we achieved a validation accuracy of 74%. However, the ResNet101 model's performance was 84% specified in terms of accuracy, refer to Table 2 along with Figs. 6 and 7. The categorization of AOM, CSOM, and pseudo membranes currently needs help due to issues with the available data. Moreover, the existing dataset for these categories needs to be bigger, further hindering the classification process.

To further improve the classification results, an ensemble classifier was created. This ensemble classifier utilized the outputs of the initial binary classification (normal vs. earwax) and performed a subsequent classification among normal, ear ventilation tube, otitis externa, and tympanoscleros. The ensemble approach exhibited superior accuracy to individual models, achieving an average diagnostic accuracy of 93.73%. These results highlight the effectiveness of our custom three-layer convolutional 2D model in accurately classifying eardrum disease images. The model outperformed the standard ResNet101 model regarding accuracy and parameter efficiency. Additionally, the ensemble classifier further improved the classification accuracy, emphasizing its potential in clinical diagnosis. The findings from this study contribute to the field of medical image classification and have implications for the accurate identification and treatment of eardrum diseases. Please refer to Table 2 for detailed information on accuracy, parameters, learning rate, epochs, input pixels, and batch size for each model.

Table 1 Binary classification: (*normal, earwax*)

Model	Accuracy (max) (%)	Train accuracy (%)	Learning rate	Input pixels	Batch size	Epochs	Total param.	Trainable param.	Non-trainable param.
Our model	93	94	0.001	64 * 64	32	200	249,537	249,537	0
ResNet101	89	87	0.001	224 * 224	32	150	65,137,698	22,479,522	42,658,176

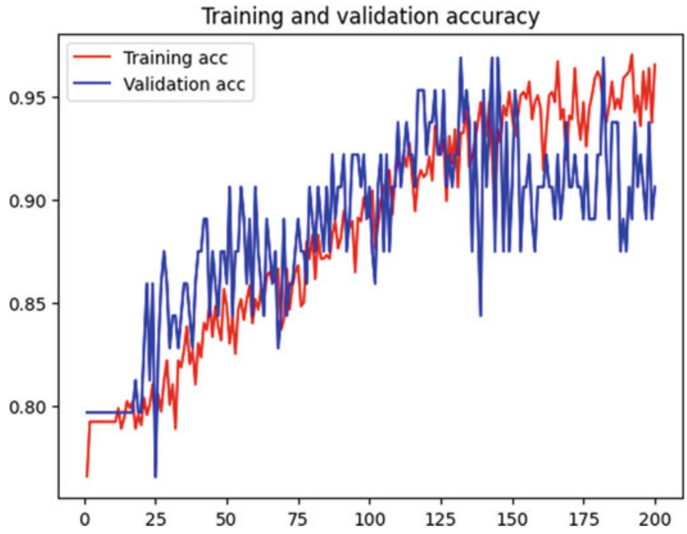


Fig. 4 Training and validation accuracy for binary classification using our model

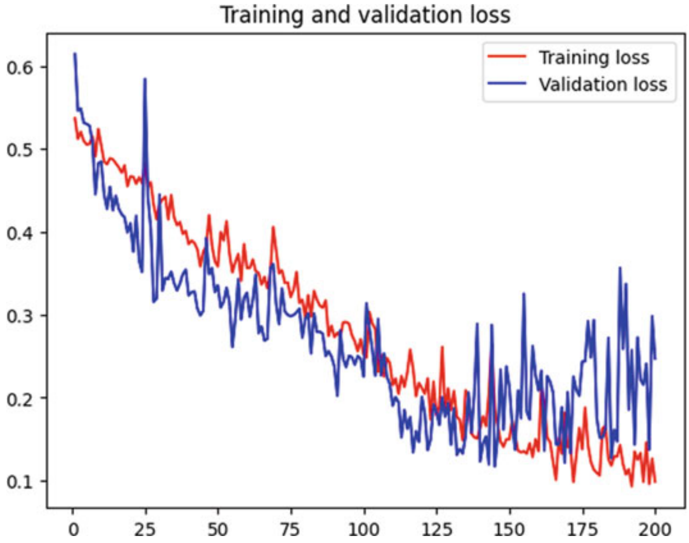


Fig. 5 Training and validation loss for binary classification using our model

4 Discussion

This article focuses on classifying ear diseases using two different approaches: convolutional 2-D neural networks and ResNet101. Both models were trained and evaluated on a dataset of ear images containing various disease classes. This discussion will

Table 2 Multi-class classification: (*1. Ear ventilation tube, 2. Ear wax, 3. Normal, 4. Otitis externa, 5. Tympanoscleros*)

Model	Val accuracy (max) (%)	Train accuracy (%)	Learning rate	Input pixels	Batch size	Epochs	Total param.	Trainable param.	Non-trainable param.
Our model	74	75	0.001	64 * 64	32	100	249,537	249,537	0
ResNet101	80	82	0.001	224 * 224	32	100	65,138,148	22,479,972	42,658,176

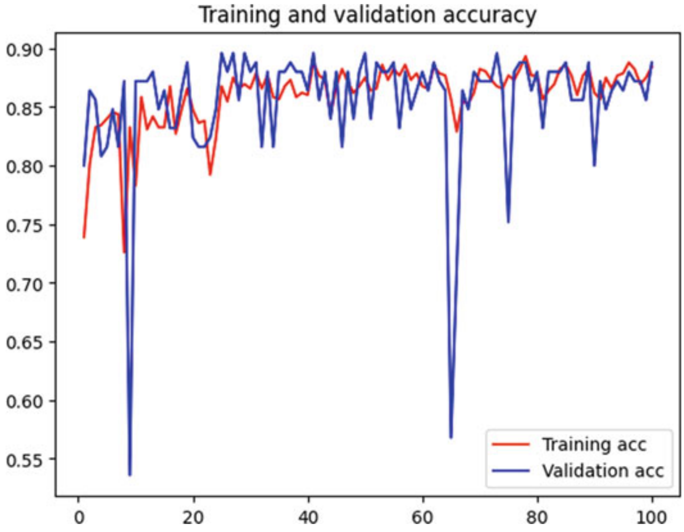


Fig. 6 Training and validation accuracy for binary classification using the ResNet101 model

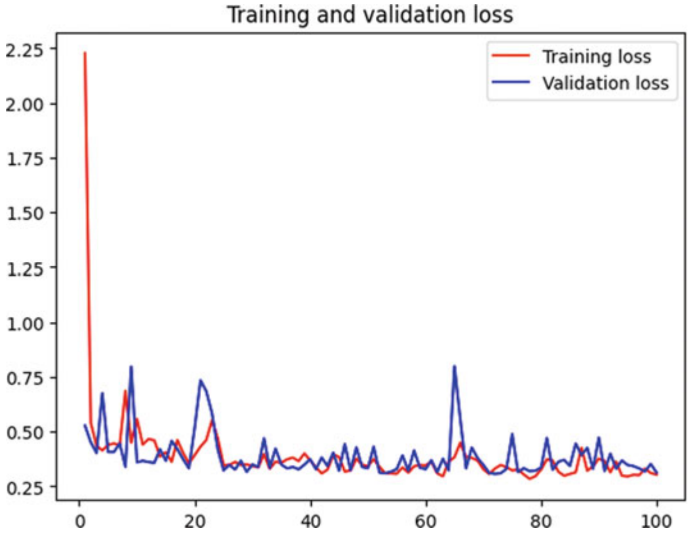


Fig. 7 Training and validation loss for binary classification using the ResNet101 model

compare the performance of these models, highlight their significance, address the challenges encountered, and provide insights for future improvements.

4.1 Performance Comparison

As given in Table 3, the convolutional 2-D neural network demonstrated exceptional performance with an accuracy of 93% in the binary classification of ear disease (concerning Figs. 8 and 9). The model effectively distinguished between normal ears and those with abnormalities. This high accuracy indicates that the 2-D CNN successfully captured relevant features and patterns for discriminating between healthy and diseased ears. On the other hand, ResNet101 outperformed the convolutional 2-D neural network in the multi-class classification of ear diseases (concerning Figs. 10 and 11). As shown in Fig. 12, the ResNet101 model, with its deeper architecture and the integration of residual connections, achieved higher accuracy and better generalization across multiple disease classes.

4.2 Significance

The convolutional 2-D neural network and ResNet101 have significant implications in the field of ear disease classification. The convolutional 2-D neural network demonstrates the potential for accurate binary classification, allowing for the early detection of abnormalities and the identification of normal ears. This can aid in preventive measures and prompt treatment, improving healthcare outcomes. The model's high accuracy and robust performance make it a valuable tool for screening ear diseases, particularly when binary classification is sufficient. ResNet101, with its deep architecture and advanced residual connections [18], provides a powerful tool for the multi-class classification of ear diseases.

4.3 Challenges and Future Directions

Despite the promising performance of the convolutional 2-D neural network and ResNet101, several challenges remain in ear disease classification. One challenge is the limited availability of labeled datasets. Obtaining large, well-curated datasets with diverse ear disease cases is crucial for training robust models [19]. The need for such datasets poses a challenge to training accurate and generalized models. Efforts should be made to collect comprehensive datasets, ensuring the representation of various ear diseases across different demographics and populations. Another challenge lies in the interpretation [20] of model predictions. Deep learning models are often considered black boxes, making it difficult to understand the decision-making process. Developing techniques to explain the model's predictions and provide insights into the features influencing classification can enhance trust and facilitate clinical adoption. Research in interpretable deep learning methods specific to ear disease classification is necessary.

Table 3 Performance comparison of the tympanic membrane classification algorithm

Study	Tympanic membrane classification	Number of classification	Algorithm used	Accuracy (%)
The present study	Normal versus ear wax Ear ventilation tube, ear wax, normal, otitis externa, and tympanoskleros	2 5	3 Convolutional 2D layer CNN model and ResNet101 3 Convolutional 2D layer CNN model and ResNet101	93 and 89 74 and 80
Hayoung Byun et al. (2022)	Normal versus abnormal Normal, OME, and COM Normal, OME, perforation, cholesteatoma	2 3 4	Teachable machine® Teachable machine® Teachable machine®	90.8 ± 1.5 87.8 ± 1.7 85.4 ± 1.7
Alhudhaif et al. (2021) [7]	Normal, AOM, CSOM, earwax	4	CBAM	98.26
Crowson et al. (2021) [16]	Normal versus OME	2	ResNet34	84.06
Tsutsumi et al. (2021) [14]	Normal versus abnormal	2	InceptionV3 MobileNetV2	73.0 77.0
Habib et al. (2020) [8]	Normal versus perforation	2	InceptionV3	76.00
Cai et al. (2021) [17]	Normal, OME, CSOM	3	ResNet50	93.4
Wu et al. (2021) [4]	Normal, AOM, OME	3 3	Xception MobileNetV2	90.66 88.56
Cha et al. (2019) [15]	Normal versus abnormal	2 2 2	InceptionV3 ResNet101 Ensemble network	93.31 91.88 94.17
Livingstone et al. (2019) [18]	Normal, earwax, tympanostomy tube	3	CNN	84.44

5 Conclusion

In the proposed model, the classification of ear diseases using convolutional 2-D neural network and ResNet101 models. The convolutional 2-D neural network demonstrated excellent performance in binary classification, distinguishing between

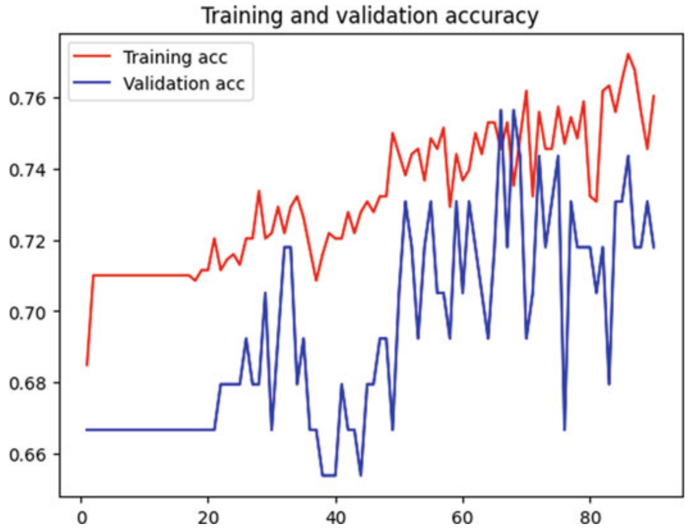


Fig. 8 Training and validation accuracy for multi-class classification using our model

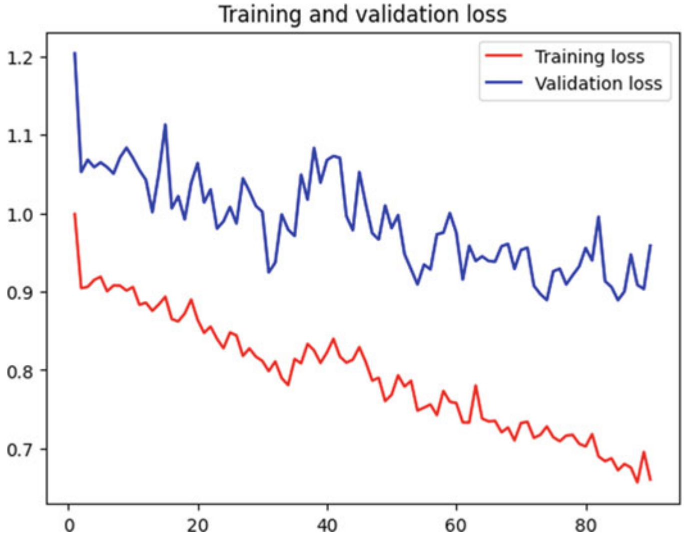


Fig. 9 Training and validation loss for multi-class classification using our model

normal ears and those with abnormalities. On the other hand, ResNet101 outperformed the 2-D CNN model in multi-class classification, showcasing its ability to classify various types of ear diseases accurately. Both models hold significant promise in the field of ear disease classification. The convolutional 2-D neural network provides a valuable tool for early detection and screening, while ResNet101's

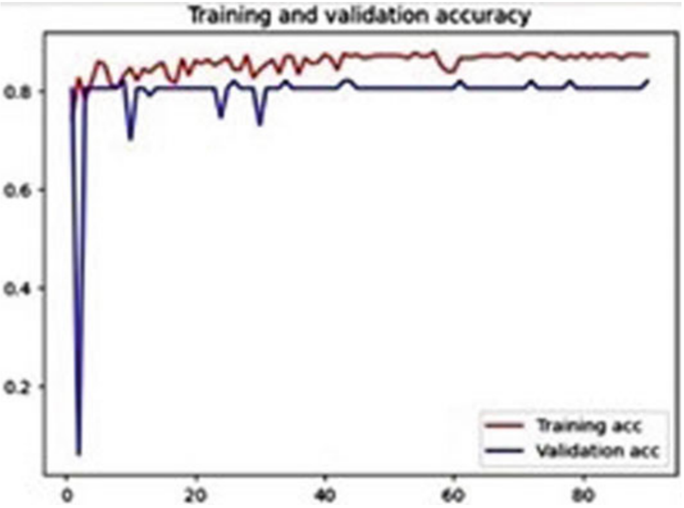


Fig. 10 Training and validation accuracy for multi-class classification using ResNet101 model

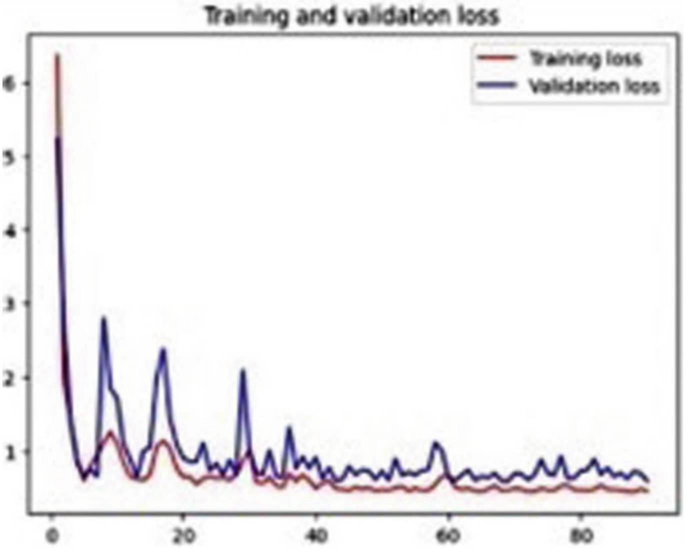


Fig. 11 Training and validation loss for multi-class classification using ResNet101 model

advanced architecture and transfer learning capabilities enable accurate multi-class classification. However, challenges, such as limited datasets, model interpretability, and real-world deployment constraints must be addressed. Efforts in data collection, interpretability techniques, and collaborations between researchers and healthcare professionals are necessary to overcome these challenges and facilitate the adoption

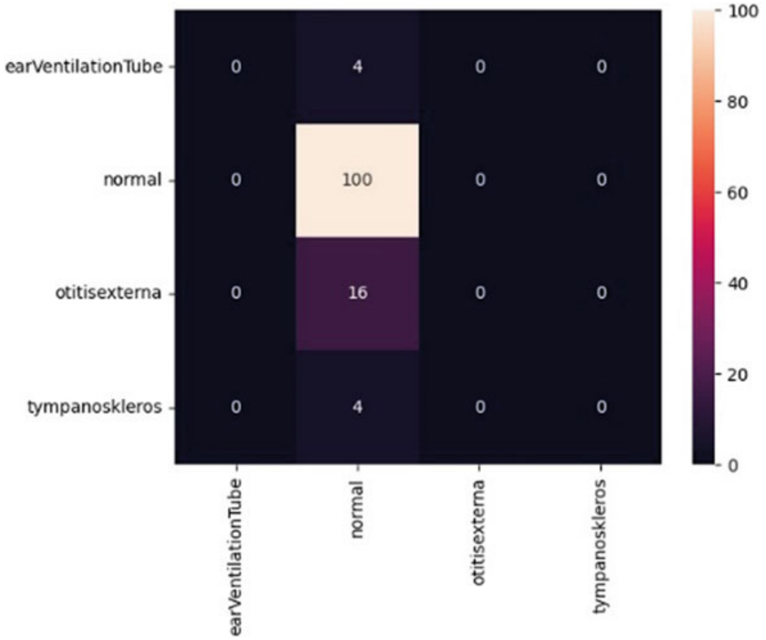


Fig. 12 Performance matrix for the proposed multi-class model

of these models in clinical scenarios. As research continues, exploring alternative deep learning architectures, ensemble techniques, attention mechanisms, and multimodal approaches can further enhance the accuracy and prevention of community spread and genetic disorders for the practicality of ear disease classification models.

Future Directions and Recommendations: Moreover, deploying these models in real-world healthcare requires careful consideration of practical constraints. Integration with existing clinical workflows, data privacy and security, and real-time processing are among the challenges to address. Collaborations between researchers, clinicians, and industry professionals can help bridge the gap between technical advancements and real-world healthcare applications.

Future directions for research include exploring the potential of other deep learning architectures and assembling techniques. Investigating the use of attention mechanisms, which focus on relevant regions of the ear images, can further improve classification accuracy. Additionally, exploring multimodal approaches by combining image data with clinical information, such as patient demographics and medical history may enhance the overall performance of the models.

References

1. Buchanan CM, Pothier DD (2008) Recognition of pediatric otopathology by General Practitioners. Int. J. Pediatr. Otorhinolaryngol 72:669–673. https://doi.org/10.1016/j.ijporl.2008.01.030, http://www.ncbi.nlm.nih.gov/pubmed/18325603
2. Pichichero ME (2021) Can machine learning and AI replace otoscopy for diagnosis of otitis media? Pediatrics 147:e2020049584. https://doi.org/10.1542/peds.2020-049584, http://www.ncbi.nlm.nih.gov/pubmed/33731368
3. Byun H, Yu S, Oh J, Bae J, Yoon MS, Lee SH, Chung JH, Kim TH (2021) An assistive role of a machine learning network in diagnosis of middle ear diseases. J. Clin. Med. 10:3198. https://doi.org/10.3390/jcm10153198, http://www.ncbi.nlm.nih.gov/pubmed/34361982
4. Wu Z, Lin Z, Li L, Pan H, Chen G, Fu Y, Qiu Q (2021) Deep learning for classification of pediatric otitis media. Laryngoscope 131:E2344–E2351. https://doi.org/10.1002/lary.29302, http://www.ncbi.nlm.nih.gov/pubmed/33369754
5. Khan MA, Kwon S, Choo J, Hong SM, Kang SH, Park IH, Kim SK, Hong SJ (2020) Automatic detection of the tympanic membrane and middle ear infection from to-endoscopic images via convolutional neural networks. Neural Netw 126:384–394. https://doi.org/10.1016/j.neunet.2020.03.023, http://www.ncbi.nlm.nih.gov/pubmed/32311656
6. Zeng X, Jiang Z, Luo W, Li H, Li H, Li G, Shi J, Wu K, Liu T, Lin X et al (2021) Efficient and accurate identification of ear diseases using an ensemble deep learning model. Sci Rep 11:10839. https://doi.org/10.1038/s41598-021-90345-w
7. Alhudhaif A, Cömert Z, Polat K (2021) Otitis media detection using tympanic membrane images with a novel multi-class machine learning algorithm. PeerJ Comput Sci 7:e405. https://doi.org/10.7717/peerj-cs.405, http://www.ncbi.nlm.nih.gov/pubmed/33817048
8. Habib AR, Kajbafzadeh M, Hasan Z, Wong E, Gunasekera H, Perry C, Sacks R, Kumar A, Singh N (2022) Artificial intelligence to classify ear disease from otoscopy: A systematic review and meta-analysis. Clin Otolaryngol 47:401–413. https://doi.org/10.1111/coa.13925, http://www.ncbi.nlm.nih.gov/pubmed/35253378
9. Korot E, Guan Z, Ferraz D, Wagner SK, Zhang G, Liu X, Faes L, Pontikos N, Finlayson SG, Khalid H et al (2021) Code-free deep learning for multi-modality medical image classification. Nat Mach Intell 3:288–298. https://doi.org/10.1038/s42256-021-00305-2
10. Jeong H (2020) Feasibility study of Google's teachable machine in diagnosis of tooth-marked tongue. J Dent Hyg Sci 20:206–212
11. Oyewumi M, Brandt MG, Carrillo B, Atkinson A, Iglar K, Forte V, Campisi P (2016) Objective evaluation of otoscopy skills among family and community medicine, pediatric, and otolaryngology residents. J Surg Educ 73:129–135. https://doi.org/10.1016/j.jsurg.2015.07.011, http://www.ncbi.nlm.nih.gov/pubmed/26364889
12. Pichichero ME, Poole MD (2011) We are assessing diagnostic accuracy and tympanocentesis skills in managing otitis media. Arch Pediatr Adolesc Med 155:1137–1142. https://doi.org/10.1001/archpedi.155.10.1137, http://www.ncbi.nlm.nih.gov/pubmed/11576009
13. Lee JY, Choi S-H, Chung JW (1827) Automated classification of the tympanic membrane using a convolutional neural network. Appl Sci 2019:9. https://doi.org/10.3390/app9091827
14. Tsutsumi K, Goshtasbi K, Risbud A, Khosravi P, Pang JC, Lin HW, Djalilian HR, Abouzari M (2021) A web-based deep learning model for automated diagnosis of otoscopic ımages. Otol Neurotol 42:e1382–e1388. https://doi.org/10.1097/MAO.0000000000003210, http://www.ncbi.nlm.nih.gov/pubmed/34191783
15. Cha D, Pae C, Seong SB, Choi JY, Park HJ (2019) Automated diagnosis of ear disease using ensemble deep learning with a big otoendoscopy image database. EBioMedicine 45:606–614. https://doi.org/10.1016/j.ebiom.2019.06.050, http://www.ncbi.nlm.nih.gov/pubmed/31272902
16. Crowson MG, Hartnick CJ, Diercks GR, Gallagher TQ, Fracchia MS, Setlur J, Cohen MS (2021) Machine learning for accurate ıntraoperative pediatric middle ear effusion diagnosis. Pediatrics 147:e2020034546. https://doi.org/10.1542/peds.2020-034546, http://www.ncbi.nlm.nih.gov/pubmed/33731369

17. Cai Y, Yu JG, Chen Y, Liu C, Xiao L, Grais EM, Zhao F, Lan L, Zeng S, Zeng J et al (2021) Investigating the use of a two-stage attention-aware convolutional neural network for the automated diagnosis of otitis media from tympanic membrane images: a prediction model development and validation study. BMJ Open 11:e041139. https://doi.org/10.1136/bmjopen-2020-041139, http://www.ncbi.nlm.nih.gov/pubmed/33478963
18. Livingstone D, Talai AS, Chau J, Forkert ND (2019) Building an otoscopic screening prototype tool using deep learning. J Otolaryngol-Head Neck Surg 48:66. https://doi.org/10.1186/s40463-019-0389-9, http://www.ncbi.nlm.nih.gov/pubmed/31771647
19. Myburgh HC, Jose S, Swanepoel DW, Laurent C (2018) Towards low-cost automated smartphone- and cloud-based otitis media diagnosis. Biomed Signal Process Control 39:34–52. https://doi.org/10.1016/j.bspc.2017.07.015
20. Uz Zaman S, Rangankar V, Muralinath K, Shah VKG, Pawar R (2022) Temporal bone cholesteatoma: typical findings and evaluation of diagnostic utility on high resolution computed tomography. Cureus 4: e22730
21. Tympanic membrane/eardrum dataset/otitis media. Kaggle.com. Accessed on 09 Nov 2023

Pay-by-Palm: A Contactless Payment System

Sridevi Saralaya, Pravin Kumar, Mohammed Shehzad, Mohammed Nihal, and Pragnya Nagure

Abstract Current payment systems, including cash, credit cards, and UPI can be inconvenient for users, prompting the need for a more robust and user-friendly payment system. Biometric authentication methods like palm prints can enhance security and the user experience, but there is a lack of a reliable system that integrates palm print recognition with e-wallets to facilitate payments at participating merchants. Existing payment systems fail to provide a secure and convenient way to pay using palm prints, with challenges regarding the accuracy, reliability, and privacy of palm print recognition technology. By integrating palm print recognition technology with e-wallets, this work aims to meet the growing demand for a more advanced payment system that enhances the user experience while providing a secure way to make payments.

Keywords Palm print · Autoencoders · AWS · Contactless payment · Biometric · e-wallets

S. Saralaya (✉) · P. Kumar · M. Shehzad · M. Nihal · P. Nagure
St Joseph Engineering College, Mangaluru, Karnataka, India
e-mail: sridevis@sjec.ac.in

Visvesvaraya Technological University, Belagavi, Karnataka, India

P. Kumar
e-mail: pravinkumar.8222@gmail.com

M. Shehzad
e-mail: m.shehzadsajid@gmail.com

M. Nihal
e-mail: mnihal425@gmail.com

P. Nagure
e-mail: pragnya.nagure2002@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_25

1 Introduction

The advent of e-commerce together with the growth of the Internet promoted the digitization of the payment process with the provision of various online payment methods like electronic cash, debit cards, credit cards, contactless payment, mobile wallets, etc. [1]. Online payments have several advantages, such as a significant impact on the economy, uprooting shadow economies, efficiency, better economic performance, safety, and ease of operation [1]. In order to be widely accepted payment methods across the globe, online payment systems must follow an efficient security protocol that must ensure high security for online transactions along with other requirements such as a) confidentiality of information shared by customers, b) data integrity, c) authentication of all participants, d) non-repudiation, and e) end-user requirements that include usability, flexibility, affordability, reliability, speed of transaction, and availability [2].

Despite there are numerous advantages of online payment systems, they have their own difficulties and challenges such as infrastructure, regulatory, legal, and sociocultural issues. Infrastructure is a fundamental challenge for the effective execution of online payments, as appropriate infrastructure for online payments is not always available or accessible to everyone. Regulatory and legal issues can also pose challenges, as different countries have different laws and regulations regarding online payments. Sociocultural issues, such as a lack of trust in online payment systems or a lack of awareness about how to use them can also be obstacles to widespread adoption of online payment methods. Premium pricing of the payment system, perceived security risks, incompatibility with large payments, and the immaturity of the mobile payments market are some of the barriers in adoption of online payment methods.

With the advancements in digital payments, customers in the current decade use various options for payment, as mentioned above. This has made a more user-friendly and secure payment mechanism necessary. Using biometric identification techniques like palm prints could be a solution. However, there is not currently a trustworthy system that combines palm print recognition with e-wallets for payments at participating businesses without any physical contact. In order to provide a safe and practical method of making payments, this work integrates palm print recognition technology with e-wallets. The aims of our work are as follows:

- To provide faster and more convenient payment options for customers in busy locations.
- Enhancing the security of the payment system by using palm as a form of biometric authentication.
- To enhance the privacy of customers by not having to share personal bank account details.
- Eliminate the need for customers to remember any secret pin or password for their transactions, simplifying the payment process and enhancing the user experience.

Palm print recognition process consists of four steps: image acquisition, image processing, feature extraction, and matching [3, 4]. The popular PolyU dataset from

IITD [5] was used in our study for training the autoencoders [6, 7]. The matching is performed using the prominent distance metrics such as Euclidian distance or Hamming distance and many more. The convolutional autoencoder architecture consists of an encoder and decoder with low information loss and simple structure. The encoder is used to convert the input image into low-dimensional features, which result in a loss of information. The decoder projects the feature back to the original space to obtain a reconstructed image. By constraining difference between the reconstructed image and the original image, the lost information can be minimized, and the discriminative features can be extracted [6].

Our study aims to capture palm images effectively, create an algorithm to extract features (palm prints) from these images, convert them into low-level features using an autoencoder, and enhance the precision of palm recognition for repeated payments.

2 Literature Survey

The work by author Wei Wu discusses two main methods for palm vein recognition: edge detection algorithms and texture-based approaches. The edge detection algorithm is used to extract the orientation and location information of ridges, lines, or feature points from the palm vein images [3]. Texture-based approaches utilize various statistical texture features such as local derivative patterns (LDP), local binary patterns (LBP), and Gabor filters [3]. The advantages of using palm vein recognition mentioned are: high security, liveness detection, user acceptability, and convenience [3]. On the other hand, there are some disadvantages associated with palm vein recognition: vulnerability to spoofing attacks; low cost and miniaturization acquisition; extraction from low contrast; high noise; and uneven illumination images [3].

The work by A. S. Ungureanu provides a comprehensive review of palm print recognition and various state-of-the-art methods for feature extraction in palm print recognition. Mainly two approaches are highlighted in this paper: (1) Conventional approaches: (a) encoding the line orientation at pixel level; (b) encoding the line orientation at region level, both with (i) generic texture descriptors and (ii) palm print-specific descriptors; (2) Neural network approaches: (a) fixed kernels, such as Scat Net; (b) kernels learned based on a training distribution, such as PCANet; and (3) Deep learning approaches [4]. The advantages of feature extraction methods mentioned in this paper include robustness, improved performance, and flexibility, along with disadvantages such as complexity due to intensive computations and resource-constrained devices, data requirements, and generalization limitations [4].

The work by Shao provides insights on cross-domain palm print recognition, where the palm prints are acquired with different illumination conditions, sensors, and resolutions. Two databases are randomly selected to form cross-domain pairs. The transfer convolutional autoencoder is trained using the labeled data from the multispectral palm print database and then fine-tuned using the unlabeled data from the uncontrolled palm print database. This autoencoder extracts low-dimensional features and reduces the domain gap between different databases. The experimental results show that the proposed transfer autoencoder approach can greatly enhance cross-domain recognition accuracy, up to 23.26%. Specifically, the proposed method outperforms other state-of-the-art methods, such as PCA, LBP, and CNN, in terms of recognition accuracy in different domains [6].

The methodology proposed in the study by Harivinod is a bimodal biometric system that includes: (a) palm print and face image acquisition followed by preprocessing; (b) feature extraction using Gabor filters and feature fusion; (c) Kernel Fisher analysis and preparation of a knowledge base; and (d) feature matching. The proposed bimodal biometric system described in this paper combines two modalities, palm print and face, which are simple and easy to access and incorporate contactless image capturing systems. The system uses Gabor filters to extract features from both modalities and combine them at the feature level using feature fusion [8]. Kernel Fisher discriminant analysis is then performed to obtain the features and create a knowledge base. The combination of modalities and feature fusion improves the accuracy of results and reduces the drawbacks of single biometric traits [9].

In the study by Qian Zheng, the authors evaluated palm print matching using both contactless and contact-based methods. Contactless palm print matching involves capturing images of the palm without physical contact, while contact-based palm print matching involves capturing images of the palm with physical contact. The authors used publicly available databases of 177 and 200 subjects to evaluate the effectiveness of their proposed 3D palm print feature extraction and matching method for both contactless and contact-based palm print matching. Their experimental results showed that their proposed method significantly outperformed other 3D palm print methods in terms of matching accuracy, template size, feature extraction time, and matching time [10]. Also, in general if we consider, palm print images have more discriminative features such as ridges and palm line which ensures recognition accuracy [11].

3 System Design and Development

Functional Requirements The goal of this work is to develop a system that can recognize palm prints and use that data to facilitate payment. Users can link their palm prints to an e-wallet and then use that palm print to make payments by scanning their palm at certain stores or merchant endpoints.

To use this system, the users will first need to scan and register their palm prints with the system and link them to an e-wallet through their mobile device. Once registered, users can make payments hassle-free. This process only must be done one time, after that the users can make payments without having to carry around cash or credit cards.

Non-functional Requirements Our study utilizes palm prints to verify and authorize user's identity and enable payment transactions, providing increased security, convenience, and speed. The non-functional requirements are:

- **Faster**: This should be designed such that the customer can make payments faster through the e-wallet connected with their palm print.
- **Performance**: The palm print recognition and payment transaction should be fast and responsive with low latency and less failure rates to ensure seamless user experience.
- **Usability**: The interface and usage of the system should be user-friendly and easy to use, even for ordinary non-professional individuals.
- **Integration**: The system enhances the flexibility in payment by allowing users to add money into the e-wallet from their personal bank accounts.

Architectural Diagram Figure 1 shows the structure and services used in our work. A Raspberry Pi controls the camera and display and sends requests to an EC2 server for registration and matching. The server loads an autoencoder model from an S3 bucket to generate 32-bit vectors of images, which are stored in DynamoDB and sent to the merchant interface to display a QR code. The customer scans the code with an app and enters their details to complete registration and enable their e-wallet. Firebase stores user and transaction details for easy integration with Flutter applications.

Activity Diagram Figure 2 describes the merchant interface, the login page appears, and the user enters their credentials. After logging in, the system is presented with two options: 'Registration' and 'Payment'. If the customer is not registered, they can scan their palm on the interface, the interface then captures the palm print, processes

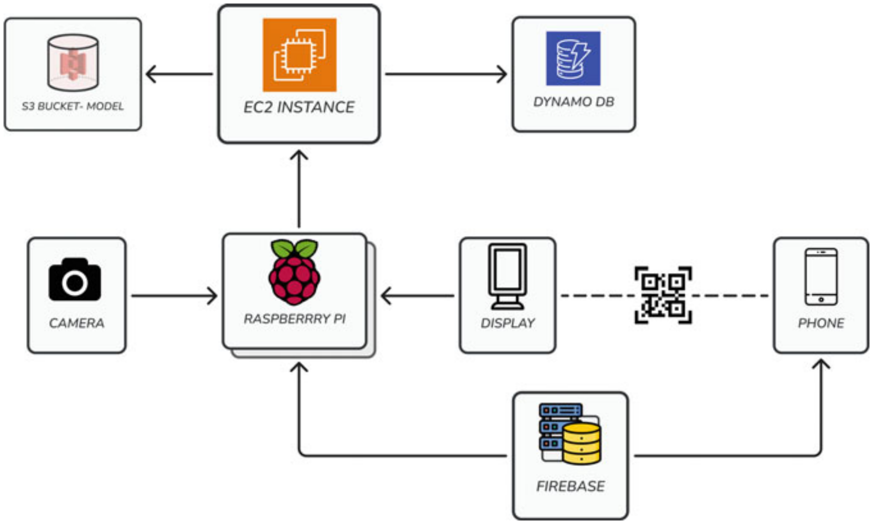


Fig. 1 Architectural diagram

it, and generates a QR code that is to be scanned by the customer using our mobile app and enter the required credentials. After registration, the customer can checkout using their palm by scanning on the interface. During checkout if the scanned palm matches with the registered palm, then payment is facilitated, and the activity is completed.

Sequence Diagram The sequence of events described in Fig. 3 begins with the user registering themselves by scanning their palm on the payment interface. The palm image is saved on the Raspberry Pi. The palm ROI is then extracted, and filters are applied to enhance the image quality. Once this is done, the preprocessed image is sent to the API, where the autoencoder model generates a feature vector. This vector is stored in the database for future matching. This marks the end of user registration, and an appropriate message is displayed on the interface.

Once the user is registered, they can initiate payment by scanning their palm over the interface. Again, the palm image is saved on the Raspberry Pi, preprocessed,

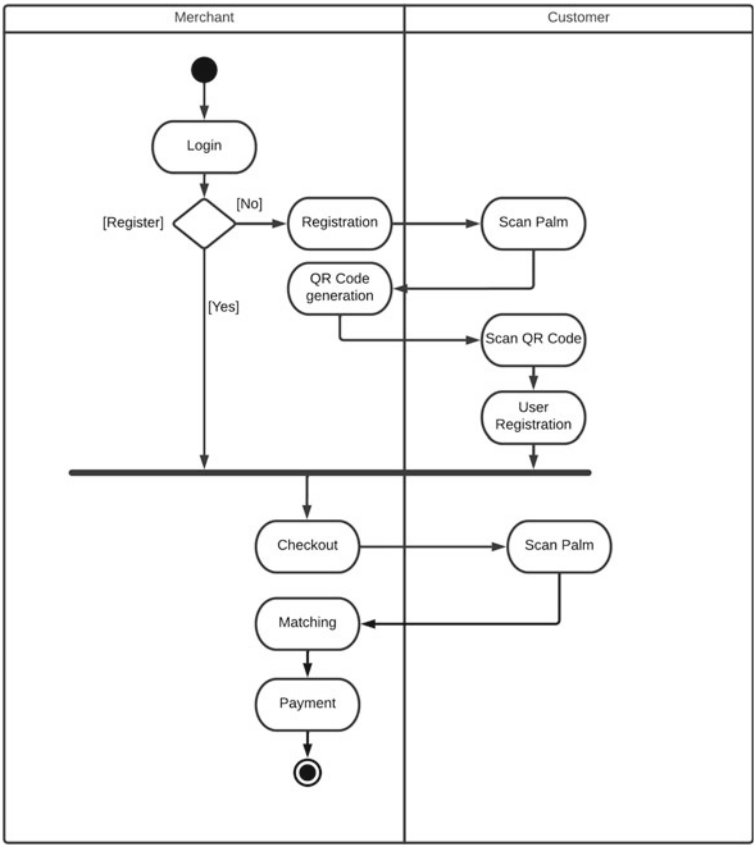


Fig. 2 Activity diagram

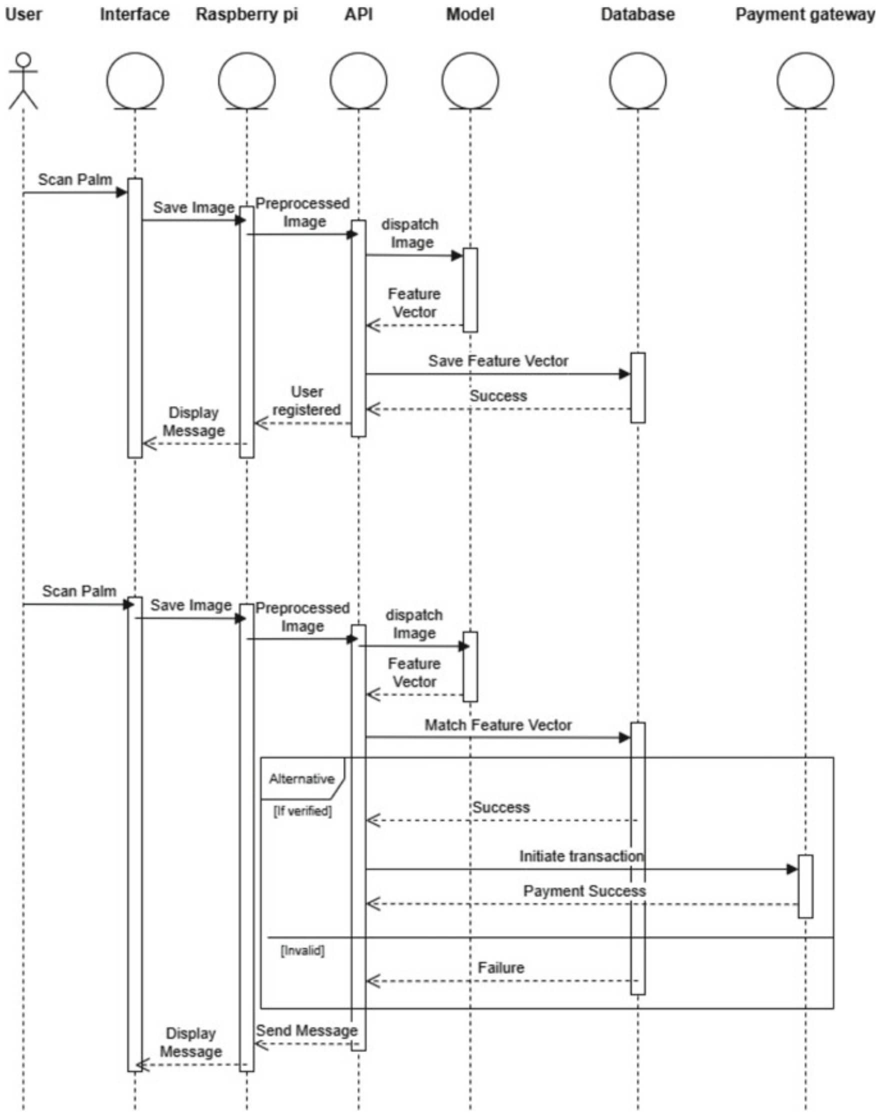


Fig. 3 Sequence diagram

and sent to the API. The API generates a new feature vector using the preprocessed image. This vector is then matched with the one present in the database. If the user is verified, the transaction will be facilitated; otherwise, the transaction will be denied and appropriate messages will be displayed on the interface.

Control Flow Diagram Figure 4 describes the control flow that begins with the merchant-side application. Upon launching the application, the user is prompted to choose whether the customer is a registered user or not. If the customer is not registered, the system presents a window to activate the live camera feed once the IR sensor detects a palm in front of the camera. Next, the ROI is extracted from the customer's palm. This image is processed by OpenCV and sent to the server, where a 32-bit vector is generated from the image and saved in the database, thus marking the registration of the customer as complete.

In case the customer is already registered, they will go through a similar process for having their palm scanned and the ROI extracted using OpenCV, after which the image is sent to the server, where the 32-bit vector of the newly generated image and the ones existing in the database are matched. Finally, if the match is successful, the merchant-side application will show appropriate messages on the display indicating transaction success; if the match has failed, then the display will indicate transaction failure.

4 Methodology

The approach that we chose to solve this problem is by allowing the customers to pay using their palm without the hassle of standing in huge lines to pay using cash or fiddle with the cards. Figure 5 illustrates the overall flow of the system. We studied various research papers related to biometrics, palm print extraction, and palm vein extraction. The research papers contained various methodologies and techniques for ROI extraction and preprocessing palm images. The dataset that most of the research papers mentioned were the PolyU dataset and the CASIA dataset. Hence, we requested a PolyU dataset from IIT Delhi, which consists of images of the palms of 230 people, with six left palm images and five right palm images. According to the research papers we referred to, the first step to building palm print recognition was developing an algorithm to extract the ROI from the palm with the help of OpenCV. To achieve this, the watershed algorithm was used, which segments areas of high pixel intensity. Using this, an idea to shine a light source on the palm was obtained, making the palm print a region of high intensity as an ROI for our purpose. The captured image of the palm was blurred and smoothened to remove noise, and a distance transform of 50% was applied to the image produced by thresholding. After the contours were detected, a bounding box was drawn around it, and the region of interest was cropped. This ROI image was then resized to maintain uniformity and weighted to darken the region. This algorithm was applied to all the images in the IITD PolyU dataset, and a new dataset was created from our ROI extraction algorithm.

Once the required ROI dataset was ready, we decided to apply deep learning techniques for feature extraction using convolutional autoencoders. An autoencoder contains two functions: an encoding function that transforms the input data into low-dimensional feature representation and a decoding function that recreates the input data from the encoded data. We designed a convolutional encoder architecture to

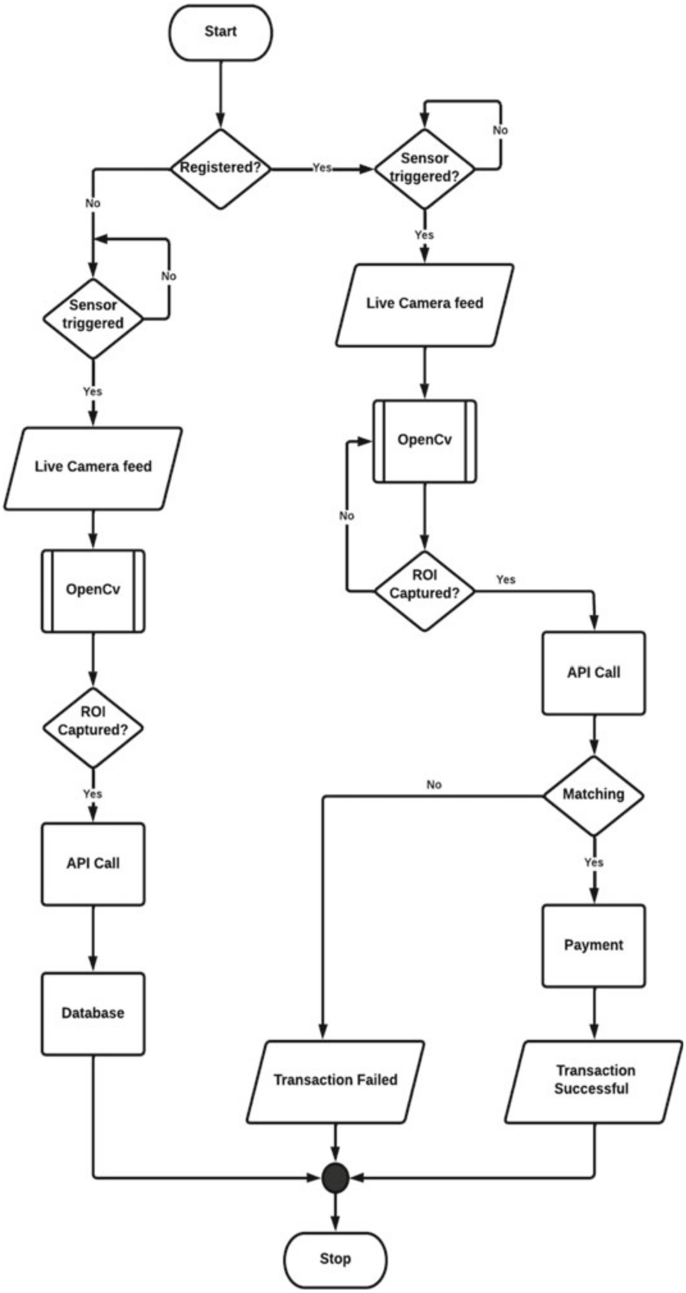


Fig. 4 Control flow diagram

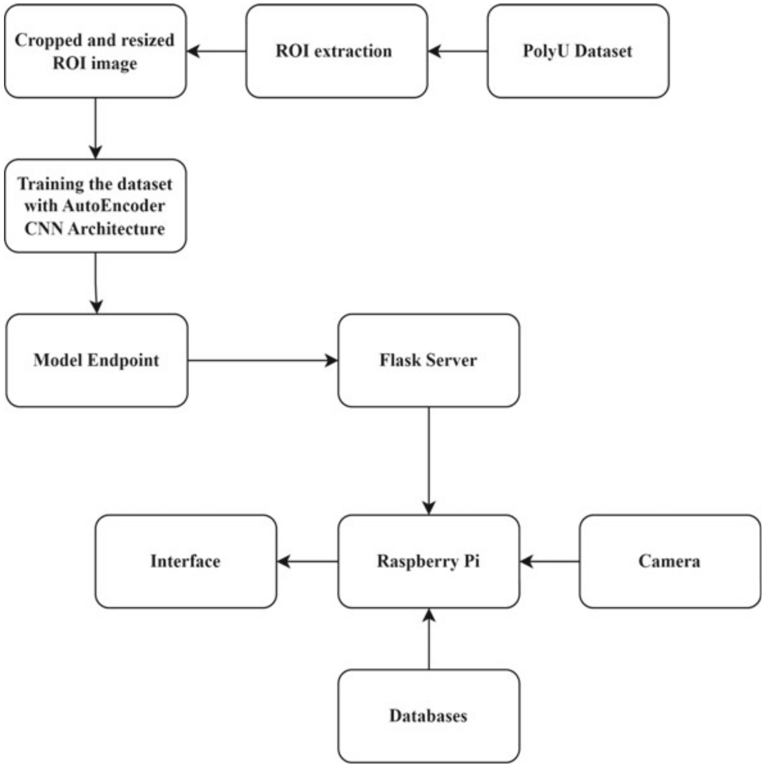


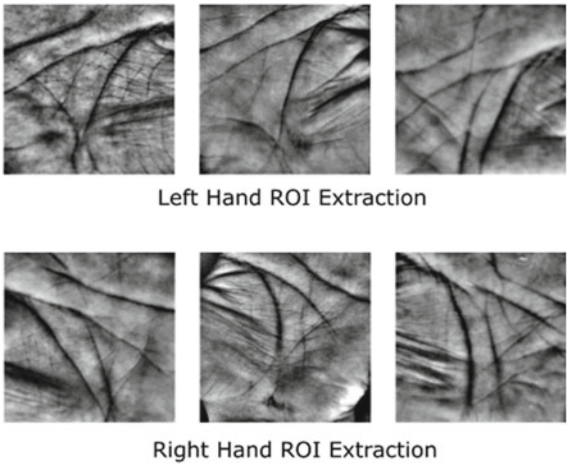
Fig. 5 Methodology flowchart

take a grayscale image of shape 150 × 150 as input and generate a 32-bit vector as an intermediate representation that can be stored in the database. The decoder architecture is the reverse of the encoder architecture and recreates the image from the generated 32-bit vector representation.

Out of 2600 ROI-extracted images, 2400 were split into the training set, and the remaining 200 were split into the validation set. Figure 6 represents some sample images from the ROI-extracted dataset. The ROI-extracted dataset was trained on the designed autoencoder model for 100 epochs, where accuracy and loss are measured based on how well the model recreates the input image. Figure 7 describes the pictorial representation of the measured metrics.

The trained model was then saved as TensorFlow model files and uploaded to the AWS S3 bucket. We are using Raspberry Pi 4 as a centralized system to facilitate the merchant-side GUI and functionalities such as palm image capturing, ROI extraction, live feedback in the display, API calls for registrations, and facilitating payment transactions. Pi Camera Module 3 with 12 mega-pixels is used to capture high-definition images, and the camera is triggered using an IR sensor, which activates when the palm is in closer proximity to the camera and sensor. An AWS EC2 instance

Fig. 6 Images of extracted ROI of palm from IITD dataset using watershed algorithm



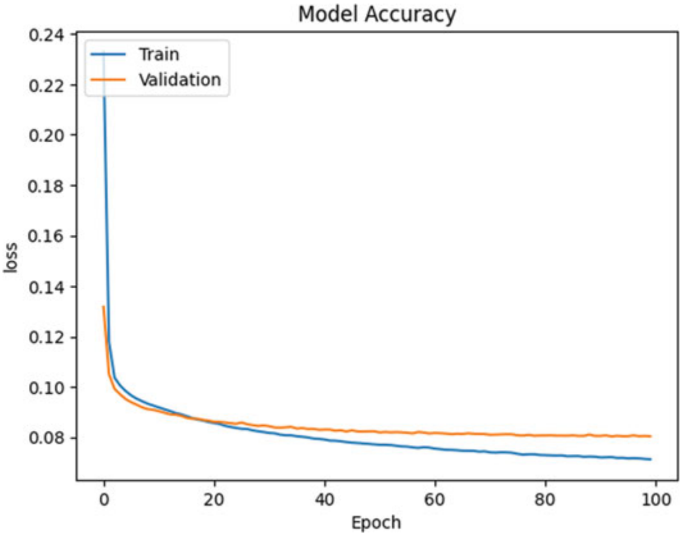


Fig. 7 Pictorial representation of model accuracy with training and validation set

is used to run a Flask app, which works as an API server to perform registrations, write into a database, and facilitate transactions. The model to predict the 32-bit vector for the given image is loaded from S3 buckets.

The end users are provided with two interfaces: one for merchants and another for customers. The merchant interface, i.e., desktop application has options to register new users and checkout during shopping. The customer interface, i.e., mobile application has features to monitor their previous transactions, balance amount, and transfer money into the e-wallet. During user registrations, the server generates a unique ID from their palm print image using an autoencoder model and a timestamp. This ID is embedded in a QR code and displayed for the user to scan with their mobile app to register. The user enters their credentials to enable their e-wallet and link it to their palm for contactless payments. At checkout, the merchant enters the amount, and the customer can just scan their palm. The server matches the processed palm image with the stored data and transfers the amount from the user's account to merchant's account. A success or failure signal is returned to the merchant's application.

The interface for merchants was developed using PyQT5, and the interface for customers was developed using Flutter. A custom enclosure was designed and fabricated using a 3D printer to accommodate all the hardware components of our prototype product.

5 Implementation

Software Implementation The system consists of two parts, namely merchant-side applications and customer-side applications. Beginning with the merchant-side application, the interface is implemented using PyQT, a cross-platform GUI toolkit in Python. The interface begins with a login page prompting the employee to enter their credentials, after which they are provided with two options: one for registering new customers and the other for checking out the items purchased by the existing customer. In both cases, the customer will be presented with a user-friendly live camera feedback interface integrated with PyQT, which is processed by OpenCV, a library for real-time image manipulation functionalities. Using this, the ROI of the palm is extracted and sent to the Flask server running on an AWS EC2 instance through an API call. The Flask server has several dependencies imported, such as TensorFlow for loading and using the trained autoencoder model, the AWS Python SDK to connect and integrate with S3 buckets, and DynamoDB, which are services provided by AWS to store files and NoSQL databases, respectively.

During registration process, the server will use the autoencoder model to generate a 32-bit vector from the supplied image and generate a unique hash ID using uuid4 hashing based on the current timestamp and the generated vector values. This hash ID is then returned as a response to the requested client. The client or merchant-side application embeds this hash ID within the QR code and displays it on the interface. The customer then uses their customer-side interface, a Flutter mobile application, to scan the QR code displayed. In the customer-side mobile application, the customer can click "Sign up" to scan the displayed QR code. On successful scan, the app takes them to a page where they must enter their credentials for their e-wallet to be enabled. Once the e-wallet is enabled, the customer's palm is registered and linked to the e-wallet, allowing for future contactless payments.

In the case of checkout, the merchant employee enters the amount to be billed and navigates to the scanner page, requesting the customer to scan their palm. Then the captured ROI of the customer is sent to the Flask server, which again converts the supplied image to a 32-bit vector using the autoencoder model and then applies the matching algorithm, where the distance between the newly generated vector and all the vectors stored in the database is calculated, the lowest distance is selected to be a match, and the unique hash ID with respect to that record is retrieved. This unique hash ID acts as a key to deduct the amount from the respected person's account to confirm the payment. Following the completion of all subprocesses, a success signal is returned to the merchant-side application. If there are any issues, the failure signal is returned.

Hardware Implementation The Raspberry Pi is a small, single board computer that can be connected to a computer display or TV and operates with a regular keyboard and mouse. It consists of general purpose input/output (GPIO) pins to monitor and control the outside world by connecting to external electronic circuits and sensors. It also consists of multiple slots and ports to connect to cameras, displays, external monitors, and the Internet.

We have used a Raspberry Pi 4, with a 64-bit quad-core Cortex-A72 processor, 2 GB of LPDDR4 RAM, 2 micro-HDMI ports, 4 USB ports, and a Gigabit Ethernet port. Pi Camera Module 3 is connected to the camera slot. Standard monitor connected through an HDMI cable.

An infrared sensor acts as a trigger to enable the camera, when the customer brings their palm closer to the IR sensor the motion is detected, and signal is sent to enable the camera. The IR sensor is placed closer to the camera, and it is powered by GPIO pins. The data pin of the IR sensor is connected to pin 16 of the Raspberry Pi to take the input from the sensor to enable or disable the camera. A prototype of the system is represented in Fig. 8.

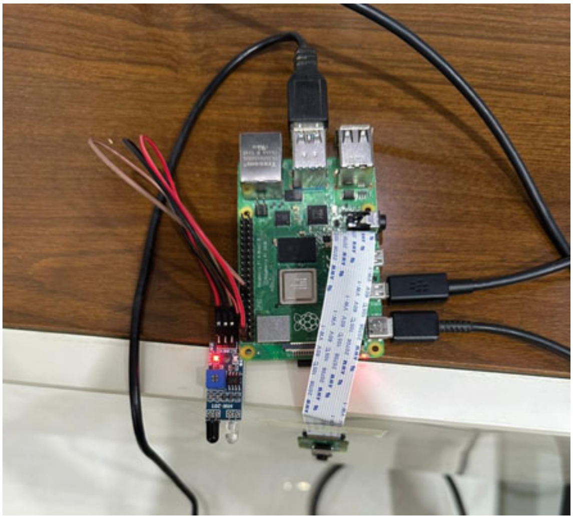


Fig. 8 Raspberry Pi 4 with camera module and IR sensor

Table 1 Accuracy results of various classification algorithms applied on ROI dataset

Algorithm	Accuracy (%)
K nearest neighbors	55
Random forest	54
SVM	45
GaussianNB	37

6 Results and Discussion

The developed system segments ROI in real time, generates the intermediate low-dimensional feature vector, finds the best match in the registered palm database, and facilitates contactless payment. In order to match the low-dimensional features of the palm, we tried both clustering and classification techniques.

From the ROI dataset that we generated, five people were randomly selected with five image ROIs each whose low-dimensional 32-bit feature vector predictions were used for the K-Means algorithm. The outcomes fell short of expectations.

Another approach that was used to test the matching capability was to evaluate the distance between each of the feature vectors and every other feature vector present in the dataset that we chose earlier. Four prominent distance metrics such as (i) Euclidean distance, (ii) Manhattan distance, (iii) MinKowski distance and (iv) Canberra distance were chosen and heatmaps were used for the representation. The anticipated pattern was to see light-colored shades along the diagonal region in the heatmap. The Manhattan distance heatmap provided significant pattern compared with other heatmaps.

The final approach was to use classification by applying labels to all palm image vectors according to who they belonged to. The different classification algorithms were applied, and the metrics are given in Table 1.

7 Advantages and Limitations

The current study offers a revolutionary approach to financial transactions with several advantages. Firstly, it enables seamless transactions, eliminating the need for physical cards or cash. By simply placing their palm on a scanner, users can effortlessly complete payments, streamlining the process and saving valuable time. Furthermore, the technology is contactless and hygienic, which is particularly advantageous in today's world. Users can make payments without touching any surfaces, reducing the risk of transmitting germs or viruses. Additionally, the work integrates e-wallets, preventing direct access to bank accounts. This enhances security by keeping sensitive financial information separate and adding an extra layer of protection against potential fraud.

Proper positioning of the user's palm is essential for accurate recognition, but challenges arise when the palm is wet, sweaty, covered in dirt or debris, or affected by skin conditions like eczema. Moisture interferes with recognition, while obstructions hinder the system's ability to capture clear features. Special consideration is needed for individuals with skin conditions, accommodating variations in palm characteristics. Furthermore, maintaining a high level of accuracy is crucial to prevent fraudulent exploitation, and robust encryption protocols should be employed to protect the transmission and storage of sensitive payment data.

8 Conclusion and Future Work

Current payment systems such as cash, credit cards, and UPI are not always convenient for users, which indicates the need for a more user-friendly payment system. Biometric authentication methods like palm prints can improve security and the user experience, but there is a lack of a reliable system that integrates palm print recognition with e-wallets for payments at participating merchants or store outlets. However, by integrating palm print recognition technology with e-wallets, it is possible to meet the growing demand for a more advanced payment system that enhances the user experience and provides faster payment alternative.

By integrating palm print recognition technology with e-wallets, our solution offers a secure, user-friendly payment option that enhances the overall payment experience. With our system, customers can quickly register their palm prints using a simple QR code scanning through mobile app and link it to their e-wallets [12]. This allows for future contactless payments, eliminating the need for physical cards or cash. In addition to that, our system's merchant-side application provides an easy-to-use interface for store employees, allowing them to quickly register new customers and process payments with just a palm scan. Our system uses technologies such as PyQT, OpenCV, and TensorFlow to ensure reliable and accurate palm print recognition and matching. By addressing the limitations of existing payment systems, our solution provides a more advanced payment option that meets the growing demand for faster, more convenient, and more user-friendly payment methods.

Although our system presents a viable solution for improving the payment experience, there is still room for further development and improvement. In the future, we plan to conduct more extensive user testing and gather feedback to refine the user interface and improve the overall user experience. Finally, we aim to make our system more accessible and scalable by developing a cloud-based solution that can be easily deployed and used in various locations. By continuously improving and expanding our system, we hope to provide a more advanced and inclusive payment option for users around the world.

References

1. Khan BUI, Olanrewaju RF, Baba AM, Langoo AA, Assad S (2017) A compendious study of online payment systems: past developments, present impact, and future considerations. Int J Adv Comput Sci Appl 8(5)
2. Bezovski Z (2016) The future of the mobile payment as electronic payment system. Eur J Bus Manage 8(8):127–132
3. Wu W, Elliott SJ, Lin S, Sun S, Tang Y (2020) Review of palm vein recognition. IET Biometrics 9(1):1–10
4. Ungureanu AS, Salahuddin S, Corcoran P (2020) Toward unconstrained palmprint recognition on consumer devices: a literature review. IEEE Access 8:86130–86148
5. Kumar A (2008) Incorporating cohort information for reliable palmprint authentication. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. IEEE, pp 583–590
6. Shao, H, Zhong D, Du X (2019) Cross-domain palmprint recognition based on transfer convolutional autoencoder. In: 2019 IEEE international conference on image processing (ICIP) IEEE, pp 1153–1157
7. Bachay FM, Abdulameer MH (2022) Hybrid deep learning model based on autoencoder and CNN for palmprint authentication. Int J Int Eng Syst 15(3):41
8. Han WY, Lee JC (2012) Palm vein recognition using adaptive Gabor filter. Expert Syst Appl 39(18):13225–13234
9. Harivinod N, Shekar BH (2019) A bimodal biometric system using palmprint and face modality. In: Data analytics and learning: proceedings of DAL 2018. Springer, Singapore, pp 197–206
10. Zheng Q, Kumar A, Pan G (2015) Suspecting less and doing better: New insights on palmprint identification for faster and more accurate matching. IEEE Trans Inf Forensics Secur 11(3):633–641
11. Zhong D, Du X, Zhong K (2019) Decade progress of palmprint recognition: a brief survey. Neurocomputing 328:16–28
12. Baidong H, Yukun Z (2019) Research on quickpass payment terminal application system based on dynamic QR code. J Phys: Conf Ser 1168(3):032059 (IOP Publishing)

Steganalysis of Reversible Digital Watermarking Algorithm Based on LWT and SVD

Geeta Sharma and Vinay Kumar

Abstract The processes of Digital Rights Management are used to limit access to proprietary and copyrighted material. Original creator of any digital data needs protection against his rights. Digital Rights Management involves two main concepts, watermarking and cryptography. We are working with reversible digital watermarking to keep cover and hidden data safe after extracting hidden message. In this work, digital image data is used as a mask, and using MD5 (Message Digest) algorithm, we have generated a digest of data. The specific digital data ID is created. In an embedding algorithm, this watermark is then integrated with digital data. Here, we worked on a reversible embedding algorithm that enabled us to use the cover even after the secret message had been removed. We have considered the different kinds of digital watermarking and the form of cover data in this report. Due to imperceptibility features, a watermarked data may be transmitted over the open network. It will therefore draw multiple attacks and must be robust. We have used LWT-QR-MD5 techniques that are a lossless approach. The degree of robustness is improved when a consistent procedure is used to build the digest of the watermark. The results are showing a remarkable improvement over LWT.

Keywords Digital watermarking · LWT · MD5 · Quantum steganography · PSNR

1 Introduction

The information about ownership of the digital data is inserted as a watermark in an imperceptible manner to prevent copyright and unauthorized duplication of the data over the network. The watermark should be added in a manner that it cannot be perceived by just looking at the image. When data is sent using a network, a number

G. Sharma (✉)
Jagan Institute of Management Studies, Sector-5, Rohini, Delhi, India
e-mail: geeta.sharma@jimsindia.org

V. Kumar
NIC, Govt. of India, Delhi, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_26

of intentional and unintentional changes are made to this data. Thus, the watermark can be destroyed in the due course of transmission and the retrieval of original image and watermark may get affected adversely. A watermark that can handle multiple attacks is known as “robust watermark”. Robustness and imperceptibility are adversely correlated. The two primary categories of watermarking methods are transform domain and frequency domain.

A frequency-domain method is simple and has very less computation overhead. It lacks in robustness at the same time. On the other hand, robustness is better in transform-domain methods but generally uses complicated calculations. “Discrete Wavelet Transform (DWT), Lifting Wavelet Transform (LWT) and Singular Value Decomposition (SVD)” are the few important instances of the frequency-domain watermarking. “When a particular frequency coefficient of the tampering area is modified only after any change applied to image, then it is known as spatio-frequency localization” [1] and makes the discrete wavelet transform a good choice for watermarking techniques. Many approaches in the research have used the discrete wavelet transform. Sweldens [2], suggested a novel type of DWT technique, called “Lifting Wavelet Transform, that is also known as the next generation of wavelet. DWT and LWT methods were applied combinedly in numerous watermarking techniques used for pictures” [3]. “DWT was used to embed watermarks using DWT with discrete-time system. One more research was conducted in [4], in which the DWT was paired with DCT for processing RGB pictures. The blend of DWT and DCT is also employed in [5] to create an efficient watermarking technique.” There are a number of researches done while combining the LWT with SVD, in order to exploit the benefits of both the worlds.

Filter banks or LWT can be used to perform the discrete wavelet transformation. A filter bank approach uses filters like low-pass and high-pass filter, later the “downsampling” is performed. These filters are applied on original signal. The important information of the signal are stored in approximation band and the high-frequency data which is almost half of the original size is stored in details band. They are known as filters’ outputs. Reversing the sequence of operations and restoring back the original message are the inverse operation. “After applying the low-pass filter (L) and the high-pass filter (H) to digital images, four bands are formed in each DWT decomposition, and such bands are indicated as LL, LH, HL, and HH” [5].

The Lifting Wavelet Transform has less computational process than the filter banks that are used in a normal DWT. Moreover, the Lifting technique uses integer-to-integer computing, which eliminates the need for floating point calculations. Three stages are essential to create the approximation and detail bands in LWT:

1. **Divide**—Firstly, the number of elements in the original message are counted and then these elements are categorized as even and odd. This way the message is split into two parts.
2. **Predict**—“In this phase, the details band is formed by calculating the difference between the original sample at an odd number and the anticipated odd sample, derived as the mean of the two adjacent even samples”[6].

3. **Update**—Generates the "estimate band" by changing the average quantities with the difference computed in the predict step.

In this paper, a thorough experimentation and evaluation are presented to compare the performance after embedding the one watermark with same intensity using different algorithms.

2 Related Work

The authentication of images is a challenging job on Internet due to heavy traffic and large number of images. A number of researchers have proposed different perspectives in their work. In [6], the identical quantum text message is stored using two qubit sequences in an enhanced ASCII-based quantum format. The secret quantum text of size $2n$ $2n1$ is scrambled using the Gray code transform method before being embedded in the cover image, which is initially separated into eight blocks of size $2n$ $2n$ $2n$ 1. The disorder quantum text is then inserted into the eight cover picture blocks using the Gray code as a criterion. Meanwhile, the corresponding quantum circuits are being sketched. It may be inferred from the analysis of all quantum circuits that the scheme has an $O(n)$ lower complexity. Additionally, the suggested scheme's effectiveness is evaluated using simulation results for three different criteria: robustness, circuit complexity, and visual quality. In [7], researchers conducted an initial analysis of the use of the MD5 algorithm in digital watermarks. It suggested that copyright data can be encrypted with the MD5 technique and established guidelines for second-value picture watermarks using the "DCT algorithm, which embeds an image by the carrier". The watermark can be detected and the MD5 code has restored by the extraction techniques [7]. The performance of the second-generation wavelet "lifting wavelet transform (LWT) and the discrete wavelet transform (DWT)" in the watermarking process is compared in this study. The center frequency bands of both transforms contain a watermark that is identical to the other in terms of intensity. According to the experimental findings, LWT is superior to "DWT in terms of objective image quality as measured by PSNR. After performing various assaults such JPG compression and LPF, DWT watermarked images have better robustness than LWT watermarked images as determined by NCC and BER. By taking into account the benefits of each alteration and combining them, the experimental tests presented in this research can be used to improve picture watermarking algorithms. The development of a watermarking algorithm based on both transformations is a future project" [8].

In [9], author suggested use of blockchain-based authentication with the Internet of Things (IoT) to improve security. "Therefore, to achieve better resilience, high image data security, increased imperceptibility, and embedding capacity, future researchers in the hybrid transform sector will need to combine machine learning and artificial neural network techniques [9]. To resist both the geometric distortion attack and the signal processing attack is the primary objective of watermarking. A watermarking method's effectiveness depends on a number of factors that must be taken into account. There is no watermarking method that can withstand all sorts of attack in classical information. However, many experts are currently looking for a better method that will produce results that are more reliable.

In this research, "a unique Least Significant Bit-based quantum watermarking method is analyzed" in which the owner of the carrier image watermarks an m-pixel grayscale picture in an m-pixel carrier image using the m-bit embedding key K1 and the m-bit extraction key K2. "In this technique for modeling quantum images, the Novel Enhanced Quantum Representation of digital images (NEQR protocol) proposed in [10] is employed". A detailed inspection of the suggested LSB-based quantum watermarking process reveals that no attacker can gain access to the watermarked secret image since the extraction key K2 is only known to the lawful owner of the original carrier image. As a result, the suggested watermarking protocol is a trustworthy watermark protocol. Furthermore, "it can be seen from the histogram graphs that the original cover image and the watermarked signal are perceptually indistinguishable and that the histogram graphs of the initial pictures and the resulting watermarked images are in acceptable agreement, proving that the protocol is imperceptible. The Peak Signal-to-Noise Ratio (PSNR) of the described LSB-based quantum watermarking has improved by roughly 6% when compared to Nan Jiang's LSB-based quantum steganography technique" [10].

3 Methodology

The algorithm analyzed in this paper is able of recovering the secret message losslessly. For authentication and copyright management, reversible automated watermarking is largely utilized. In general, analysis is revolved around the invisible algorithm, which was stable and less complicated. A watermark that still prioritizes more data than others and can archive it. We were considered the different kinds of digital watermarking and the form of cover data in this report. Due to imperceptibility features, a watermarked data may be transmitted over the open network. It will therefore draw multiple attacks and must be resilient. Reversible digital watermarking and more precisely the histogram changing variants of reversible digital watermarking were the focus of the literature review. This has contributed to the latest algorithm being further applied based on the histogram shifting algorithm. Through the usage of Quantum steganography, LSB Quantum Watermarking, LWT-QR-MD5 techniques, it is a lossless approach. The degree of robustness is improved when a consistent procedure is used to build the digest of the watermark. The detailed algorithm is discussed in paper [11].

4 Steganalysis

The analysis of proposed method is done in this section. It includes PSNR, Mean Square Error (MSE), Structural Similarity Index Measures (SSIMs), and Correlation.

4.1 PSNR

We have used "Peak-Signal-to-Noise-Ratio" (PSNR) as a quantitative indicator of the deterioration impact induced by the attacks. It is employed to measure and calculate the signal's strength. We used a single picture to be embedded in ten different pictures in order to have a better visual representation. This watermark image's visual decay can be seen as a visual indicator of the impact of attacks on any form of watermark signal. In Table 1 and Fig. 1, there are ten images which are observed with different methods like "Quantum steganography, LSB Quantum Watermarking, LWT-QR-MD5" and last column represents PSNR improvement with % improvement.

Figure 1 represents the Mean Value of PSNR of Quantum-based techniques, and "it was found that Quantum steganography was lower than LSB Quantum Watermarking as well as LWT-QR-MD5. It was found that maximum value for all images with LWT-QR-MD5 was 62.45%, respectively. It was found in Fig. 2 (PSNR results after improvement in absolute terms), [11] the image Sailboat showed maximum PSNR improvement of 12.84 and increased up to 23.7%. The minimum value of image Peppers 3D showed PSNR improvement of −3.09 and decreased up to −5.7%.

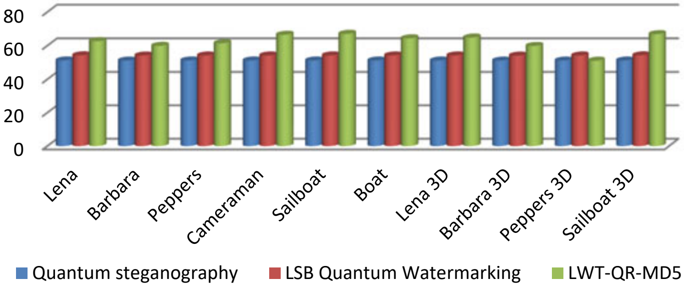


Fig. 1 PSNR graphical presentations

Table 1 PSNR results

Image	Quantum steganography	LSB quantum watermarking	LWT-QR-MD5	PSNR improvement	% improvement
Lena	51.1789	54.24	62.69	8.44	15.6
Barbara	51.0889	54.10	59.94	5.85	10.8
Peppers	51.1549	54.13	61.44	7.31	13.5
Cameraman	51.1576	54.15″	66.47	12.32	22.8
Sailboat	51.1422	54.18	67.01	12.84	23.7
Boat	51.1363	54.14	64.36	10.22	18.9
Lena 3D	51.1358	54.14	64.82218	10.68	19.7
Barbara 3D	51.0989	54.04	59.83811	5.80	10.7
Peppers 3D	51.1173	54.14	51.04390	−3.09	−5.7
Sailboat 3D	51.1528	54.23	66.84407	12.62	23.3
Average	51.13636	54.15	62.44722	8.30	15.3

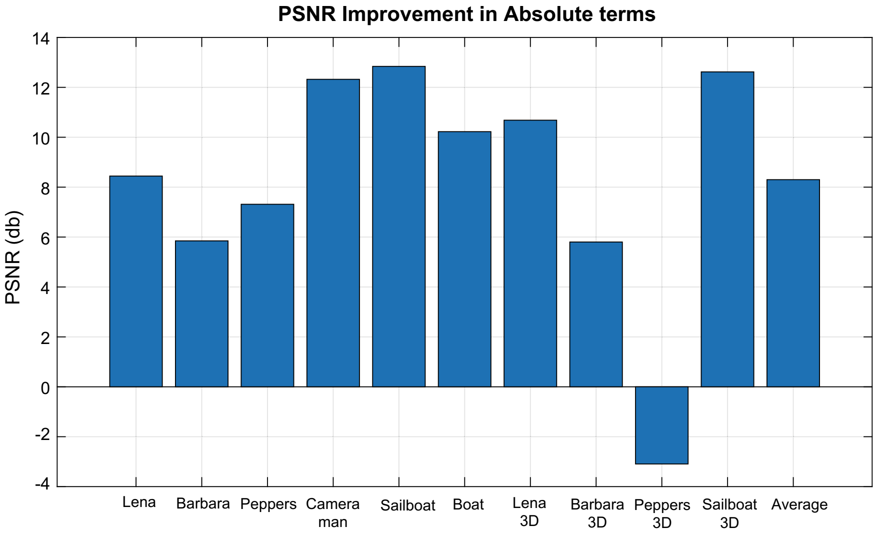


Fig. 2 PSNR results after improvement in absolute terms

4.2 Mean Square Error (MSE)

"Table 2 presents the "Mean Square Error Results after applying proposed algorithm with LSB quantum watermarking, LWT-QR-MD5, MSE improvement, % improvement with ten images" [11]."

Figure 3 represents the graphical presentations of "Mean Square Error Results after applied proposed algorithm LSB quantum watermarking and LWT-QR-MD5. It

Table 2 MSE results after applying proposed method

Image	LSB quantum watermarking	LWT-QR-MD5	MSE improvement	% improvement
Lena	24.68	8.14	16.55	67.02
Barbara	25.50	11.74	13.73	53.91
Peppers	25.32	17.08	8.24	32.52
Cameraman	25.20	4.59	20.61	81.78
Sailboat	25.03	4.25	20.78	83.03
Boat	25.26	6.22	19.04	75.38
Lena 3D	25.26	0.987	24.28	96.10
Barbara 3D	25.85	0.959	24.89	96.29
Peppers 3D	25.26	0.987	24.28	96.10
Sailboat 3D	24.74	0.987	23.76	96.01
Average	25.21	5.60	19.62	77.8

was found that maximum value of image Barbara 3D in LSB quantum watermarking was 25.85" [12], and pepper image showed 17.09 in LWT-QR-MD5. And minimum value of LSB quantum watermarking was 24.69 for Lena image and value of LWT-QR-MD5 Barbara 3D 0.96, respectively. Hence, we have observed that more stable pattern was in LSB quantum watermarking rather than LWT-QR-MD5.

Figure 4 presented the MSE improvements in Absolute terms. We have found that the maximum MSE value of Barbara 3D was 24.89 and it is increased up to 96.29%, The minimum MSE value of Peppers was 8.24 and it is increased up to 32.52%.

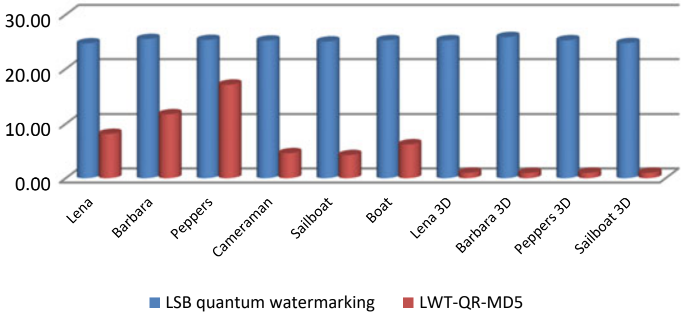


Fig. 3 Graphical presentations of MSE results after applying proposed method

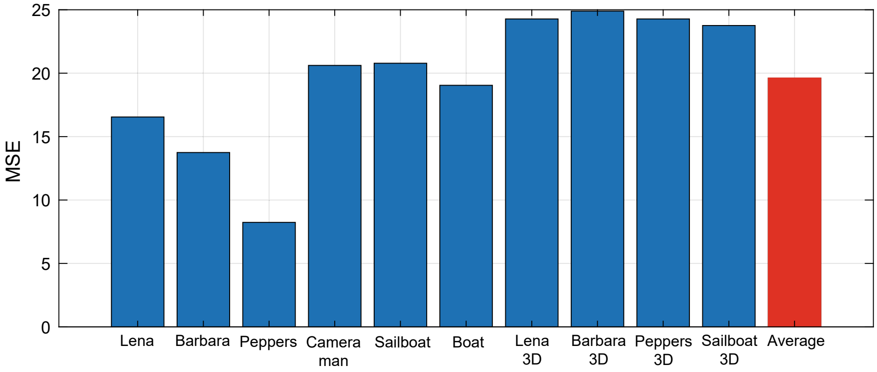


Fig. 4 MSE improvements in absolute terms

4.3 SSIM

Structural Similarity Index Measures (SSIMs) are used to compare two digital images and find the similarity between them. It is reference metric that is it used the original image as reference and compares it with processed image to find the similarities between them. The distinction between these methods and others is that they estimate absolute errors, unlike MSE or PSNR. The concept of structural information holds that pixels have high interdependencies, particularly when they are spatially close to one another. Table 3 presents the Mean Square Error results after applying SSIM improvement. "The ten images were observed under LSB quantum watermarking, LWT-QR-MD5, SSIM improvement, and % improvement. Average value of ten images was found as LSB quantum watermarking: 84.20, LWT-QR-MD5: 94.19, SSIM improvement: 9.99,and it is increased up to 12.0%" [12].

Figure 5 shows the MSE results after applying SSIM improvement and LSB quantum watermarking. It was found that maximum MSE value LSB quantum watermarking was 93.20 for sailboat 3D, LWT-QR-MD5 99.91 for Lena 3D. And minimum MSE value of value LSB quantum watermarking was 81.39 for sailboat 3D, LWT-QR-MD5 88.47 for sailboat, respectively.

Figure 6 represents the SSIM improvement in absolute and percentage term. It was found that the maximum SSIM value of Peppers 3D was 14.39, and it was increased up to 17.65%. And minimum SSIM value of Sailboat 3D was 5.95, and it was increased up to 6.38% respectively.

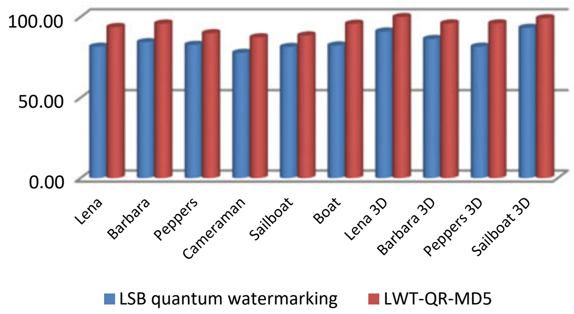
4.4 Correlation

The correlation coefficient measures how similar two signals are. There are two varieties of it: cross-correlation and auto-correlation. Cross-correlation is used to

Table 3 Mean square error results after applied SSIM improvement

Image	LSB quantum watermarking	LWT-QR-MD5	SSIM improvement	% improvement
Lena	81.53	93.71	12.18	14.94
Barbara	84.33	95.83	11.50	13.64
Peppers	82.69	89.88	7.19	8.70
Cameraman	77.75	87.36	9.61	12.36
Sailboat	81.39	88.47	7.08	8.70
Boat	82.31	95.71	13.40	16.28
Lena 3D	90.92	99.91	8.99	9.89
Barbara 3D	86.34	95.94	9.59	11.11
Peppers 3D	81.55	95.94	14.39	17.65
Sailboat 3D	93.20	99.15	5.95	6.38
Average	84.20	94.19	9.99	12.0

Fig. 5 Mean square error results after applied SSIM improvement



compare two different signals, whereas auto-correlation compares the signal to itself. Table 4 shows “correlation results after applied SSIM improvement with Image, LSB quantum watermarking, LWT-QR-MD5, correlation improvement, and % improvement” [12]. Finally, average value of correlation results was found for ten images as LSB quantum watermarking: 92.95, “LWT-QR-MD5”: 99.84, correlation improvement: 6.90 and the average is increased up to 7.5%.

Figure 7 represents the correlation results after applied SSIM improvement with LSB quantum watermarking 92.95, “LWT-QR-MD5”. The maximum value of LSB quantum watermarking is 94.915 for Barbara 3D and LWT-QR-MD5 99.997 for Lena 3D, and this shows more stable figure throughout rather than LSB quantum watermarking. And minimum value of Cameraman was 89.94 and LWT-QR-MD5 99.757 for Barbara, respectively.

Figure 7 represents the Mean Correlation results after applied SSIM improvement quantum-based techniques in absolute and percentage term. The maximum value of Cameraman was shown that correlation improvement is 9.99, and it increases up

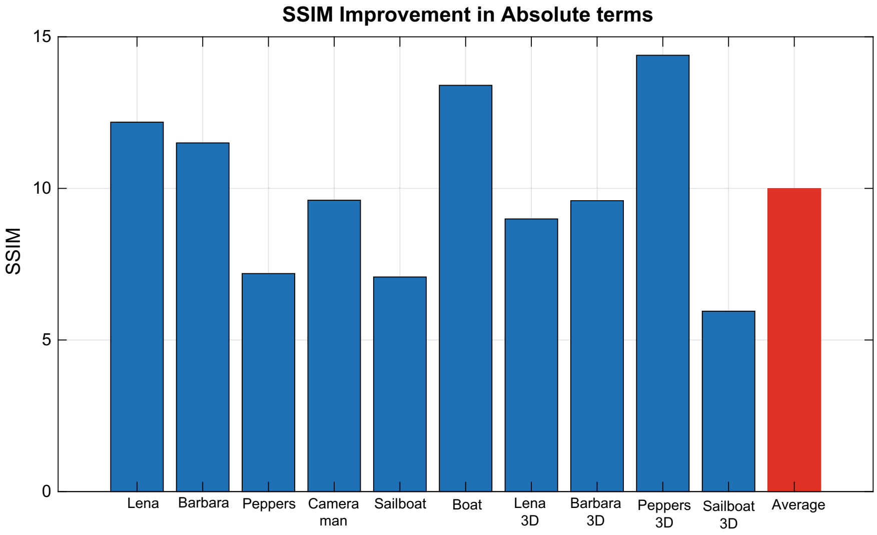
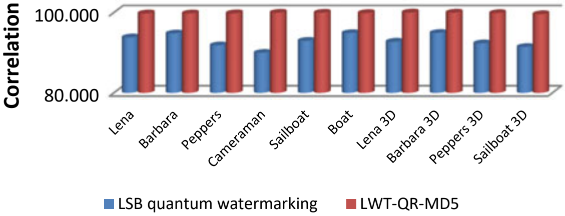


Fig. 6 SSIM improvement

Table 4 Correlation results after applying SSIM improvement

Image	LSB quantum watermarking	LWT-QR-MD5	Correlation improvement	% improvement
Lena	93.775	99.761	5.99	6.38
Barbara	94.769	99.757	4.99	5.26
Peppers	91.806	99.789	7.98	8.70
Cameraman	89.945	99.939	9.99	11.11
Sailboat	92.938	99.933	7.00	7.53
Boat	94.865	99.858	4.99	5.26
Lena 3D	92.721	99.997	7.28	7.85
Barbara 3D	94.915	99.971	5.06	5.33
Peppers 3D	92.316	99.850	7.53	8.16
Sailboat 3D	91.412	99.581	8.17	8.94
Average	92.95	99.84	6.90	7.5

Fig. 7 Correlation results after applied SSIM improvement LSB quantum watermarking



to 11.11%, respectively. And minimum value for Barbara and Boat was shown that correlation improvement is 4.99, and it increases up to 5.26%, respectively.

5 Results and Discussion

Here, three different types of damages to the data have been covered. The study would cover the theoretical background of each attack, its impacts on the watermark signal, and some illustrations of these effects. The attack types under study include rotation attack, additive Gaussian noise, Gaussian smoothing, and crop attack. Each attack has unique characteristics, and they all fall under the category of removal attacks.

5.1 Rotation Attack

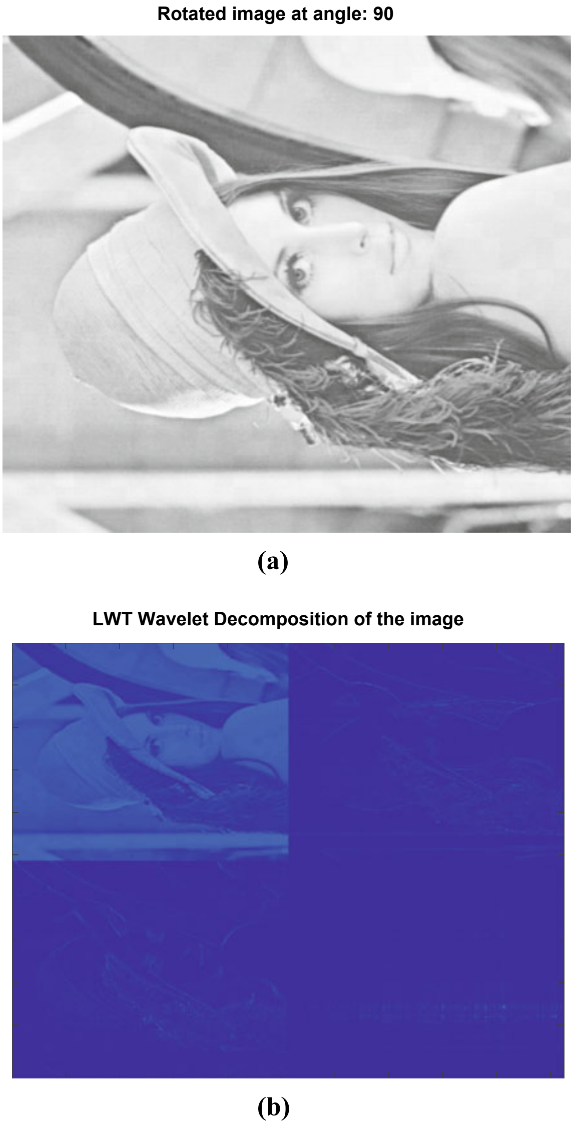
The contour let transform is used to disassemble the original carrier image as shown in Fig. 8a. Sub-blocks are used to split the low-pass section, which can be seen in Fig. 8b. By quantizing the direct current coefficients of these sub-blocks, the watermark bits are incorporated. The radon transform is used to translate the image onto the projection space and transfer the rotation of the original image to a projection transformation. The bi-spectrum measurement technique is then used to estimate the rotation angle of the reference image in order to defend against the rotation attack. This rotation change makes the detecting method robust to rotational attack.

5.2 Gaussian Noise Attack

Gaussian smoothing shares some qualities with other smoothing procedures such as Median Filtering and Gaussian Smoothing Strike. It is a method of image processing that tries to enhance the accuracy of an image by reducing the impact of noise.

5.3 Additive Gaussian Noise

Additive Gaussian noise is a method for purposefully reducing the visual quality of an image by applying a noise signal to it. A Gaussian Probability Density Function follows the statistical properties of this noise (PDF). The most popular probability distribution function for generating random numbers is the Gaussian PDF, often known as the regular PDF, because it is believed to accurately represent a variety of randomly occurring events in daily life. In addition, the central limit theorem asserts



Rotated image at angle: 90

(a)

LWT Wavelet Decomposition of the image

(b)

Fig. 8 **a** Rotation attack at angle 90°. **b** LWT wavelet decomposition of the image after rotation attack

that for a significant number of samples, the distribution of the sample means might frequently resemble the Gaussian distribution. The corresponding PSNR values can be used to quantitatively calculate this phenomenon. In Fig. 9, we can observe that when 0.02% Gaussian noise is added to the image, the same is applicable to the watermark without changing much of its original properties.

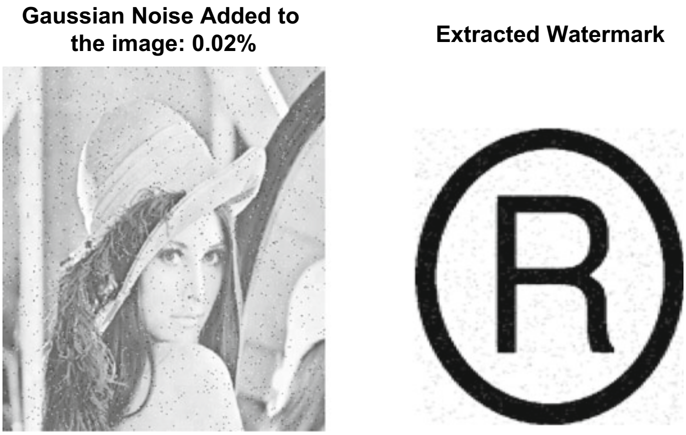
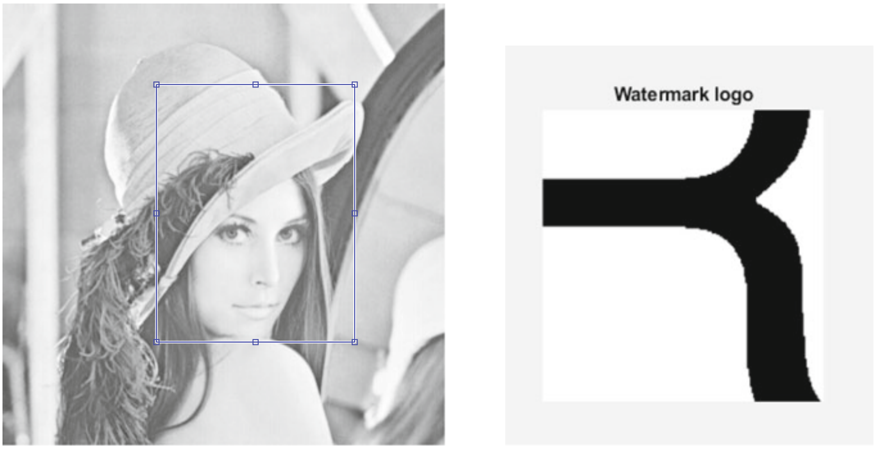


Fig. 9 Gaussian noise added to the image 0.02% with extracted watermark

5.4 Crop Attack

A cropping attack is applicable to perceptible and imperceptible watermarks both. When it is applied to perceptible watermark, the purpose is generally to remove it completely, but when it is applied to imperceptible watermark, then the purpose is majorly to dislocate the watermark. The cropping attack was executed on the 512 × 512 "Lena" image; see the figures below where Fig. 10a is showing the cover image and Fig. 10b is showing the cropped watermarked image.



(a) **(b)**

Fig. 10 **a** Crop attack. **b** Watermark logo

6 Conclusion

The PSNR values vary in a wide range for an enhanced embedding capability relative to the latest suggested methods. In the current process, the PSNR obtained is greater than that achieved in the methods of differential expansion and the value is constant and this is done with a very large potential for embedding. The new algorithm thus outperforms the PSNR and embedding power methods available in the literature. Similarly, the values of MSE and SSIM are most promising with the new method. The watermarked image is able to handle a selected range of attack including rotation and crop attack. In future, this work can be further examined on the larger set of images for extensive comparisons.

References

1. Barni M, Bartolini F, Piva A (2001) Improved wavelet-based watermarking through pixel-wise masking. IEEE Trans Image Process 10(5):783–791
2. Sweldens W (1995) Lifting scheme: a new philosophy in biorthogonal wavelet constructions, In SPIE’s 1995 International symposium on optical science, engineering, and instrumentation. International Society for Optics and Photonics, Sept 1995, pp 68–79
3. Hannoun K, Hamiche H, Lahdir M, Laghrouche M, Kassim S (2018) A novel DWT domain watermarking scheme based on a discrete-time chaotic system. IFAC-PapersOnLine 51(33):50–55
4. Abdulrahman AK, Ozturk S (2019) A novel hybrid DCT and DWT based robust watermarking algorithm for color images. Multimedia Tools Appl 1:1–23
5. Laskar RH, Choudhury M, Chakraborty K, Chakraborty S (2011) A joint DWT-DCT based robust digital watermarking algorithm for ownership verification of digital images, In Computer networks and intelligent computing, Springer, Berlin, Heidelberg, pp 482–491
6. Zhou RG, Luo J (2019) A novel quantum steganography scheme based on ASCII. In: Advances in quantum communication and information. IntechOpen, p 11
7. Xijin W, Linxiu F (2012) The application research of MD5 encryption algorithm in DCT digital watermarking. Phys Procedia 25:1264–1269
8. Taha DB, Taha TB, Al Dabagh NB (2020) A comparison between the performance of DWT and LWT in image watermarking. Bull Electr Eng Inf 9(3):1005–1014
9. Begum M, Uddin MS (2020). Analysis of digital image watermarking techniques through hybrid methods. Adv Multimedia
10. Heidari S, Naseri M (2016) A novel LSB based quantum watermarking. Int J Theor Phys 55(10):4205–4218
11. Sharma G, Kumar V (2020) Preserving IPR using reversible digital watermarking. In: Sharma H, Govindan K, Poonia R, Kumar S, El-Medany W (eds) Advances in computing and intelligent systems. Algorithms for intelligent systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-0222-4_38
12. Sharma G, Kumar V, Chaudhary K (2022) Authentication of digital media using reversible watermarking. In: Khanna K, Estrela VV, Rodrigues JJPC (eds) Cyber security and digital forensics. Lecture notes on data engineering and communications technologies, vol 73. Springer, Singapore. https://doi.org/10.1007/978-981-16-3961-6_6

Analyzing the Effectiveness of Image Augmentation for Soybean Crop and Broadleaf Weed Classification

Michael Justina and M. Thenmozhi

Abstract Data is the key for every artificial intelligence (AI)-based application irrespective of the type of data (numerical, categorical, image) being used. Quality and the depth of information in the data determine the performance of the AI model. Before the data is given as input to the classifier, it must be cleaned using pre-processing techniques. The data must also be sufficient enough to produce satisfactory results. The data considered for this work is images and thus emphasizes image augmentation techniques. This paper focuses on analyzing the best image augmentation techniques for deep learning classifiers. It is essential to analyze the effective data augmentation technique for a particular dataset. In this work, 2382 observations (images) from a crop-weed dataset are used to build the classifier. To expand this dataset, 11 image augmentation methods are applied to the training images. Six out of 11 methods show a high level of effectiveness and are chosen for further process. The outcome of every augmentation method is depicted for an in-depth understanding of augmentation techniques. Sixteen convolutional neural network (CNN)-based pre-trained models are built for evaluating the results. However, MobileNet outperformed other models by resulting in an overall accuracy of 99.58% and F1score of 1.0. Moreover, the performance of the model is evaluated using 24 metrics, and the formulas used for calculation are also tabulated in detail. Tables and graphs are represented for understanding the outcome precisely. Future works in image processing with deep learning are also discussed before concluding.

Keywords Image augmentation · Crop-weed classification · Pre-trained models · AI-based applications · Deep learning · Precision agriculture

M. Justina (✉) · M. Thenmozhi
Department of Computer Science and Engineering, SRMIST, Kattankulathur, Chennai, India
e-mail: jj0170@srmist.edu.in

M. Thenmozhi
e-mail: thenmozm@srmist.edu.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_27

1 Introduction

The essence of artificial intelligence is data. The performance of any AI-based models like machine learning models and deep learning models depends on the data which is fed to the model. The model is framed with respect to the data. Hence, data is a vital part of an AI model irrespective of its application. Data can be categorized as supervised data, unsupervised data, and semi-supervised data. When a label is provided to each and every piece of data, then such data is known as supervised data. When there is no label provided for the data, then these come under unsupervised data. When the data is a mixture of supervised and unsupervised data, then it is known as semi-supervised data. These data are used to predict the output based on available inputs.

There exist several forms of data such as tabular data, text data, audio data, visual data which includes both images and videos, temporal and time series data, network data, geospatial and location data, emotional data, and the data which flows from the internet of things. The forms of data are listed in Table 1 with examples.

Most AI applications such as lung disease identification, handwritten character recognition, and object tracking use images as input to build their model. Raw images are analyzed and pre-processed before being used in a machine learning/deep learning model [1].

The prerequisite for building a successful AI model is to process the image so that it matches the model. Some of the image processing steps that are crucial for machine learning or deep learning models are as follows [2]:

- **Image gathering**: Gathering benchmark images from valid sources.
- **Image acquisition**: Capturing images that are required for the model.
- **Image generator**: Generating synthetic images for the respective application.
- **Image resizing**: Changing the width and height of the images without transformations.
- **Image scaling**: Changing the width and height of the images with transformations.

Table 1 Forms of data with examples

Forms of data	Example
Tabular data	Financial accounting
Text data	Content generation
Audio data	Speech recognition
Visual data	Facial recognition
Temporal data	Stock ticks
Network data	Integrated Data Store (IDS)
Geospatial data	Satellite imagery
Emotional data	Signs of fatigue
Internet of things	Connected cars

- **Image enhancement**: Highlighting the key features of the image through brightening, sharpening, and such.
- **Image restoring**: Improving the quality of the image through noise removal and such.
- **Image augmentation**: Multiplying existing images through transformation techniques.
- **Image annotation**: Defining the objects present in an image through various annotation types.
- **Image labeling**: Naming the images to let the model learn from it.
- **Image normalization**: Normalizing the pixel values to a smaller size (0 to 1 instead of 0 to 255) for faster computation.

In this work, agriculture images are used as input. In particular, crop and weed images are considered as input and are given as input to a deep learning model. Effective data augmentation techniques are applied to the input data to achieve better results.

The rest of the paper is structured as follows: The latest research works done in image processing, deep learning, and metrics for performance evaluation are presented in Sect. 2. Materials and methods used for undertaking this work are discussed in Sect. 3 followed by experimental findings in Sect. 4. Lastly, future improvements and conclusions are discussed in Sect. 5.

2 Related Works

Some of the related research works are briefed here:

2.1 Skin Lesion Detection from Smartphones

Gabriel et al. have developed an application to detect skin lesions during their early stage. A group of dermatologists have been involved in building the application. This application acquires data (images in this case) from smartphones, which are low-cost devices, and predicts the disease. The disease is predicted using a deep learning algorithm, color space combination, and conditional random fields [3].

2.2 Image Augmentation with New Methods

Elgendi et al. have examined the effectiveness of image augmentation using 17 deep learning models in detecting COVID-19. The results are compared with respect to classification accuracy, dataset diversity, augmentation technique, and the size of the network. Implementation without adding the recent geometric augmentation technique reduces Matthew's Correlation Coefficient [4].

2.3 GAN-Based Medical Image Synthesis Methods

Yang et al. have reviewed image generator papers that focus on GAN-based medical image synthesis methods. This article gives a detailed review of the architecture, improvements, and areas of application of GAN. This article also overviews artificial intelligence-based bio-medical analysis, upcoming algorithms for medical image compression, medical image segmentation, and such [5].

2.4 Novel Augmentation Method

Perez et al. have proposed an augmentation method that allows neural networks to learn better the features of the image. This method shows better performance when applied in tiny-imagenet-200 data and MNIST datasets. Augmentation is done by selecting two random images and concatenating them into one. This is given as input to the augmentation net for further process [6].

2.5 SMOTE in Multiclassification

Selukar et al. have solved a multi-classification problem that consists of 13 different crops and weeds. SMOTE is used to balance the unbalanced training dataset. ResNet50 is used as the feature extractor, and logistic regression is used as the classifier to predict the final output. The testing dataset results show a high F1-score of 0.9127 [7].

2.6 Texture, Shape and Color Extraction

Lit et al. have studied weed detection algorithms using machine learning technologies for weed classification. Wheat and corn are distinguished from their weeds by extracting texture, shape, and color feature present in the weeds. This classifier showed a correction rate of 97.5% while describing the grayscale distribution of each pixel and its adjacent pixels [8].

2.7 A Robust Metric—Matthew's Correlation Coefficient

Davide Chicco et al. have studied that Matthew's Correlation Coefficient (MCC) is better than accuracy and F1-score. They have also proved that Matthew's Correlation Coefficient shows the performance of a deep learning model using a single value. For this reason, it is called to be a 'Robust metric' among others. The model's performance is better when the value is near 1. They have also compared the value of MCC against balanced accuracy (BA), bookmaker informedness (BM), and markedness (MK) [9].

2.8 Metrics for Classification Models

Zeljko D. Vujovichas used four metrics and evaluated four different classification models and found the best model. Confusion matrices for the four classification models are compared for evaluating the model's performance. In addition, Type I and Type II error tables are compared, and finally, the best model is selected [10].

2.9 Deep Metric Learning Methods

Xiaoxu et al. have provided an overview of deep metric learning methods for few-shot image classification. The review is taken from 2018 to 2022 and has been grouped into three stages depending on the metric learning. The three stages are learning feature embeddings, learning class representations, and learning distance measures [11].

3 Materials and Methods

3.1 Image Gathering and Image Resizing

This project utilizes an open-source crop-weed dataset in which soybean crops and their associated weeds are present [12, 13]. The original dataset contains four classes namely: broadleaf, grass, soil, and soybean. This work considers soybean (class 0) and broadleaf (class 1) for crop-weed classification which are depicted in Fig. 1. To enhance effective learning, the classes in the dataset are balanced by considering the first 1191 images from the classes. All images are resized to 224 × 224 pixels to meet the requirement of the pre-trained MobileNet classifier [14].

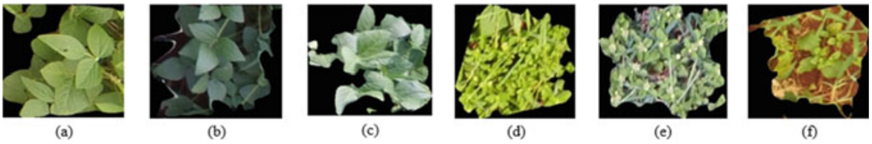


Fig. 1 **a**–**c** are sample observations from segmented soybean crops and **d**–**f** are from segmented broadleaf weeds

3.2 Data Augmentation

The performance of any classifier predominantly depends upon the data. Hence, data augmentation (image augmentation in this case) creates a major impact in improving the overall performance of the classifier. The effectiveness of applying various image augmentation methods varies from one dataset to another [15].

To analyze the most effective image augmentation methods for soybean crop and broadleaf weed dataset, 11 augmentation methods are applied to the randomly chosen images. The outcome of various augmentation methods is compared with the original image to identify the differences. Six out of 11 augmentation methods are observed to be highly effective for this dataset.

Table 2 shows the augmented images of Fig. 1a during different epochs, their assessed level of effectiveness(High/Medium/Low), and the cause for this effectiveness. Augmentation methods with a high level of effectiveness are considered for further processing; however, methods with low/medium levels of effectiveness are not considered in building the classifier.

45° Rotation—Method 1 Rotation is one of the most commonly used image augmentation techniques which randomly rotates the given image and creates a variant of the same. The degree to which the image is to be rotated can be any value between 0 and 360. In method 1, value 45 is chosen to rotate the training images. The results (augmented images) are observed after rotation and three samples are tabulated in Table 2 for reference. The augmented images show only a small difference in the images when it is rotated 45°, and hence, the effectiveness is noted as ‘medium.’

180° Rotation—Method 2 In method 2, value 180 has been chosen to rotate the training images. From the sample images, it is clear that a rotation of 180° produces a better augmentation than a rotation of 45°. The effectiveness of this method is noted as ‘high’ and is considered for model training.

Horizontal flip—Method 3 In this method, the input image is flipped along the horizontal axis. Since the input image is a crop, horizontal flipping has a good effect on the original image. Hence, the effectiveness of this method is noted as ‘high’ and is considered for model training.

Vertical flip—Method 4 In this method, the input image is flipped along the vertical axis. The input image shows no effect on the original image and hence the effectiveness is noted as ‘low.’

30% Width shift—Method 5 Shifting is done to place the object at the center in case it is misplaced. In this method, the pixels are shifted horizontally. 30 percent of the entire image is shifted during this augmentation method. Shifting 30 percent shows only a small difference and hence the effectiveness is noted as 'medium.'

50% Width shift—Method 6 In this method, 50 percent of the pixels are shifted horizontally. This augmentation method produces a good variance from the original image. Hence, the effectiveness is noted as 'high' and is considered for model training.

30% Height shift—Method 7 In this method, the pixels are shifted vertically. 30 percent of the entire image is shifted during this augmentation method. Shifting 30 percent shows only a small difference and hence the effectiveness is noted as 'medium.'

50% Height shift—Method 8 In this method, 50 percent of the pixels are shifted vertically. This augmentation method produces a good variance from the original image. Hence, the effectiveness is noted as 'high' and is considered for model training.

25% zoom—Method 9 In the zoom augmentation method, the original image is either zoomed inside (value < 1) or zoomed outside (value > 1) randomly to generate the augmented image. In this method, a value of 0.25 is chosen to zoom inside the original image, and the results are satisfactory as the variants are unique. Therefore, the effectiveness is noted as 'high' and is considered for model training.

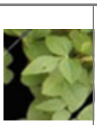
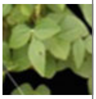
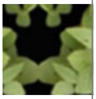
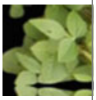
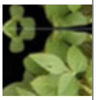
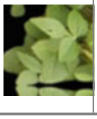
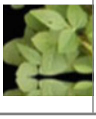
45% zoom—Method 10 In this method, a value of 0.45 is chosen to zoom inside the original image and the results are not satisfactory as the image is zoomed in-depth. This makes the image lose its shape feature. Therefore, the effectiveness is noted as 'low.'

Brightness range (0.1–0.9)—Method 11 This method brightens the image to produce a variant of the original image. A brightness value closest to 0 produces a dull image. In contrast, a brightness value closest to 1 produces an image with maximum brightness. In this method, the range of brightness is chosen to be between 0.1 and 0.9 for random value selection, and the results are satisfactory. Therefore, the effectiveness is noted as 'high' and is considered for model training.

3.3 Image Normalization

After augmentation, all images are scaled between 0 and 1 with a factor of 1/255. This factor value is chosen as pixel value ranges between 0 and 255 [16].

Table 2 Choosing the effective data augmentation techniques

Methods	Various augmented images				Effectiveness	Cause
45° rotation					Medium	No significant difference as the rotation is minimum
180° rotation					High	Better results as rotation is with higher degree
Horizontal flip					High	Change of the crop co-ordinates
Vertical flip					Low	Augmented images are same as the original image
30% Width shift					Medium	No changes in crop co-ordinates
50% Width shift					High	Crop co-ordinates changes
30% Height shift					Medium	No changes in crop co-ordinates
50% Height shift					High	Crop co-ordinates changes

(continued)

Table 2 (continued)

Methods	Various augmented images				Effectiveness	Cause
25% Zoom					High	Zooming leaves the shape of the leaf unchanged
45% Zoom					Low	Morphological features are changed, which may mislead the model
Brightness range (0.1, 0.9)					High	Crops under different lighting conditions are obtained

[a] Medium and low effective methods are not considered for building classifiers

3.4 *Deep Learning Classifier*

VGG16, VGG19, MobileNet, MobileNetV2, ResNet50, ResNet50V2, ResNet150, ResNet150V2, ResNet150, ResNet150, DenseNet201, DenseNet169, DenseNet101, InceptionV3, and InceptionResNet are used for classification. Among which MobileNet, a pre-trained CNN classifier outperformed the other 15 classifiers. Results of the 16 classifiers are detailed in [17]. While training the classifier, the crop and weed features are extracted with which the images are either classified as a crop or as a weed. MobileNet is implemented by preserving all layer structures except for the trainable layer which is on the top [18].

4 Experimental Results

The classifier's outputs are analyzed, and their performances are evaluated using 24 metrics which are listed in Table 4. The evaluation metrics and their equations are illustrated in Table 3.

The model is trained for six epochs and showed the testing accuracy as 0.9958, precision as 1.0, recall as 0.99, F1-score as 1.0, and specificity as 1.0. The values of 24 metrics are given in Table 4.

Figure 2 shows the confusion matrix, and Fig. 3 shows the receiver operating characteristic (ROC) curve and precision-recall (PR) curve of the MobileNet classifier. The classifier has produced 118 TP and 119 TN. Otherwise stated the model has predicted one soybean crop image incorrectly as a broadleaf weed; however, no broadleaf weed had been incorrectly predicted as a soybean crop [19].

Table 3 Description of evaluation metrics for soybean crop and broadleaf weed

Equations	Eq. no.
True Positives (TP) $=$ Number of soybean crops predicted as soybean crops	(1)
True Negatives (TN) $=$ Number of broadleaf weeds predicted as broadleaf weeds	(2)
False Positives (FP) $=$ Number of broadleaf weeds predicted as soybean crops	(3)
False Negatives (FN) $=$ Number of soybean crops predicted as broadleaf weeds	(4)
Accuracy $= \frac{\text{Number of crops predicted as crops (TP) and weeds predicted as weeds (TN)}}{\text{Total number of predictions (TP+TN+FP+FN)}}$	(5)
Error rate $= 1 -$ Accuracy	(6)
$L_{\text{BCE}} = -\frac{1}{N}\sum_{i=0}^{N} y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)$	(7)
True Positive Rate (TPR) $= \frac{\text{Number of crops predicted as crops (TP)}}{\text{Number of crop observations in the dataset } (P=\text{TP+FN})}$	(8)
True Negative Rate (TNR) $= \frac{\text{Number of weeds predicted as weeds (TN)}}{\text{Number of weed observations in the dataset } (N=\text{TN+FP})}$	(9)
False Positive Rate (FPR) $= \frac{\text{Number of weeds predicted as crops (FP)}}{\text{Number of weed observations in the dataset } (N=\text{FP+TN})}$	(10)
False Negative Rate (FNR) $= \frac{\text{Number of crops predicted as weeds } (FN)}{\text{Number of crop observations in the dataset } (P=\text{FN+TP})}$	(11)
Positive Predictive Value (PPV) $= \frac{\text{Number of crops predicted as crops (TP)}}{\text{Number of observations predicted as crops (TP+FP)}}$	(12)
Negative Predicted Value (NPV) $= \frac{\text{Number of weeds predicted as weeds (TN)}}{\text{Number of observation predicted as weed (TN+FN)}}$	(13)
False Discovery Rate (FDR) $= \frac{\text{Number of weeds predicted as crops (FP)}}{\text{Number of observations predicted as crops (FP+TP)}}$	(14)
$F1$ score $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	(15)
MCC $= \frac{n1 - n2}{\sqrt{d1 \times d2 \times d3 \times d4}}$ where, $n1 = \text{TP} \times \text{TN}$, $n2 = \text{FP} \times \text{FN}$ $d1 = \text{TP} + \text{FP}$, $d2 = \text{TP} + \text{FN}$, $d3 = \text{TN} \times \text{FP}$, $d4 = \text{TN} \times \text{FN}$	(16)

Figure 4 shows the zoomed-in curve of ROC and PR for the MobileNet classifier. These curves clearly show that the performance of the MobileNet classifier is high.

5 Conclusion and Future Enhancements

Artificial intelligence transforms tons of applications from various sectors. Data is key to building a successful AI application. This input data must undergo all necessary techniques with respect to its field. As data differs from sector to sector, the data pre-processing techniques also differ. Any AI application cannot be successful without data pre-processing.

In this work, the data considered are images which had undergone all necessary image processing techniques. The dataset considered for this work has 2382 images. Image resizing, 11 state-of-the-art augmentation techniques, and image normalization are done. The pre-processed images are given to the pre-trained MobileNet classifier. The classification accuracy obtained is 99.58% and the F1-score is 1.0.

Even though most of the deep learning algorithms use images as input, these images differ in a wide range from one sector to another. Application-specific image processing techniques are the need of the hour. It will be of great use to the upcoming

Table 4 Results of MobileNet classifier

Metrics	Values
Training accuracy	0.9848
Training error	0.0152
Training loss	0.1215
Validation accuracy	0.9916
Validation error	0.0084
Validation loss	0.1088
Testing accuracy	0.9958
Testing error	0.0042
Testing loss	0.043
Epochs	6
Testing time	0.08
True Positive (TP)	118
True Negative (TN)	119
False Positive (FP)	0
False Negative (FN)	1
Positive Predictive Value (PPV)	1
True Positive Rate (TPR)	0.99
F1-score	1
True Negative Rate (TNR)	1
False Positive Rate (FPR)	0
False Negative Rate (FNR)	0.01
Negative Predictive Value (NPV)	0.99
False Discovery Rate (FDR)	0
Matthew's Correlation Coefficient (MCC)	0.99

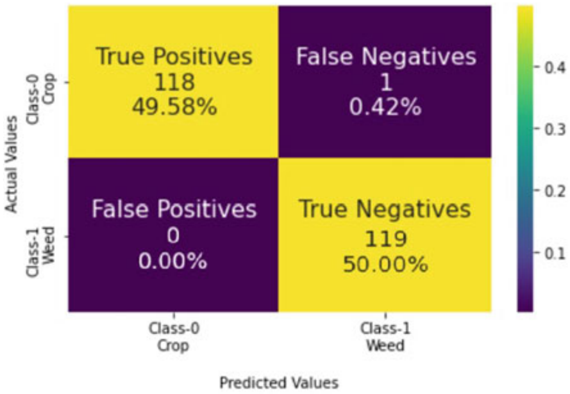


Fig. 2 Confusion matrix of the MobileNet classifier

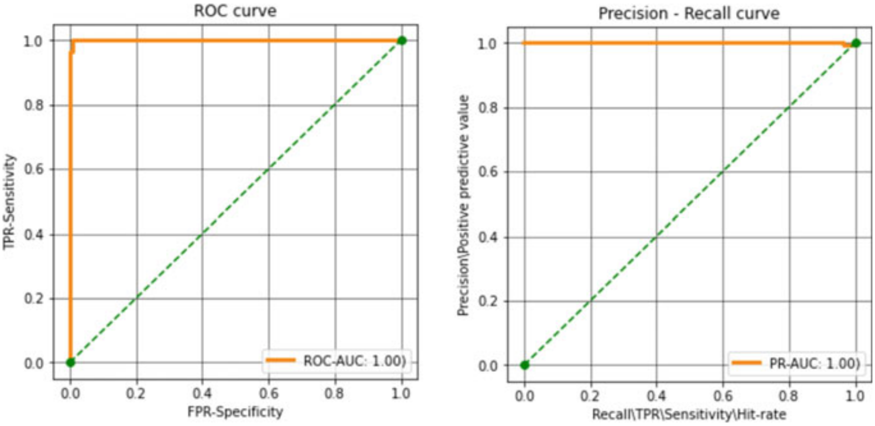
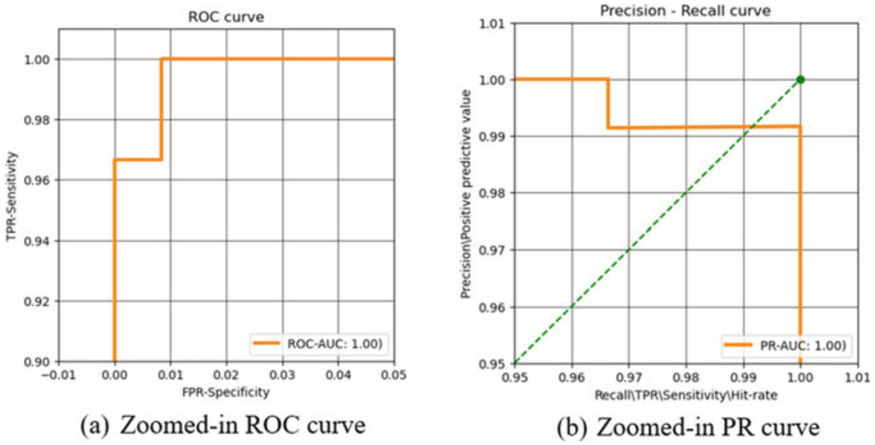


Fig. 3 ROC and PR curve of the MobileNet classifier



(a) Zoomed-in ROC curve

(b) Zoomed-in PR curve

Fig. 4 Zoomed-in ROC and PR curve of the MobileNet classifier

researchers if such specific techniques are designed. Eventually, this will result in the improved performance of the classifier.

References

1. Crawford K, Paglen T (2021) Excavating AI: the politics of images in machine learning training sets. Ai Soc 36(4):1105–1116
2. Jiao L, Zhao J (2019) A survey on the new generation of deep learning in image processing. IEEE Access 7:172231–172263
3. De Angelo GG, Pacheco AGC, Krohling RA (2019) Skin lesion segmentation using deep learning for images acquired from smartphones. In: 2019 International joint conference on neural networks (IJCNN). IEEE
4. Elgendi M et al (2021) The effectiveness of image augmentation in deep learning networks for detecting COVID-19: a geometric transformation perspective. Front Med 8:629134
5. Yang H, Qian P (2023) GAN-based medical images synthesis: a review. In: Research anthology on improving medical imaging techniques for analysis and intervention, pp 1539–1546
6. Perez L, Wang J (2017) The effectiveness of data augmentation in image classification using deep learning. arXiv:1712.04621
7. Selukar M, Jain P, Kumar T (2022) A device for effective weed removal for smart agriculture using convolutional neural network. Int J Syst Assur Eng Manag 1–8
8. Liu Y (2022) Field weed recognition algorithm based on machine learning. J Electron Imaging 31(5):051413
9. Chicco D, Tötsch N, Jurman G (2021) The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Min 14(1):1–22
10. Vujovic Z (2021) Classification model evaluation metrics. Int J Adv Comput Sci Appl 12(6):599–606
11. Li X et al (2023) Deep metric learning for few-shot image classification: a review of recent developments. Pattern Recogn 109381
12. dos Santos Ferreira A et al (2017) Weed detection in soybean crops using ConvNets. Comput Electron Agric 143:314–324
13. https://data.mendeley.com/datasets/3fmjm7ncc6/1
14. Han Z (2019) Computer vision-based agriculture engineering. CRC Press
15. Xu M et al (2023) A comprehensive survey of image augmentation techniques for deep learning. Pattern Recogn 109347
16. Pei S-C, Lin C-N (1995) Image normalization for pattern recognition. Image Vis Comput 13(10):711–723
17. Justina Michael J, Thenmozhi M, Evaluation of deep learning CNN models with 24 metrics using soybean crop and broad-leaf weed classification. In: International conference on information, communication and computing technology. Sin
18. Howard AG et al (2023) MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 (2017). Springer, Singapore
19. Sammut C, Webb GI (eds) (2011) Encyclopedia of machine learning. Springer

Iterative Thresholding-Based Shadow Detection Approach for UAV Images

Deeksha and Toshanlal Meenpal

Abstract Shadow detection is a critical task in computer vision and image processing that aims to identify shadow regions in an image. Accurate detection of the shadow is essential for various applications, such as object recognition, scene understanding, and image segmentation. The detection of shadow is difficult due to their complex and dynamic nature, as they can vary in shape, size, and intensity depending on the location of the illumination source, weather conditions, and the characteristics of the scene. In this study, a new shadow detection method has been proposed that automatically calculates the threshold value using an iterative thresholding scheme and detects shadow. The performance of the developed method is tested on four publicly available UAV image datasets related to two study areas namely urban and mining areas. The comparison of the proposed method with several state-of-the-art methods demonstrates that the proposed method performs well in both qualitative and quantitative evaluations, with good overall accuracy in all images.

Keywords Iterative thresholding · UAV · Shadow detection

1 Introduction

Nowadays, high-resolution aerial images play a vital role in the remote sensing environment as they can capture very complex details of objects on the ground across various fields including land management, marine water resources management, disaster monitoring, agricultural applications, national security, etc. [1]. Due to technological advancements, drones, or small unmanned aircraft, have undergone remarkable growth, transitioning from being mere toys to becoming valuable

Deeksha (✉) · T. Meenpal
Department of Electronics and Communication, National Institute of Technology Raipur,
Raipur, India
e-mail: deeksha.phd2022.etc@nitrr.ac.in

T. Meenpal
e-mail: tmeenpal.etc@nitrr.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_28

tools that address numerous challenges in different industries. Aerial imaging-based remote sensing solutions have been extended to the use of drones, which have facilitated various applications, such as urban, forestry, and mining analyses, precision agriculture optimization, geosciences research, and urban zone segmentation. UAVs have effectively bridged the gap between field observations and remote sensing by offering high spatial detail over extensive areas, all while remaining cost-effective.

Despite the increasing use of aerial images, several factors such as background clutter, occlusion, illumination, and shadow impede the extraction of information. UAVs, that are equipped with a light camera have the potential to capture images including pixels of targets that are illuminated either directly by the sun or indirectly by diffused light in shadowed regions. Targets like rocks, land elevations, machinery, trees, buildings, towers, poles, etc., are assumed to have the potential to cast shadow in urban and mining fields. Shadow poses a significant challenge to image processing and analysis tasks, leading to detection inaccuracies in various image vision applications, including object detection and tracking. Since shaded area reduces the quality of remote sensing, identifying and eliminating shadow is considered crucial preprocessing steps in numerous aerial image processing techniques [2]. Although some researchers view shadow as obstacles, others have utilized them in applications like object height estimation through UAV images [3]. However, The issue of shadow detection in high-resolution UAV images remains unresolved, and more efforts are necessary to improve it. A shadow is the shape of an object that appears dark due to the object obstructing the path of light. Cast shadow and self-shadow are the two major types of shadows that appear in UAV images [4]. Based on the complexity of shaded regions, they can be also divided into two regions; the umbra shadow region and the penumbra shadow region as shown in Fig. 1. Due to the fact that UAV images are captured from different heights and at different times, the generated high-resolution images contain shadows of varying sizes and degrees of brightness. Hence, developing algorithms for shadow detection in this type of image is more challenging. Nevertheless, the presence of shadow can impede the accuracy of these algorithms since they can distort the apparent shape of objects, as explained earlier.

This paper presents a novel technique for the detection of shadow from UAV images taken from mining and urban areas. The proposed algorithm uses an additional saturation (S) channel mask for addressing the misclassification of concrete regions. An iterative thresholding scheme and mean of saturation and value channel are utilized for calculating threshold value for creating a shadow mask of an image. The proposed approach offers several contributions, including the use of iterative thresholding with saturation and value channel images for automating the threshold calculation in UAV images. The proposed method is tested on four different publicly available UAV-taken image datasets related to mining and urban areas. A comparative analysis between different UAV-captured images has been performed regarding IoU, F1-score, recall, precision, and accuracy. The experimental outcomes inferred that the presented technique is adaptive and can be applied to both mining as well as urban images.

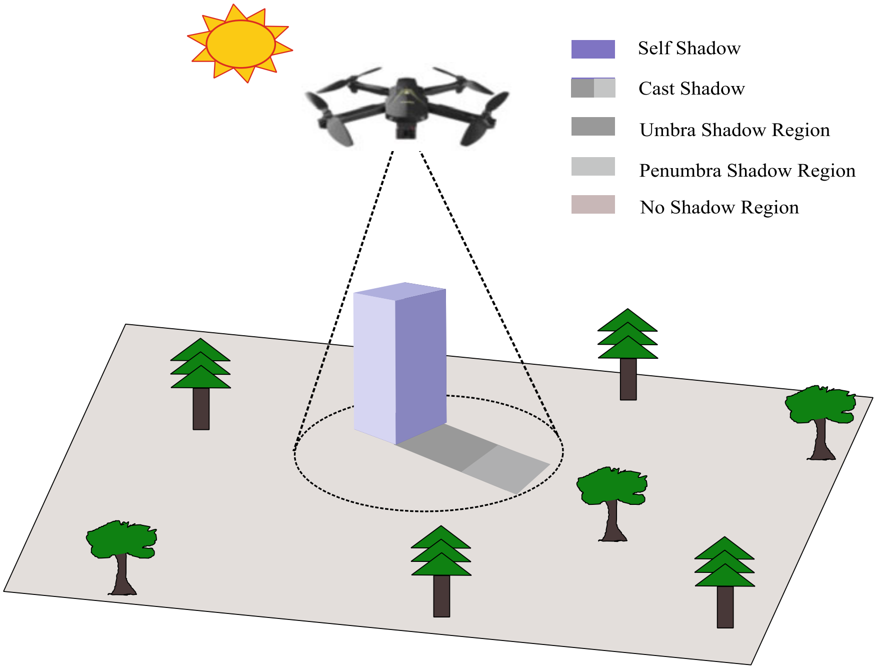


Fig. 1 UAV image capturing process resulting with different types of shadows

2 Related Work

Numerous research papers have been already published to address the problem of shadow detection. This section presents a review of various techniques of shadow detection approaches.

2.1 Property-Based Shadow Detection

The method of property-based shadow detection involves utilizing the properties of both the shadow image and its surroundings to identify the presence of a shadow in an image. Thresholding, invariant color model, and object segmentation are commonly used techniques for this type of detection, but these techniques demand additional exertion from researchers to assess the appropriateness of various characteristics and are susceptible to mistakes arising from technical glitches. Several shadow properties are mentioned already in the literature that is listed here:

1. The blocking of electromagnetic radiation from the sun results in a diminished brightness or intensity known as low luminance [5].

2. High saturation with short blue-violet wavelengths as a result of atmospheric scattering through the Rayleigh effect [6].
3. According to reference [7], the color spectrum wavelength values in shaded areas cause variations in intensity that result in higher hue values.

A considerable number of property-based shadow detection methods have been already presented by researchers such as Tsai et al. proposed a smoothed Otsu threshold-based shadow detection method where the threshold is computed by calculating the ratio of luminance and hue channels [8]. His approach involves creating a shadow mask using a technique that uses a smoothed Otsu threshold on an image transformation. In 2016, Anoopa S. et al. proposed a tri-class thresholding technique-based segmentation approach [9]. In their work, they utilize the Otsu method only on the undetermined region for segmentation in an iterative manner. Das and Shandilya proposed a novel technique that uses an automatic threshold mechanism to accurately differentiate between foreground and background pixels by detecting edges [10]. Besides this, Gilberto Alvarado et al. proposed a multi-channel statistics-based shadow detection approach from aerial images [11]. In that paper, an additional low saturation mask is used for improving the accuracy of the shadow detection algorithm.

2.2 Model-Based Shadow Detection

In the model-based method for detecting shadows in images captured by UAVs, factors like the position of the light source, the shape of objects, and the location of the sensor are considered to account for the physical and environmental characteristics of the shadows. Pons et al. used spectral reflectance measurements via the digital surface model (DSM) to identify shadows in images captured by a UAV in forested areas [12]. Wang et al. presented a geometrical-based technique for detecting shadows in very high-resolution images by utilizing the position of the sun to map shadow and non-shadow regions [13].

2.3 Machine Learning-Based Shadow Detection

The machine learning approach to shadow detection requires less information about the shadows and offers greater flexibility than the previously mentioned methods. This technique involves building a model and then fitting the data into it. The unsupervised machine learning approach commonly uses clustering techniques such as K-means clustering to group similar pixels into clusters for the binarization of an image containing shadow [14]. In 2018, Silva et al. proposed CIELCh color space channels-based shadow detection algorithm [15]. In the supervised learning-based shadow detection method, a training data sample is required to make a classifier for accurate classification. Several supervised classifiers such as convolutional neu-

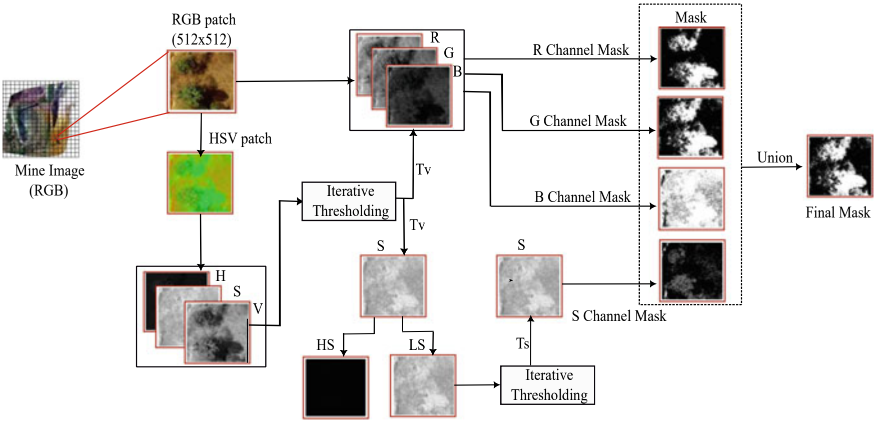


Fig. 2 Proposed shadow detection methodology

ral networks (CNN), support vector machines (SVM), and their variants such as LSSVM are used to generate an efficient and accurate binary mask and classification in shadow and non-shadow class. In 2017, Kang et al. proposed an extended random walker-based approach for detecting shadows from very high-resolution images captured remotely [16]. Currently, many deep learning approaches, such as UNET and PSPNET, are being used for shadow detection in UAV images. Author Luo S. utilized the AISD dataset and presented a network that incorporates a parallel spatial pyramid structure for extracting features at multiple scales from the input image [17]. In 2021, Siti Asiyah et al. introduced U-NET architecture for creating shadow masks from UAV images [18]. They also employed morphological operations in the post-processing step for refining the created masks.

3 Proposed Methodology

Shadow detection using shadow properties is accomplished through semantic segmentation, specifically binary segmentation [19]. This involves assigning image pixels into two categories—shadow and non-shadow pixels, resulting in binary masks where shadow pixels are represented by white labels and non-shadow pixels by black labels. The workflow for the proposed method is depicted in Fig. 2. The proposed approach is divided into three parts, namely threshold value calculation using an iterative thresholding scheme, mask generation, and in the final part union operation of all the masks.

3.1 Threshold Calculation

An iterative thresholding scheme illustrated in Algorithm 1 is used for calculating threshold values. In this scheme, firstly the mean of the input image is calculated and a pixel value greater than this mean is stored in the $S1$ group and a pixel value less than the mean value is stored in the $S2$ group. After that, mean of both $S1$ and $S2$ groups is calculated individually, and their average is taken as the new threshold value. This process is continued until the two consecutive threshold values become equal.

Algorithm 1 Iterative Thresholding Algorithm

```
Input: Image $I_s$
Output: Threshold Value $T$
  $T$ = Any random value between 0 to 255
  $T_{new}$ = Mean value of $I_s$
while ($T \neq T_{new}$) do
    $T \leftarrow T_{new}$
    for all pixels of $I_s$ do
        if (Pixel value of $I_s \geq T_{new}$) then
            $S1 \leftarrow$ Pixel value of $I_s$
        else
            $S2 \leftarrow$ Pixel value of $I_s$
        end if
    end for
end while
$m_1 =$ Mean value of $S1$
$m_2 =$ Mean value of $S2$
$T_{new} = \frac{(m_1 + m_2)}{2}$
  return T
```

3.2 Mask Generation

The development of the presented methodology is based on the properties of shadow as discussed earlier in Section II. The key feature of shadow that is utilized in the presented method is that the pixel value in the saturation channel for the shadow region is high as compared to the non-shadow region. Thus, along with R-, G-, and B-channel masks, an additional saturation channel mask is also incorporated to address the misclassification of the concrete region. Hence, in the first step, the input RGB image is converted into an HSV image, and then S and V channel image is employed for the computation of the threshold values T_{v}, and T_{s}, respectively, using an iterative threshold scheme. The value channel threshold (T_{v}) is used to generate R-channel, G-channel, and B-channel shadow masks which are defined as follows:

$$M_{\mathrm{R}}(x, y) = \begin{cases} 255 & \text{if } R(x, y) < T_{\mathrm{v}}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

$$M_{\mathrm{G}}(x, y) = \begin{cases} 255 & \text{if } G(x, y) < T_{\mathrm{v}}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$M_{\mathrm{B}}(x, y) = \begin{cases} 255 & \text{if } B(x, y) < T_{\mathrm{v}}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

T_{v} is also applied in saturation channel image to divide saturation channel image pixels into low saturation (LS) and high saturation (HS) pixels as defined in Eqs. 4 and 5.

$$\mathrm{HS}(x, y) = \begin{cases} S(x, y) & \text{if } S(x, y) > T_{\mathrm{v}}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

$$\mathrm{LS}(x, y) = \begin{cases} S(x, y) & \text{if } S(x, y) < T_{\mathrm{v}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

As mentioned earlier, the shadow region has a high saturation value so HS pixels are discarded and only LS pixels are used to calculate the saturation channel threshold (T_{s}) value. A saturation channel mask is then prepared by applying T_{s} on the saturation channel image as given in Eq. 6.

$$M_{\mathrm{S}}(x, y) = \begin{cases} 255 & \text{if } S(x, y) < T_{\mathrm{s}}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

3.3 Union Operation

After binarization, four masks are generated namely the red channel mask, green channel mask, blue channel mask, and saturation channel mask. These masks contain different parts of shadows. To obtain all the parts into a single image, union operation of all the channel masks has been performed.

4 Experiment and Results

4.1 Test Datasets

Our proposed methodology is tested on four publicly available UAV image datasets related to mining and urban area that are described below.

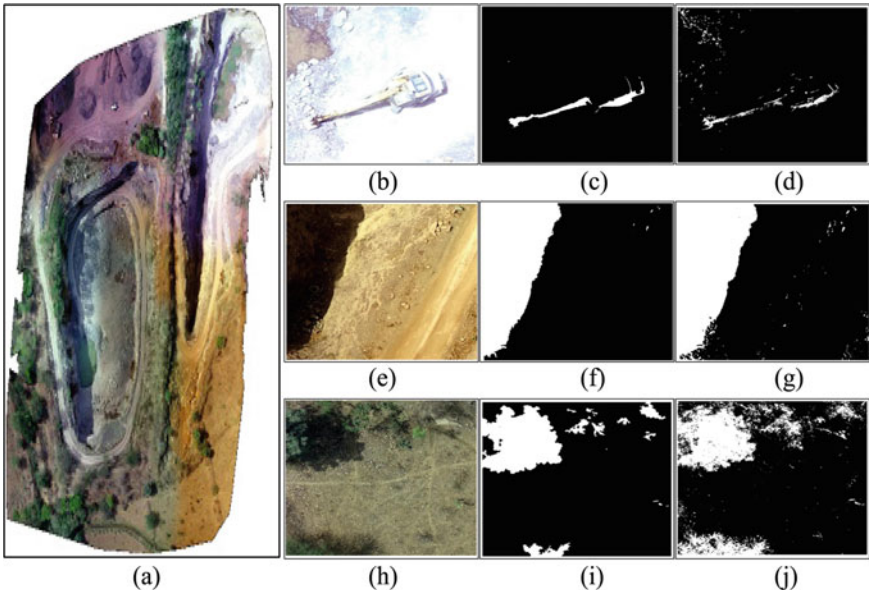


Fig. 3 **a** Sample image of Maniram Patel Mine Dataset, **b**, **e**, **f** Patch 1, Patch 2, Patch 3, **c**, **f**, **i** Corresponding reference shadow mask, and **d**, **g**, **h** Corresponding detected shadow region

Mining area-related image dataset: Two mining area-related image datasets are used in this study for testing. One mining image dataset is collected from the Geology Department at NIT Raipur. One sample image of this dataset is shown in Fig. 3a. The dimension and size of this high-resolution image are 6751 × 7690 pixels and 7.91 MB, respectively. This image was acquired from open-cast Maniram Patel Mine using a camera onboard a flying UAV. The second mining dataset used in this study is downloaded from the Drone Mapper site, and one sample image is depicted in Fig. 4a [20]. This dataset contains 45 high-resolution oblique images of size 3000 × 4000, and these images are taken from Red Rock Colorado mine. The primary sources of shadow in these images are cast by rocks, poles, small buildings, machines, vehicles, trees, etc.

Urban area-related image dataset: For comparative analysis in this work, two publicly available UAV images related to urban areas namely Mendeley Thermal and Visible Aerial Imagery Dataset and Aerial Semantic Segmentation Dataset are utilized as test image datasets. Mendeley Thermal and Visible Aerial Imagery Dataset contains 30 visible images and the dimensions and sizes of each image are 4000 × 3000 and 4.87 MB, respectively [21]. The shadow brightness of these images is high. One sample image of this dataset is shown in Fig. 4d. Kaggle's Aerial Semantic Segmentation Dataset is another source of image datasets related to the urban field [22]. Unlike the previous dataset, the images in this dataset were captured from lower altitudes, making the shadows of objects more noticeable and more prominent. There

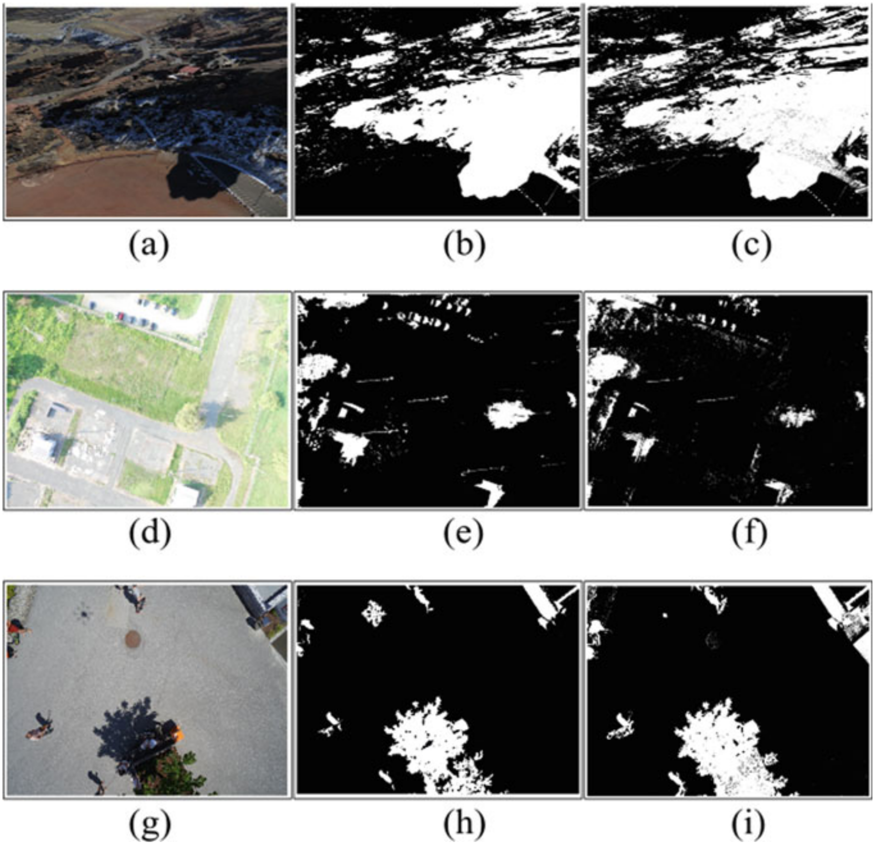


Fig. 4 UAV-captured test images, **a** sample image of Red Rock Colorado Mine Dataset, **d** sample image of Thermal and Visible Aerial Imagery Dataset, **g** sample image of Aerial Semantic Segmentation Dataset, **b**, **e**, **h** Corresponding reference shadow mask, and **c**, **f**, **i** Corresponding detected shadow region

are 598 images of dimension 6000 × 4000 in JPG format. Among them, we have taken 30 images for testing the proposed approach. Figure 4g displays one sample image from this dataset. The shadowed areas in these images are primarily caused by trees, poles, buildings, people, etc.

4.2 Results

In order to evaluate the performance of the presented shadow detection technique, the predicted shadow masks obtained from the proposed methodology are compared to the corresponding reference images. As the test images utilized in this study were not

Table 1 Comparison of the shadow detection results for all testing image datasets

Parameter	Mining image datasets		Urban image datasets	
	Maniram Patel Mine Dataset	Red Rock Colorado Mine Dataset	Thermal and visible aerial imagery dataset	Aerial semantic segmentation dataset
IoU	**0.864**	0.687	0.618	0.702
Precision	**0.805**	0.694	0.702	0.737
Recall	**0.904**	0.983	0.705	0.937
F1-Score	**0.863**	0.814	0.752	0.825
OA	**0.968**	0.943	0.938	0.956

intended earlier for shadow detection evaluation so, there were no existing reference shadow masks. Consequently, manual labeling of shadow masks is performed by annotating shadowed areas using Paint 3D software. During labeling, pixels within shadowed regions are given a value of 255 (white), whereas pixels outside shadowed regions are given a value of 0 (black).

Qualitative Comparison: The size of the open-cast mine image taken from the Maniram Patel Mine Dataset is large, so processing directly this high-resolution image is difficult. Hence for reducing the annotation error and for simplifying the analysis process, the whole images are cropped into small patches of size 512 × 512 pixels. Some patches along with their reference images and shadow detection results are shown in Fig. 3. Figure 4 shows the results of the proposed shadow detection approach for different UAV images along with their reference images. The results demonstrate that the presented shadow detection approach performs well on all of the selected test images and produces outcomes that are closest to their reference images. The method is capable of accurately detecting shadows, regardless of whether they are large or small. In addition, there are variations in the statistical distribution of color saturation and brightness in urban area images compared to mining area images, particularly with regard to saturation values. The color of shadows in such images is not consistent, and the texture of the regions is more noticeable in urban images than in mining field images, as shown in the cases presented in Fig. 4.

Quantitative Comparison: Five different performance metrics are employed to assess the proposed method's accuracy. Table 1 displays the outcomes obtained from four UAV image datasets, indicating that the presented approach yields good detection accuracy for all images. Notably, the proposed approach performs exceptionally well with the Maniram Patel Mine Dataset (highlighted in bold in Table 1), surpassing all other image datasets in terms of performance. The table demonstrates that the proposed methodology performs poorly for the Thermal and Visible Aerial Imagery Dataset, primarily due to its inability to detect shadows of small objects. This limitation is inherent to our method. Figure 5 illustrates a performance comparison between the proposed methodology and the color-based method described by Desa et al. [23]. The recall, F1-score, and accuracy of our proposed method on the

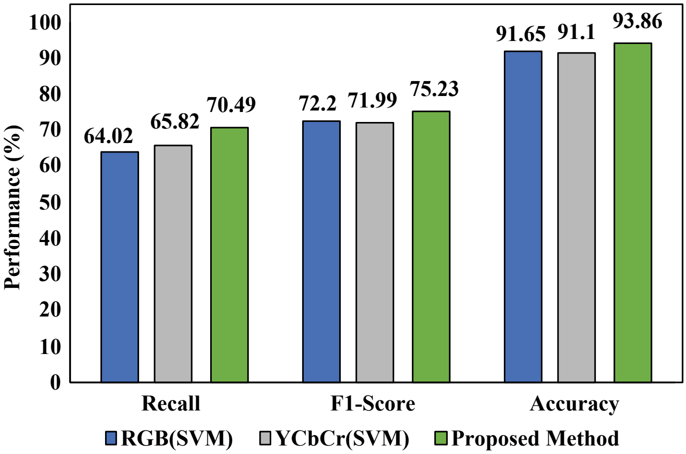


Fig. 5 Comparison of performance of the proposed shadow detection results with the performance of color-based shadow detection method based on accuracy, recall, and F1-score

Thermal and Visible Aerial Imagery Dataset are compared with those of RGB(SVM) and YCbCr(SVM) methods. The results depicted in this figure clearly indicate that our proposed method outperforms the method proposed by S. M. Desa et al.

5 Conclusion

This paper presents a novel iterative thresholding-based method for shadow detection. The method is implemented and evaluated using Python coding in the Spyder environment, employing four different UAV image datasets with diverse features that may pose challenges in shadow detection and accurate results. Experimental results demonstrate the effectiveness of the proposed approach in both mining and urban application areas. Particularly, the proposed method exhibits superior performance in mining images with an accuracy of 96.8%, as reported in this work. However, the limitations observed in shadow detection, specifically the failure to accurately identify small shadowed areas in high-resolution images, highlight the need for further research and development of a robust shadow detection framework tailored for such inputs. Additionally, further research is required to explore techniques for enhancing binary masks. Nevertheless, the outcome and analysis of these experiments provide a foundational assessment that can serve as a valuable resource for other researchers aiming to develop and improve shadow detection applications utilizing color characteristics.

References

1. Bist B (2018) Literature survey on unmanned aerial vehicle. Int J Pure Appl Math 119(12):4381–4387
2. Yang W, Guo W, Peng K, Liu L (2012) Research on removing shadow in workpiece image based on homomorphic filtering. Procedia Eng 29:2360–2364
3. Sharma D, Singhai J (2019) An object-based shadow detection method for building delineation in high-resolution satellite images. PFG-J Photogram Remote Sens Geoinf Sci 87:103–118
4. Arévalo V, González J, Ambrosio G (2008) Shadow detection in colour high-resolution satellite images. Int J Remote Sens 29(7):1945–1963
5. Tian J, Qi X, Qu L, Tang Y (2016) New spectrum ratio properties and features for shadow detection. Pattern Recogn 51:85–96
6. Polidorio AM, Flores FC, Imai NN, Tommaselli AM, Franco C (2003) Automatic shadow segmentation in aerial color images. In: 16th Brazilian symposium on computer graphics and image processing. IEEE, pp 270–277
7. Huang J, Xie W, Tang L (2004) Detection of and compensation for shadows in colored urban aerial images. In: Fifth world congress on intelligent control and automation, vol 4. IEEE, pp 3098–3100
8. Tsai VJ (2006) A comparative study on shadow compensation of color aerial images in invariant color models. IEEE Trans Geosci Remote Sens 44(6):1661–1671
9. Anoopa S, Dhanya V, Kizhakkethottam JJ (2016) Shadow detection and removal using tri-class based thresholding and shadow matting technique. Procedia Technol 24:1358–1365
10. Das RK, Shandilya M (2019) Optimization of shadow detector and color model index using automatic threshold and boundary refinement. Int J Eng Tech 5:1303–2395
11. Alvarado-Robles G, Osornio-Rios RA, Solis-Munoz FJ, Morales-Hernandez LA (2021) An approach for shadow detection in aerial images based on multi-channel statistics. IEEE Access 9:34240–34250
12. Pons X, Padró JC (2019) An empirical approach on shadow reduction of UAV imagery in forests. In: IEEE international geoscience and remote sensing symposium. IEEE, pp 2463–2466
13. Wang Q, Yan L, Yuan Q, Ma Z (2017) An automatic shadow detection method for VHR remote sensing orthoimagery. Remote Sens 9(5):469
14. Usha Nandini D, Leni ES (2019) Efficient shadow detection by using PSO segmentation and region-based boundary detection technique. J Supercomput 75:3522–3533
15. Silva GF, Carneiro GB, Doth R, Amaral LA, de Azevedo DF (2018) Near real-time shadow detection and removal in aerial motion imagery application. ISPRS J Photogram Remote Sens 140:104–121
16. Kang X, Huang Y, Li S, Lin H, Benediktsson JA (2017) Extended random walker for shadow detection in very high resolution remote sensing images. IEEE Trans Geosci Remote Sens 56(2):867–876
17. Luo S, Li H, Zhu R, Gong Y, Shen H (2021) ESPFNet: an edge-aware spatial pyramid fusion network for salient shadow detection in aerial remote sensing images. IEEE J Sel Top Appl Earth Observ Remote Sens 14:4633–4646
18. Zali SA, Mat-Desa S, Che-Embi Z, Mohd-Isa WN (2022) Post-processing for shadow detection in drone-acquired images using u-net. Future Internet 14(8):231
19. Zhang H, Sun K, Li W (2014) Object-oriented shadow detection and removal from urban high-resolution remote sensing images. IEEE Trans Geosci Remote Sens 52(11):6972–6982
20. Drone mapping software, image processing and geospatial, dronemapper (2022). https://dronemapper.com/. 14 Mar 2023
21. Garcia L, Diaz J, Correa HL, Restrepo-Girón A (2020) Thermal and visible aerial imagery. Mendeley Data 2:2020

22. Institute of Computer Graphics and Vision (2019). http://dronedataset.icg.tugraz.at. 14 Mar 2023
23. M-Desa S, Zali S, Mohd-Isa WN, Che-Embi Z (2022) Color-based shadow detection method in aerial images. In: J Phys: Conf Ser 2312:012081 (IOP Publishing)

A Subtle Design of Prediction Models Using Machine Learning Algorithms for Advocating Selection and Forecasting Sales of Garments: A Case Study

Dillip Rout, Bholanath Roy, and Prasanna Kapse

Abstract In this article, the predictive analysis is conducted for a garment retail dataset that contains the attributes of the dresses and sales information. Precisely, Random Forest (RF), Linear Regression (LR), Support Vector Machine (SVM), and Decision Tree (DT) algorithms are used for classification. That is, advising whether the dresses should be kept in store or not by automating the process of the recommendation. Moreover, two variants of the datasets are given as input to the said algorithms apart from the raw dataset. One variant is obtained through feature selection and another uses the concept of dummy variable since the majority of the features are categorical. In addition, the demand for sales is estimated over a period. Auto-Regressive Integrated Moving Average (ARIMA) is applied in particular to achieve the forecasting of the sales. The dataset contains fourteen features of dresses and sales data of alternative days over a month. The experiments on the case study show that RF algorithm is good at the classification although it is marginally better than LR. Also, the sales forecasting is producing results in an acceptable range as per the relevant performance metrics. Overall, the proposed methodology of this paper helps in the decision-making of fashion retail.

Keywords Classification · Logistic regression · Boruta · Feature engineering · Categorical features · Time series analysis

D. Rout (✉)
C.V. Raman Global University, Bhubaneswar, India
e-mail: dillip.rout.iitb@gmail.com

B. Roy
Maulana Azad National Institute of Technology, Bhopal, India

P. Kapse
Medi-Caps University, Indore, India

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_29

1 Introduction

The rapid growth of fashion industry is driven by a diverse demand which has incurred from the expedition of variance in customer trend [6]. This necessitates regular revisions of business models to keep up with the changing market trends [10, 15]. Basic rules and perceptions may not be sufficient to ensure smooth trading operations. As a result, some manufacturing units and retail outlets have automated their operations. However, achieving automation requires a deep understanding of business logic, cost, fashion attributes, demand, and customer choice [8]. Thus, automation is a critical goal in the fashion industry. The fashion retail industry faces bottlenecks due to industry characteristics and customer demand, necessitating the formulation of models at an individual level, as these factors differ from case to case [17].

In this regard, the machine learning algorithms are explored in this paper for the automated decision-making process in fashion retail. In addition, the analysis is also focused on forecasting the future demand for each garment based on history. Each of these questions deals with one or multiple machine learning algorithms such as LR, Logistic Regression (LGR), DT, RF, *etc*, as it is a data-driven approach where a single algorithm may not be sufficient. A dataset of retail information is given which contains attributes or features and sales of a set of garments. All the applied algorithms are demonstrated through a case study which consists of 15 features including attributes and sales data for 500 garments. The particular objectives of the study are as follows:

1. To compare a set of models to predict the recommendation of products for future stocks.
2. To build a forecasting model to estimate the stocks of dresses for three alternative days.

The novelty includes creating dummy variables for the variables in addition to the feature selection. This study compares the impact of various approaches to vanilla prediction models. It provides valuable insights for high-end fashion retail stores seeking business expansion.

The remainder of the article is organized as follows: The next section has a literature review followed by the research gaps. Then, a theoretical background of the applied machine learning techniques is discussed in Sect. 3. Next, a case study of the set of the available input data is described. Thereafter, the description and predictions are applied and discussed with the results (Sect. 4). Lastly, the concluding remarks are presented in Sect. 5.

2 Previous Works

This paper focuses on fashion or garment retail, but there are very few papers which have the same theme, so the review is extended to the articles which address general retail problems. The literature shows that the retail industry is studied in three facets, namely the behavior of consumers, decision-making, and demand forecasting.

Consumer behavior is crucial for determining retail industry production needs. [6, 17]. Classifiers like decision table, DT, RF, SVM, *etc.*, have been used to fit consumer behavior and build recommended systems. For instance, the decision table classifier provides the highest accuracy level for the consumer behavior for online shopping data [1]. The filtered classifier has the lowest accuracy in predicting consumer behavior. Previous attempts include clustering techniques, association rules, random tree, and forest. Combining sentimental analysis and neural networks provides better precision in product price setting [13]. Digital signage is used to study purchase decisions and situations, predicting consumer behavior [22]. Other methods include clustering techniques, association rules, random tree and forest, and sentimental analysis. In this case, SVM is proven to result in the best output with high accuracy. Differently, the usage of Internet of Things in the application of smart stores is studied through the application of indoor positioning, augmented reality, facial recognition, and interactive display [11]. Furthermore, attitudes and subjective norms are found to be the key predictors for online fashion renting which is found through confirmatory factor analysis and structural path analysis [17]. Ease of shopping is the most influencing factor in consumer behavior [28]. It is concluded that these applications improve the experience of consumers and their behavior of buying from offline stores. Overall, multiplexing technology has played a critical role in both online and offline shopping and sales have increased.

Retail management decision-making is challenging due to the variety of features to be processed [23]. Correlation analysis is used to reduce dimensions and a RF classifier is applied to lowly correlated features. Discount offers are found to increase sales, possibly with combined products. Nevertheless, there is a saturation point to the discount where the sales do not increase. Clustering-based algorithms are applied to assess the sales of retail stores [25]. Four algorithms are studied for classifying retail data: K-means, density-based, filtered, and farthest first clustering. The farthest first clustering algorithm is robust and accurately classifies retail sales. Social networking, consumer participation, and feedback are also studied to enhance decision-making in retail management [19]. The retail industry offers more scope for analysis and research on sales and decision-making for sustainability and growth.

Most research articles are found for retail demand forecasting. A comprehensive review is provided for fashion retail given operational issues for both demand and supply sides [31]. As a result of the involvement of doubts on both sides, it is complicated. Particularly, it is challenging to build mathematical models for items with short life cycles and considerable demand unpredictability. The bottleneck for sales forecasting, according to academics, is risk management, which necessitates micro-level estimates of seasonal demand, pricing delays, product differentiation, and decisions about product design and manufacture. For instance, online retailers optimize pricing through demand forecasting using machine learning techniques [9]. A comparative study found the Gradient boost algorithm as the best model for prediction due to its lowest Root Mean Square Error (RMSE) and highest R^2 value [16]. Also, it is revealed that hyperparameter tuning is required for high performance, especially for AdaBoost algorithm. Similarly, machine learning techniques are used to establish relationships between parametric models, user-selected covariates, and non-parametric approaches [2]. Deep learning methods have high prediction accuracy for retail sales, outperforming LGR due to multi-attributes [14]. Thus, machine learning is suitable for handling large data, and classification methods are more efficient than regression in this context [10]. Overall, machine learning and deep learning methods are suitable for demand prediction for retail sales.

There are a few papers that have created models for retail management. The majority of these were demand forecasting and consumer behavior-focused. A few papers, however, are discovered to be beneficial in the decision-making process for in-store retail management. Particularly, there is less emphasis placed on the connections between the properties (features). Furthermore, by modeling the attributes in terms of correlation, the relationship between the attributes is missed. Mention how a retail business may include categorical features yet correlation analysis may only be applied to numeric attributes. Therefore, it is crucial to investigate this matter further. The relationship between traits is crucial for further research on sales and price, necessitating further study of the inherent attributes' relationship beyond demand forecasting.

3 Proposed Framework

The proposed framework has the following three parts. The first part refers to the data description and preparation (Sect. 3.1). The second part describes the selection of the important features (Sect. 3.2). The third part contains the discussion of the prediction models in Sect. 3.3. All the three parts are described as follows.

3.1 Data Preprocessing

The preprocessing of the data includes data wrangling. The dataset contains two files—(i) attributes of dresses and (ii) sales of dresses [26]. The former contains the attributes such as Dress_ID, Style, Price, Rating, Size, Season, NeckLine, SleeveLength, WaistLine, Material, FabricType, Decoration, PatternType and Recommendation [24, 26]. All the features listed in the first set are categorical except Rating which is a numerical feature [12, 24]. Furthermore, the sales file contains the sales for each dress on a particular date. The date ranges from 29/8/2013 to 12/10/2013, and the sales are entered for alternative days. The date fields are properly converted for processing. The NAs are dealt with by replacing the values with 0 s and mode for numerical and categorical features respectively. Consistency of sales and attribute dataset is checked by ensuring that all dress codes are having sales and vice-versa. All labels are converted into small alphabets to adjust small and capital letter issues. Also, the hyphen is replaced by an underscore in observations since it will create issues while applying dummy variables for factors. Remove the rows with lower frequency labels say less than 5 to avoid fitting issues while segregating training and test data.

3.2 Feature Selection

The methodology adopted for selecting the features is based on hierarchy [24]. The hierarchy is generated by taking the Geometric Mean (GM) of the importance scores from various algorithms. The algorithms used for feature selection include RF, mRMRE, Boruta and LR [4, 5, 7, 18, 20]. Furthermore, a lower boundary is decided by the GM values where there is a significant gap realized and a sufficient number of features are available for the prediction.

3.3 Predictive Analysis

The predictive analysis in this context of garment retail is a binary classification. Precisely, automate recommendation column to advise if the garment should be kept or not. In this regard, four models, namely LGR, DT, RF, and SVM models are chosen for applying classification [4, 12, 18, 30]. Further, these models are tested on three variations of the data, *i.e.*, (i) using the original data which is referred to as raw, (ii) selecting important features which are tagged as feature, and (iii) by using dummy variables for each of the categorical variables which are called as dummy (Sect. 4.1).

The analysis of classification is carried out based on sensitivity (Eq. 1), specificity (Eq. 2), and accuracy (Eq. 3) [32]. All these metrics are defined using true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In particular, these metrics are specified mathematically as follows:

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \tag{1}$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \tag{2}$$

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TN} + \text{TP} + \text{FN} + \text{FP}) \tag{3}$$

In addition, a time series analysis is conducted to predict the sales of dresses over the next three alternative days. In order to achieve it, first the sales dataset was converted into time series data. Then, the prediction was done using auto.arima() model of forecast package [21]. However, the issue was this model is for univariate forecasting. Hence, each Dress_ID was estimated separately in an iterative manner.

Table 1 Comparison of the impact of features on TotalSales using various models [24]

Feature	RF[a]	mRMRE	mRMRE-D	Boruta	LR[b]	GM
Rating	0.381	0.236	0.236	13.226	1.000	0.775
Pattern.Type	0.454	0.098	0.035	4.633	0.813	0.358
Recommendation	−1.395	0.033	0.033	1.958	0.708	0.185
Style	0.139	0.029	−0.016	3.670	0.356	0.097
Size	−0.180	0.065	0.001	4.746	0.707	0.075
NeckLine	−0.729	−0.086	−0.044	1.140	0.406	0.066
Season	−0.357	0.002	0.004	2.885	0.597	0.057
WaistLine	−0.501	0.054	0.010	−0.441	0.429	0.055
FabricType	−0.451	−0.059	−0.024	0.706	0.558	0.048
Decoration	0.214	0.027	−0.013	−0.552	0.339	0.042
SleeveLength	0.044	−0.096	−0.008	0.351	0.308	0.032
Price	−1.251	−0.021	−0.048	0.107	0.173	0.030
Material	−0.214	−0.098	−0.004	−0.573	0.529	0.019

RF Random Forest [4, 18], *mRMRe* Minimum Redundancy Maximal Relevancy Ensemble [7], Boruta [20], *LR* Linear Regression [5], *GM* Geometric Mean

[a] The values are obtained by taking natural logarithm and subtracting 22

[b] The values are obtained by taking the complement of mean *p*-value of all the levels per feature

4 Experimental Study

The techniques used include feature selection, recommendation automation, and sales estimation, with experiments conducted using R programming language. The prediction algorithms are trained on 60% of the data, with the remaining 40% for testing.

4.1 Feature Selection

The importance of the features is listed in Table 1. Two major threshold points of separation are observed as separated in the said table as per the discussion in Sect. 3.2. The lower threshold point is chosen based on GM values [24]. As a consequence, SleeveLength, price, and material features are dropped from the input variables as these have a very low influence on TotalSales, and have substantial different GM from the next upper-level selected features.

4.2 Automation of Recommendations

The comparison of models for classifying recommendation is shown in Table 2. The results are evaluated based on sensitivity, specificity, and accuracy [32]. It is

Table 2 Comparison of classification models for predicting recommendation

Metrics	LGR	DT	RF	SVM
Raw				
Sensitivity	0.640	0.631	0.703	0.631
Specificity	0.479	0.487	0.610	0.551
Accuracy	0.569	0.563	0.669	0.606
Feature				
Sensitivity	0.652	0.647	0.636	0.614
Specificity	0.563	0.507	0.509	0.522
Accuracy	0.625	0.581	0.594	0.588
Dummy				
Sensitivity	0.697	0.596	0.649	0.610
Specificity	0.393	0.459	0.565	0.483
Accuracy	0.538	0.544	0.625	0.563

observed that all models performed better with raw data input except DT model whose efficiency is good when applied with feature data. Thus, feature selection and the use of dummy data are not impactful for the prediction.

Note that the performance of DT and SVM is not as par with the rest of the models. In other words, LGR and RF models perform better but those are non-dominant with respect to various types of data, *i.e.*, raw, feature, and dummy. Thus, either of these can be used for the automation of recommendation as per the business logic.

4.3 Sales Forecast

A snippet (up to first 10 Dress_ID) of the forecast data is given in Table 3. It is observed that there is no significant variation among the alternative days. Moreover, the forecasting is transformed into a moving average although there is no way to prove the phenomenon.

The fitness of the model is provided in terms of RMSE and mean absolute percentage error (MAPE) metrics as shown in Table 4 which are collected over each Dress_ID. It is observed that RMSE values are not satisfactory although the median is likely to be acceptable. However, MAPE values are good although outliers exist with a higher percentage. So, overall, this approach is acceptable in terms of predicting with reasonable accuracy.

Table 3 Sales forecasting of dresses for three alternative days

Dress_ID	14/10/2013	16/10/2013	18/10/2013
1006032852	4110	4173	4235
1212192089	4464	4652	4839
1190380701	11	11	11
966005983	1967	1971	1975
876339541	2815	2894	2973
1068332458	27	27	28
1220707172	575	598	621
1219677488	274	286	298
1113094204	34	35	36
985292672	14	14	14
1117293701	143	149	154
898481530	210	218	226
957723897	3058	3137	3216
749031896	4246	4322	4398
1055411544	53	53	53
1162628131	229	239	249
624314841	2568	2611	2654
830467746	19	19	19
840857118	17	18	19
1113221101	667	691	715
861754372	411	424	437
856178100	1800	1842	1884
1122989777	235	233	232
840516484	2417	2459	2505
768517084	5	5	5
1139843344	30	31	32
1004212992	3173	3266	3362
1235426503	465	485	505
942808364	538	539	540
629131530	5654	5556	5458
851945460	1152	1152	1152
1150275464	249	249	249
1026634314	790	790	790

Table 4 Fitting accuracy of forecasting model—auto.arima

Metrics	RMSE	MAPE
Min.	0.00	0.00
1st Qu.	5.61	3.29
Median	62.29	5.33
Mean	119.99	5.32
3rd Qu.	148.32	6.64
Max.	1956.56	23.64

5 Conclusions

A dataset containing retail information on a set of dresses is investigated in this article to predict recommendation and to forecast sales. The machine learning algorithms like LGR, DT, RF, and SVM are used for classifying recommendation. These algorithms are tested using three types of inputs, *i.e.*, with the raw data, selected features, and dummy variables. Furthermore, the time series forecasting is conducted for sales for a couple of days more for the given set of two months sales data. In this context, ARIMA model is being used to estimate the sales for each dress.

The evaluation shows that LGR and RF are useful for classifying recommendation. These two algorithms work well with all types of input data. However, it is observed that the dummy variable has almost no impact on the classification. Also, the impact of scaling was not significant enough as per the observations. However, feature selection has a slight impact on classification and prediction as per the results. The accuracy is degraded for RF while using selected features and dummy variables. Furthermore, the forecasting can find out the future sales but is not prominent enough since it results roughly as a moving average.

The future scope redirects to improving the accuracy of the models used. Also, the discretization of the features must be applied for those numerical variable which actually mean categorical properties. In addition, testing the proposed methodology with a larger dataset remains a challenge.

References

1. Ahmed RAED, Shehab ME, Morsy S, Mekawie N (2015) Performance study of classification algorithms for consumer online shopping attitudes and behavior using data mining. In: Proceedings—2015 5th international conference on communication systems and network technologies, CSNT 2015, pp 1344–1349
2. Bajari P, Nekipelov D, Ryan SP, Yang M (2015) Machine learning methods for demand estimation. Am Econ Rev 105:481–485
3. Bradlow ET, Gangwar M, Kopalle P, Voleti S (2017) The role of big data and predictive analytics in retailing. J Retail 93(1):79–95

4. Brownlee J (2019) How to perform feature selection with categorical data. https://machinelearningmastery.com/feature-selection-with-categorical-data/. Last accessed on Dec 2021
5. Brownlee J (2020) How to perform feature selection for regression data. https://machinelearningmastery.com/feature-selection-for-regression-data/. Last accessed: Dec 2021
6. Dahana WD, Miwa Y, Morisada M (2019) Linking lifestyle to customer lifetime value: an exploratory study in an online fashion retail market. J Business Res 99(1):319–331
7. De Jay N, Papillon-Cavanagh S, Olsen C, El-Hachem N, Bontempi G, Haibe-Kains B (2013) MRMRe: an R package for parallelized mRMR ensemble feature selection. Bioinformatics 29(18):2365–2368
8. Dekimpe MG (2020) Retailing and retailing research in the age of big data analytics. Int J Res Mark 37(1):3–14
9. Ferreira KJ, Lee BHA, Simchi-levi D (2016) Analytics for an Online Retailer?: demand forecasting and price optimization. Manuf Serv Oper Manage Publ 18(1):69–88
10. Huber J, Stuckenschmidt H (2020) Daily retail demand forecasting using machine learning with emphasis on calendric special days. Int J Forecast
11. Hwangbo H, Kim YS, Cha KJ (2017) Use of the smart store for persuasive marketing and immersive customer experiences: a case study of Korean apparel enterprise. Mob Inform Syst
12. Jain D (2020) Feature handling: categorical and numerical. https://towardsdatascience.com/feature-handling-3f14c12ecbb8. Last accessed: Jan 2022
13. Kalaiselvi N, Aravind KR, Balaguru S, Vijayaragul V (2017) Retail price analytics using backpropogation neural network and sentimental analysis. International conference on signal processing, communication and networking, ICSCN 2017:16–21
14. Kaneko Y, Yada K (2016) A deep learning approach for the prediction of retail store sales. In: IEEE International Conference on Data Mining Workshops, ICDMW, vol 16, pp 531–537. IEEE
15. Khakpour A (2020) Data science for decision support: using machine learning and big data in sales forecasting for production and retail. Ph.D. thesis, Ostfold University College
16. Krishna A, Akhilesh V, Aich A, Hegde C (2018) Sales-forecasting of retail stores using machine learning techniques. In: Proceedings 2018 3rd international conference on computational systems and information technology for sustainable solutions, CSITSS 2018, pp 160–166. IEEE
17. Lee SH, Chow PS (2020) Investigating consumer attitudes and intentions toward online fashion renting retailing. J Retail Consum Serv 52(1):101892
18. Lewinson E (2019) Explaining feature importance by example of a random forest. https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e. Last accessed: Dec 2021
19. Lorenzo-Romero C, Andrés-Martínez ME, Cordente-Rodríguez M, Gómez-Borja MÁ (2021) Active participation of e-consumer: a qualitative analysis from fashion retailer perspective. SAGE Open 11(1)
20. Mazzanti S (2020) Boruta explained exactly how you wished someone explained to you. https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a. Last accessed: Jan 2022
21. Pulagam S (2020) Time series forecasting using Auto ARIMA in python. https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd. Last accessed: Jan 2022
22. Ravnik R, Solina F, Zabkar V (2014) Modelling in-store consumer behaviour using machine learning and digital signage for audience measurement data. Springer International Publishing Switzerland 8811:123–133
23. Razmochaeva NV, Klionskiy DM (2019) Data presentation and application of machine learning methods for automating retail sales management processes. In: Proceedings of the 2019 IEEE conference of Russian young researchers in electrical and electronic engineering, ElConRus 2019, IEEE, pp 1444–1448, no 1
24. Rout D, Roy B (2023) Multi-algorithm machine learning model for important feature extraction for sales of a garment retail dataset. In: 1st international conference on recent trends in multidisciplinary research and innovation, pp 282–286

25. Shrivastava V, Narayan Arya P (2012) A study of various clustering algorithms on retail sales data. Int J Comput 1(2):68
26. Simplilearn (2013) Garment sales dataset. https://github.com/dilliprout/GarmentSales2013. Last accessed: Dec 2021
27. Soguero-Ruiz C, Gimeno-Blanes FJ, Mora-Jiménez I, Martínez-Ruiz MP, Rojo-Álvarez JL (2012) On the differential benchmarking of promotional efficiency with machine learning modeling (I): principles and statistical comparison. Exp Syst Appl 39(17):12772–12783
28. Suleman D, Zuniarti I, Marginingsih R, Susilowati IH, Sari I, Sabil S, Nurhayaty E (2021) The effect of decision to purchase on shop fashion product in Indonesia mediated by attitude to shop. Manage Sci Lett 11:111–116
29. Tokatli N (2008) Global sourcing: insights from the global clothing industry–the case of zara, a fast fashion retailer. J Econ Geogr 8(1):21–38
30. Varghese D (2018) Comparative study on classic machine learning algorithms. https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222. Last accessed: Jan 2022
31. Wen X, Choi TM, Chung SH (2019) Fashion retail supply chain management: a review of operational models. Int J Prod Econ 207:34–55
32. Zhu W, Zeng N, Wang N (2010) Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS® implementations. In: Northeast SAS Users Group 2010: Health Care and Life Sciences, pp 1–9

BGKnow-Medical Chatbot: A Hybrid Approach Based on Knowledge Graph and GPT-2

Disha Sunil Nikam, D. Nisha Murthy, Sreeramya Dharani Pragada, and H. R. Mamatha

Abstract Accurate and timely diagnosis is critical in ensuring patients receive care and treatment for their medical conditions. Traditional symptom checkers often lack accuracy and efficiency in diagnosing diseases, as they typically rely on preprogrammed decision trees or rule-based algorithms that may not account for the complexity and variability of symptoms. By using natural language processing (NLP) and machine learning techniques, such as knowledge graphs and Bio-Bidirectional Encoder Representations From Transformers (BioBERT), chatbots can provide a more accurate and personalized approach to disease diagnosis. This paper introduces a hybrid chatbot framework called "BGKnow." BGKnow represents the combination of a knowledge graph and Generative Pre-trained Transformer 2 (GPT-2) model that uses BioBERT embeddings for effective diagnosis based on symptoms entered by users. The proposed system shows promising potential for assisting healthcare professionals in accurately and efficiently addressing medical inquiries.

Keywords Natural language processing · Knowledge graphs · BERT · GPT-2 · Aho-Corasick · Chatbots

1 Introduction

Patients have long faced obstacles while accessing primary care services, including difficulties in scheduling appointments, extended waiting times, and inconvenience in seeing a doctor. Our system aims to bridge the gap created due to these obstacles and aid in timely diagnosis. Artificial intelligence (AI) is a significant contributor to the advancement of information technology in healthcare, with chatbots emerging as a prominent AI solution for boosting healthcare service quality and efficiency. Chatbots are software systems that offer interactive interfaces for patients or medical professionals, enabling tasks such as knowledge extraction and personalized

D. S. Nikam (✉) · D. Nisha Murthy · S. D. Pragada · H. R. Mamatha
PES University, 560085 Banashankari, Bengaluru, Karnataka, India
e-mail: sup.disha@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_30

feedback in real-time. The medical field has seen rapid development in chatbot technologies, with various applications such as assisting patients in symptom identification and directing them to appropriate healthcare service departments. These include medical assistants and front desk systems for medical services, streamlining the patient experience. In recent years, NLP models such as GPT-2 [1] and BioBERT [2] have shown significant promise in improving the accuracy of chatbots for disease diagnosis. Knowledge graphs provide structured information on diseases and their associated symptoms. However, the variability and complexity of symptoms can limit the accuracy of knowledge graphs, which can lead to misdiagnosis. To address this challenge, our chatbot BGKnow introduces the novelty of a diagnosis system that combines knowledge graphs with a GPT2 model that uses BioBERT embeddings to increase the accuracy of disease diagnosis. It is fine-tuned on biomedical data, enabling it to understand the nuances of medical language and provide accurate diagnoses based on the user symptoms. The research question that guides this study is: Can the use of knowledge graphs and BioBERT in a chatbot-based symptom diagnosis system improve the accuracy and efficiency of disease diagnosis compared to traditional symptom checkers? To answer this question, we conducted experiments to evaluate the performance of our proposed system. The results demonstrate that our system achieved higher accuracy rates and improved efficiency in disease diagnosis, making it a valuable tool for healthcare professionals. The knowledge graph contains above 600 distinct varieties of disease-type records from prominent medical forums and resources and can answer six distinct categories of inquiries. When a user poses a question, our system queries the knowledge graph. If no relevant results are found, the model passes the input to the BioBERT-trained GPT-2 model and gives a response. This approach improves the coverage of our system, as it can handle a wide range of questions and queries related to medical conditions.

2 Literature Review

Chatbots: Professor Joseph Weizenbaum of MIT developed the chatbot “Eliza” in the 1960s and was the first chatbot known to exist [3]. The program was designed in a way that mimics human conversation. “Parry” was made by psychiatrist Kenneth Colby in the year 1972 [4]. It imitated a patient suffering from schizophrenia. In 2009, “WeChat” in China built a highly advanced chatbot.

Users can log in to the system and register on the chatbot application. On interaction with the system, the words used and symptoms defined are recognized using NLP techniques. The Naive Bayes algorithm further predicts the user’s disease. An admin is responsible for the working of the chatbot application. The admin views user details and adds, deletes, or updates respective symptoms and diseases [5]. Dharwadkar et al. [6] proposed a system that aids medical institutes and hospitals to assist the respective users by enabling them to voice-activately ask queries about medicinal doses. SVM algorithm has been used with disease symptoms system to predict diseases. For the conversion of voice-to-text and text-to-voice, Google API is employed.

The chatbot receives the query and retrieves the corresponding query answer which is displayed on the android application. The user inputs a question into the UI, which is then sent to the chatbot app. The chatbot applies pre-processing phases like tokenizing, removing stop words, and extracting features using techniques like TF-IDF, N-grams, and cosine similarity. The questions and respective answers are stored in a knowledge-oriented database for retrieval [7]. Mahajan et al. [8] proposed a system that contains a question and answer covenant pertaining to the style of chatbot to respond to user queries. Answers to such sentences are derived via the sentence's major keywords. If a match is found or vital answers are provided or interchangeable answers are displayed, the system identifies the respective type of illness of the user supported by user symptoms. In the system proposed, the user conversation is made up of a linear design which contains extraction of symptoms, mapping of symptoms, where the corresponding symptoms are identified; further, the patients are diagnosed based on the disease falling into the major or minor category. If major, the respective doctor is directed to the patient, the doctor details are retrieved from the database and the user identification takes place by usage of the login details present in the database. String Searching Algorithm is used where the substring representing the desired symptoms is mapped in the natural language text input to extract the user symptoms [9]. Three LSTM approaches which include BiLSTM, stacked LSTM, and simple LSTM were used which contain three word embedding types, namely one hot encoding, BERT embeddings, and fastText. Five architectures based on different encoder and decoder vectorization units were utilized. The size of the datasets was extremely small having few different subjects addressed in various topics, causing the research to become more intriguing along with more challenging. This marked the initial effort to train generative chatbots to understand a language with complicated morphology [10]. Three different approaches to deep learning were explored within the study. Using real-valued vector or word embeddings in the form of the input layer to the three models, the transformers depicted an average similarity score of 80% and an average BLeU score of 58% which, in turn, performed better than LSTM and Bi-LSTM models [11].

3 Proposed Methodology

3.1 Dataset

The dataset used for establishing the medical knowledge graph was collected from a diverse range of medical websites [12]. These websites include (1) NHS Inform and (2) MedicineNet. Additionally, the dataset prepared by Lasse Regin Nelson was employed for this project. This dataset is publicly available on his GitHub repository. It consists of a question and their corresponding answers taken from well-known medical websites such as eHealth Forum, iCliniq, Question Doctors, and WebMD. Medical professionals have answered patients' questions on these platforms. The dataset encompasses approximately 25,000 question–answer pairs, each associated with relevant tags. These tags were assigned to categorize the questions based on the corresponding diseases. A unique set of tags was employed in this dataset. Furthermore, access to the source URLs of the question–answer pairs were granted to validate the accuracy and reliability of the dataset.

3.2 The Knowledge Graph Architecture

Neo4j Graph Database Storage: Neo4j graph database software stores a knowledge graph consisting of three entities: department, disease, and symptom. It also comprises six properties: name, cause, description, accompany, prevent, and cure way, accompanied by five relationships: have symptom, accompany with, disease prevent, disease cause, and disease cureway. The graph consists of approximately 3502 entities (including 677 diseases and 2825 symptoms) and 4501 relationships, which encompass connections between diseases, symptoms, and the aforementioned properties.

Answer Selection Process from the Graph Database: From the graph database, the answers are selected in five steps. First, the user inputs a question. Then, an entity extractor called D&S Extractor identifies the entities related to diseases and symptoms. Next, an Intention Recognizer determines the user's intention. Following that, the system performs answer selection based on the identified entities and intentions. Finally, the selected answer is returned to the user. Figure 1 [12] illustrates an example of this process, where the word "cold" is recognized as a keyword for disease, the intention "has symptom" is identified, and the answer "fever" is selected.

Design of the Problem Analysis Module: Figure 2 [12] represents the functionality of the system's disease symptom entity extraction. This function extracts disease keywords using the Aho-Corasick algorithm, which searches a medical keywords dictionary. If the algorithm fails to identify the symptoms and disease in a given question, the system makes use of a semantic similarity computation module to determine the entities that are the most similar. The user's interaction is recognized by predicting the intentions of the user using predefined predicate libraries. If the intention cannot be recognized, the system prompts the user to rephrase or clarify the question, indicating a lack of understanding. Overall, the system can answer six typical questions based on the five relationships established in the graph database.

3.3 Generative Pre-trained Transformer 2

With approximately 1.49 billion parameters, GPT-2 is a parameter transformer that performs well on 7 of 8 language modeling datasets. However, WebText is its underlying weakness where it seems to fall short. It is taught to anticipate the following

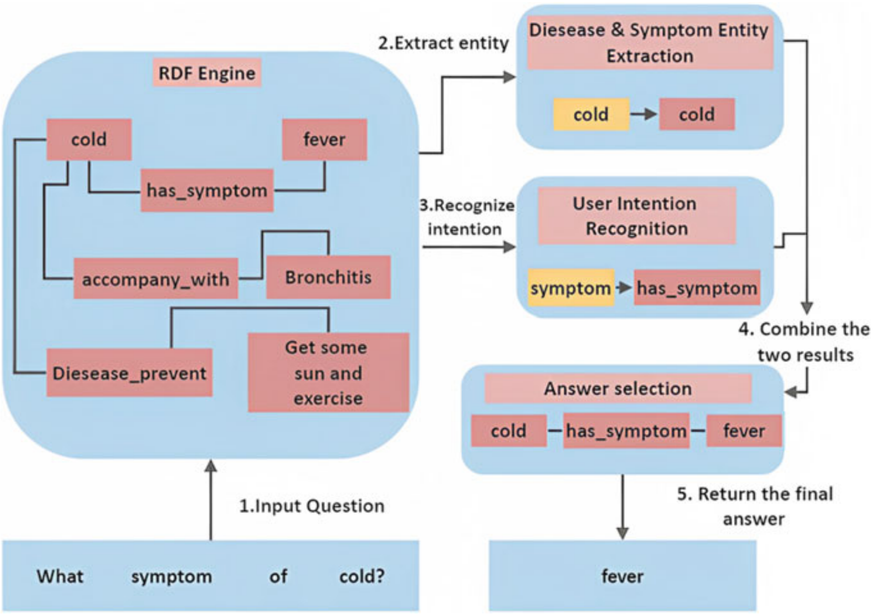


Fig. 1 Knowledge graph flow diagram [12]

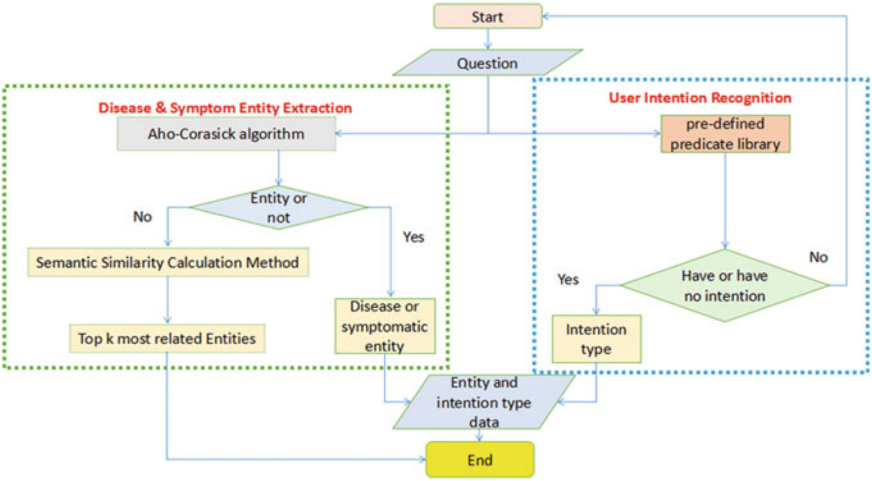

Start
Question
Disease & Symptom Entity Extraction
Aho-Corasick algorithm
Entity or not
No
Yes
Semantic Similarity Calculation Method
Top k most related Entities
Disease or symptomatic entity
User Intention Recognition
pre-defined predicate library
Have or have no intention
Yes
No
Intention type
Entity and intention type data
End

Fig. 2 Entity detection and intention recognition [12]

word in a passage of text given all the words that have come before it. More specifically, targets are considered as the same sequences as inputs, but with one token (chunk or word) advanced to the right. The inputs are predetermined length continuous text sequences. Figure 3 depicts the GPT-2 architecture where to ensure that forecasts for the ith token only take into account inputs in the range 1 to i and not subsequent tokens, the model internally employs a mask mechanism. It contains 12 different layers, each containing 12 independent attention mechanisms, named “heads” and the result consists of 12 × 12 = 144 distinguishable attention patterns. GPT-2 is trained on ten times the data used for GPT. It exhibits a wide range of capabilities, such as generating high-quality synthetic text samples with impressive continuity when given an initial prompt. It surpasses other language models which have been trained on particular domains like Wikipedia, books, and Internet sources even though it doesn’t rely on domain-specific training datasets. Moreover, with the ability to learn these tasks directly from unprocessed text without the requirement for task-specific training data, GPT-2 exhibits increasing progress on language-related tasks such as question answering, reading comprehension, summarizing, and translation.

Transformers use a semi-supervised approach for language understanding tasks by combining unsupervised pre-training and supervised fine-tuning [13]. The transformer architecture is a neural network that uses self-attention to learn long-range dependencies between words in a sequence. The attention layer is the basic unit of the transformer architecture which takes a sequence of vectors as input and outputs a new vector. This new vector represents the attention weights for each word in the sequence. The attention weights are used to determine how much weight to give to each word when computing the output vector. The attention function is computed for a set of queries at the same time, where the queries are packed together into a vector Q [14].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where: Q is the query vector, which is a representation of the current word in the sequence. K, the key vector, represents all the other words in the sequence and V, the value vector, represents the meaning of all other words present in the sequence. d_k is the dimension of the key and value vectors. softmax is a function that normalizes the attention weights so that they sum to 1. The GPT-2 model consists of a stack of attention layers where the output of each attention layer is fed into the next attention layer, and so on. The final output received from the model is a vector that represents the probability distribution over all possible words in the vocabulary

3.4 Overall System Architecture

Figure 4 illustrates the overall system architecture. The user enters the query in the user interface provided. The proposed system, BGKnow, first scans the knowledge graph to retrieve the appropriate answer for the user query. If found, the answer is displayed to the user. Else, the program flow passes to the GPT2 model to retrieve an

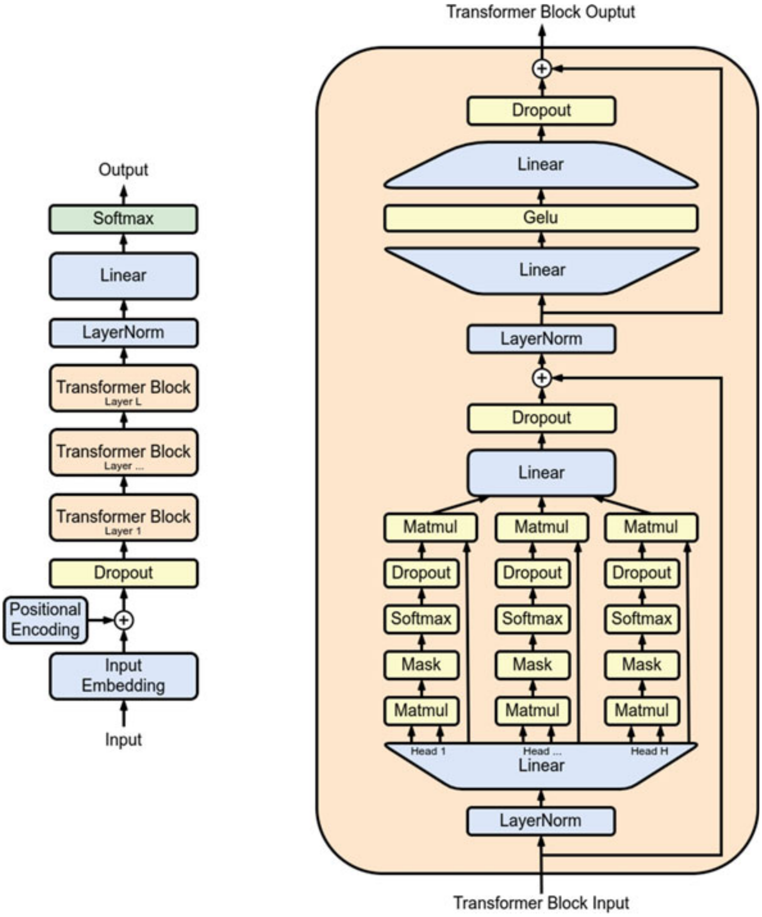


Fig. 3 GPT architecture

appropriate answer. The GPT2 model is trained using BioBERT embeddings which in turn are trained on the dataset used. The application of the neural-based model post the knowledge graph increases the accuracy of the system and provides a wider coverage of user queries.

4 Results and Discussion

4.1 Cosine Similarity

It is a distance assessment metric that assesses the degree of similarity between two vectors in an inner product space. The angle formed by two vectors is cosine-

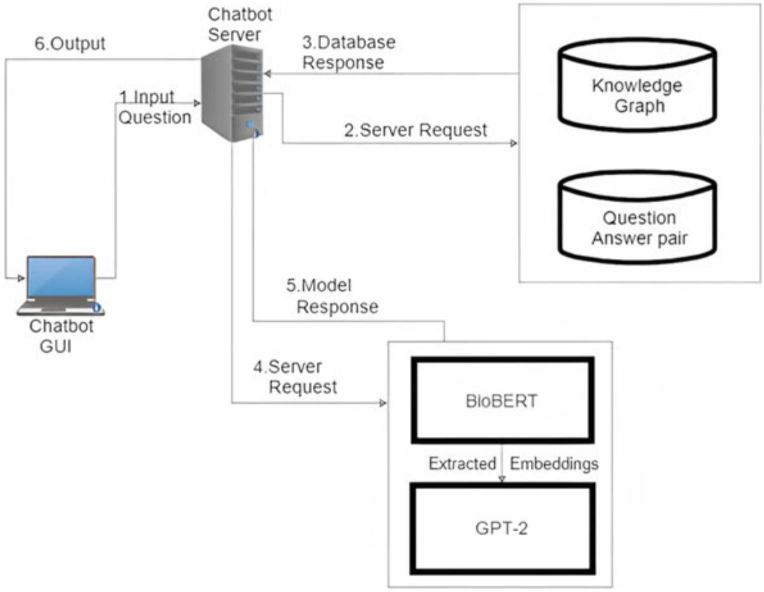


Fig. 4 Overall system architecture

measured. In text-based data, it is used to find the similarity between the original text and the vectorized text. In mathematical terms, the cosine similarity can be represented by the formula:

$$\cos(x, y) = \frac{x \cdot y}{||x||||y||} \tag{1}$$

where:

- x and y are the embeddings of two words or phrases
- . is the dot product operator
- $\|x\|$ and $\|y\|$ are the Euclidean norms of x and y

With respect to this measure, the similarity of the data points diminishes with increasing distance. To deploy cosine similarity in text-based data, the raw data is tokenized initially, and further a similarity matrix is generated which can be passed on to the cosine similarity metrics. This assesses the degree of textual resemblance. In BGKnow, the similarity between each current case and the prior cases is obtained using scikit cosine similarity from sklearn.metrics.pairwise.

4.2 Training and Testing Data

The dataset is a mixture of four json files named as icliniqQAs, ehealthforumQAs, questionDoctorQAs and webmdQAS [12] containing question answer pairs. The entire data was split into train test data in the ratio 80: 20 with random state = 42. Another dataset used contains only the ehealthforumQAs json file. Kalla et al. [15] proposed a system where the medical chatbot created uses the linear design, where it shows extraction symptoms toward the mapping symptom and the cosine similarity threshold set is 0.2. In some cases, the output may have a low cosine similarity score and not necessarily be an exact match due to the nature of the dataset used. Thus, a threshold of 0.3 is chosen for BGKnow to measure the similarity using cosine similarity. Accuracies for both mentioned datasets are compared [16].

4.3 Comparison

To evaluate the performance of BGKnow, accuracy is calculated by comparing the generated answers to the ground truth answers. Here, generated answers refer to the answers generated by the GPT-2 models used. TF-IDF vectorization and cosine similarity are used to measure the similarity between the generated and ground truth answers. A threshold(here 0.3) is used to compare the cosine similarity scores, and if any score surpasses it, the solution is deemed accurate. The ratio of the number of accurate responses to all of the questions is then used to measure accuracy.

4.4 Sample Results

Considering a sample input to the system, “Tell me something about chest pain.” BGKnow first scans the knowledge graph and retrieves the corresponding answer “Symptoms chest pain may be infected with: sinus tachycardiasilicosiscongenital pulmonary cystalveolar proteinosiseosinophiliaidiopathic hypereosinophilic syndromepneumonia pseudotumorbreast cancerlung abscessmycoplasmal pneumonia in children” as it is available in the knowledge graph stored in the Neo4j software. However, considering another sample input “Head ache” retrieves the answer “hi i am not sure what you are experiencing but i would suggest you consult a neurologist and get a diagnosis thanks” from the GPT-2 model post-failure of fetching the answer from the knowledge graph.

Table 1 Accuracies (accuracies are in %)

Models	Accuracy
Model 1 (entire dataset)	76.68
Model 2 (small portion of the dataset)	55.78

4.5 Accuracies Obtained

Model description: Two GPT-2 models have been deployed which are distinguished based on the quantity of data used to obtain Bio-BERT embeddings and carry out further training via GPT-2 architecture. Model 1 is trained on the entire dataset.(icliniqQAs, ehealthforumQAs, questionDoctorQAs, and webmdQAS) Model 2 is trained on ehealthforumQAs only. The cosine similarity threshold set is 0.3

As seen in Table 1, Model 1 gives a higher accuracy of 76.68% whereas Model 2 gives an accuracy of 55.78%. The difference in accuracy is due to the difference in sizes of the training data when test data is uniform. Since the aforementioned computed accuracies only apply to the GPT-2 models, the total system accuracy of BGKnow will be greater.

5 Conclusion and Future Work

The paper introduces a hybrid chatbot framework which is obtained by integrating a neural-based model with a knowledge graph. This in turn provides the advantages of both techniques mentioned, i.e., neural-based models and knowledge graphs. The future aspects of the project include the attainment of higher accuracy via higher cosine similarity thresholds enabling the chatbot to be deployed in the real-world scenario as home healthcare robots or hospital enquiry assistant robots. This can be possible by self-curating more refined datasets. In addition, a mobile application can be developed encapsulating user profiles, user history, and suggestions/help sections. We plan to seek the aid of medical professionals to validate the answers generated by our system.

References

1. Alec R, Wu J, Rewon C, David L, Dario A, Ilya S (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9
2. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36(4):1234–1240
3. Joseph W (1966) ELIZA-a computer program for the study of natural language communication between man and machine. Commun ACM 9(1):36–45
4. Colby KM, Weber S, Hilf FD (1971) Artificial paranoia. Artif Intell 2(1):1–25

5. Shraddha S, Bhat S, Shubhashri K, Karnik S, Narender M (2021) Disease prediction chatbot. Int J Sci Res Comput Sci Eng Inform Technol 632–636. https://doi.org/10.32628/CSEIT2173172
6. Rashmi D, Deshpande Neeta A (2018) A medical chatbot. Int J Comput Trends Technol (IJCTT) 60(1):41–45
7. Athota L, Shukla VK, Pandey N, Rana A (2020) Chatbot for healthcare system using artificial intelligence. In: 2020 8th international conference on reliability, infocom technologies and optimization (trends and future directions) (ICRITO), pp 619–622. IEEE
8. Mahajan P, Wankhade R, Jawade A, Dange P, Bhoge A (2020) Healthcare chatbot using natural language processing. In: 8th international conference on reliability, infocom technologies and optimization (trends and future directions)
9. Divya S, Indumathi V, Ishwarya S, Priyasankari M, Kalpana Devi S (2018) A self-diagnosis medical chatbot using artificial intelligence. J Web Dev Web Des 3(1):1–7
10. Jurgita K-D (2020) A domain-specific generative chatbot trained from little data. Appl Sci 10(7):2221
11. Mohammed A, Ammar M, Hefny Hesham A (2023) Deep learning for Arabic healthcare: MedicalBot. Soc Netw Anal Min 13(1):71
12. Bao Q, Ni L, Liu J (2020) HHH: an online medical chatbot system based on knowledge graph and hierarchical bi-directional attention. In: Proceedings of the Australasian computer science week multiconference, pp 1–10
13. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training
14. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, p 30
15. Kalla D, Samiuddin V (2020) Chatbot for medical treatment using NLTK Lib. IOSR J Comput Eng 22
16. Mittal H, Srivastava V, Yadav SK, Prajapati SK (2021) Healthcare chatbot system using artificial intelligence. No. 6086. EasyChair

Video Integrity Checking Using X25519 and Nested HMAC with BLAKE2b

Linju Lawrence and R. Shreelekshmi

Abstract Latest developments in video editing or manipulation tools have facilitated effortless alteration of video content without any discernible traces left behind. As a result, it is imperative to subject video data to an integrity verification process before utilizing it as evidence. This paper presents a novel, lightweight approach for verifying the integrity of video data. The proposed method uses Hash-based Message Authentication Code (HMAC) and Elliptic Curve Diffie-Hellman Key Exchange utilizing Curve 25519 (X25519) with BLAKE2b. Nodes store video verification codes that are generated for video clips of a specific predetermined size. To enhance the security level, each node stores the nested HMAC value of the prior node. The integrity check involves the regeneration and comparison of nested HMAC of the node. The proposed method's experimental results demonstrate better performance in terms of both speed and security when compared to state-of-the-art methods. With minimal additional storage requirements, our method can identify any type of forgery on any video file, at any given time, by an authorized individual. Security analysis indicates that the method can withstand a range of attacks, such as timing attacks, key substitution attacks, side channel attacks, and brute force attacks.

Keywords Key exchange algorithm · Hash-based message authentication · Video integrity

L. Lawrence (✉)
Department of Computer Science and Engineering, College of Engineering Trivandrum Affiliated to APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala 695016, India
e-mail: linjulawrence680@cet.ac.in

R. Shreelekshmi
Department of Computer Applications, College of Engineering Trivandrum Affiliated to APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala 695016, India
e-mail: shreelekshmi@cet.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_31

1 Introduction

With the availability of affordable and easy-to-use video editing software, combined with the development of sophisticated forgery methods [1–3], digital video content can now be altered to a degree where it becomes nearly impossible to differentiate from the original material. Video evidence is commonly sourced from a variety of devices, including but not limited to Closed Circuit Television (CCTV), Accident Data Recorder (ADR), digital cameras, and mobile phones. Since video data can be easily manipulated using video editing software without leaving any discernible traces, it is essential to conduct an integrity verification process before presenting such evidence to any authority.

The remaining part of this paper is organized as follows: Sect. 2 discusses related works relevant to the proposed scheme and Sect. 3 outlines the relevant algorithms and presents our proposed method for video integrity verification. Section 4 offers a comprehensive evaluation of the proposed method, including a detailed experimental validation, security analysis, and comparison with existing methods. In Sect. 5, we conclude the paper by summarizing our findings and highlighting future research directions that can build on the proposed method.

2 Related Works

To verify the integrity of video data, two approaches can be utilized: active and passive [4]. Adding watermarks, fingerprints, and digital signatures to the video data is the primary focus of the active approach. The passive approach to detecting tampering involves analyzing compression artifacts, noise residues, and other structural changes that arise from manipulations.

Sarala et al. [5] propose a method that utilizes Elliptic Curve Cryptography combined with randomized hashing for the integrity verification of video content. The video data recorded with a predetermined length (video segments) is randomized with a uniquely generated random value followed by hashing. The hash algorithm uses a randomly generated initialization vector, which is created by using a secret key. The key and the output from the randomized hashing are combined and encrypted using an ECC encryption algorithm called the Elliptic Curve Integrated Encryption (ECIES) which integrates Elliptic Curve Diffie- Hellman Algorithm and Advanced Encryption Standard.

Sarala et al. [6] propose a method which combines ECIES with message authentication code based on hashing (ECIESH) in blockchain framework. The video segment is keyed hash at the time of recording and stored in order. During verification process, the same sequence operations is applied to the video data and the generated hash value is compared with the hash in the blockchain. This method is tested on various forgeries such as insertion, deletion, and copy-paste. This integrity verification method is more robust against several attacks. This method is tested on five publicly available videos segments [7–11].

Sowmya et al. [12] developed a method for the detection of inter/ intra-frame video forgery based on content-based signatures (CBSs). The video integrity verification is done by generating a unique message digest of 128-bit from variable-length video data, which is taken as a fingerprint. The spatial/ temporal level changes in the video content will result in a different fingerprint. No one will be able to recreate the original content by only knowing the signature as the signature generated from the combination of spatial and temporal fingerprints. This technique is tested and verified in benchmark data sets SULFA [13] and Derf's Collections [14]. The video sequence clip is split into frames and the local key points are used to uniquely represent the content of each frame. The features used by this method are the positions of these extracted key points. Then, the centroid/ center of gravity of these features is obtained and for each frame content-based signature is generated. Coupling the spatio-temporal features of the generated signatures and the centroid of this acts as the fingerprint of the video. For verification, these signatures are stored in a database. This method tested alterations such as frame deletion, shuffling, insertion, and object removal. This method cannot locate the spatial parts that have been manipulated. If the storage area where the signature is kept is compromised, the method fails.

Linju Lawrence and R. Shreelekshmi [15] proposed a method that combines the benefits of an Elliptic Curve Digital Signature and blockchain (CS-ECDSA). The method does not concern about the frame types or coding artifacts and applicable to video of any type and can detect forgeries of any kind. Every block comprises of the signature of the current segment and the signature of the prior block. The private keys are randomly generated and public keys for verifying these signatures are stored in the blockchain. This method is tested on the publicly available video segments [7–11] and a few of the videos from the datasets SULFA [13], Derf's collections [14], VIRAT [16] are also examined.

Our method for verifying the integrity of videos involves utilizing the Elliptic Curve Diffie-Hellman Key Exchange Algorithm, specifically with Curve 25519, and implementing a Hash-based Message Authentication Code using BLAKE2b for the video clips. The main features of our approach are outlined below.

1. Faster video integrity verification compared to existing state-of-the-art approaches utilizing X25519 for key exchange and BLAKE2b for hash generation.
2. The proposed method achieves greater security compared to state-of-the-art methods with the use of X25519, HMAC with BLAKE2b, Password-Based Key Derivation Function (PBKDF), and ephemeral public keys for video verification code generation in the nodes.
3. 100% detection of any kind of forgery on any type of videos by an authorized person from anywhere at any time by utilizing X25519 for secret key generation with ephemeral keys and linking the nodes using nested HMAC.

3 Proposed Scheme

3.1 Backgorund

X25519 X25519 is an Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithm [17]. It is widely used for secure key agreement in various cryptographic protocols and applications. X25519 is based on the elliptic curve Curve25519 [18], which is defined over the prime field of $2^{255} - 19$. The design of Curve25519 aims to achieve a high level of security while maintaining efficiency. In the X25519 key exchange protocol, private key is a random 256-bit integer, while the public key is derived by performing scalar multiplication on a generator point with the private key. The resulting public key is a 256-bit value on the elliptic curve. During the key exchange process, shared secret value generated by performing scalar multiplication of their own private keys and the received public key. The stages involved in an X25519 key exchange are as follows:

- The private key Pr_a or Pr_b that each side creates is a random number between 1 and $p - 1$ where $p = 2^{255} - 19$.
- Each side determines their corresponding public key by using generator point G and their private keys. Party 1 computes public key Pu_a as

$$\mathrm{Pu}_a = \mathrm{Pr}_a G \tag{1}$$

  Party 2 computes public key Pu_b as

$$\mathrm{Pu}_b = \mathrm{Pr}_b G \tag{2}$$

- The two parties then communicate across the unsecured channel to exchange public keys.
- When both parties have obtained the other party's public key, they multiply their private keys together after receiving the other party's public key. The resulting point s is used as the shared secret. Party 1 calculates s as

$$s = \mathrm{Pr}_a \mathrm{Pu}_b \tag{3}$$

  and party 2 calculates s as

$$s = \mathrm{Pr}_b \mathrm{Pu}_a \tag{4}$$

- This point is converted into a binary representation and used as a cryptographic key for encryption, authentication, or other purposes.

3.2 Proposed Scheme

We present a novel method for verifying and detecting forgery in video content. Our approach leverages the advantages of the Diffie-Hellman key exchange, utilizing Curve25519 (X25519), in combination with the robust BLAKE2b hashing algorithm [19]. The primary objective of this method is to ensure the integrity and authenticity of the video content itself, rather than delving into the specifics of frame types or coding parameters. Our method offers broad applicability, making it suitable for verifying any type of video content and capable of detecting various forms of forgeries. Video clips are recorded at regular intervals of a few minutes. The path specifying recorded video clip is stored in a structure called node. Each node contains header and data part. The header part includes ephemeral public key for each node and nested HMAC for linking with adjacent node, which is set to null in the first node. The nodes are created in chronological order while recording videos. Figure 1 shows the overall structure of the proposed method which comprises of video verification code generation and video verification code validation.

Video Verification Code Generation Consider k is a large prime number and Z_k is finite field with number of points divisible by k. E is curve 25519, a Twisted Edwards curve over the finite field Z_k. G is generator point of the curve E and q is subgroup order of the curve. To generate the common public key–private key pair for all nodes, a private key is selected at random and used in conjunction with the curve's parameters by authorized party. The public key, Pu is computed by multiplying the private key, Pr with the generator point G on the curve. The resulting public key is used by each node to generate secret key at the video verification code generation phase, while the private key remains confidential and used at the video verification code validation phase. The process of video verification code generation is depicted using Algorithm 1. Besides the common key pair, ephemeral key pair is generated for each node using random seed value. The additional memory required is negligible in comparison to the size of the video content. This method provides advanced security with a minimal key size. The seed undergoes hashing, and then the first 256 bits of the hash value is derived as the private key. The private key will always belong to the same subgroup of points on the curve and that the private keys will always have the same bit length to protect from timing-based side channel attack.

Ephemeral public key for each node, EPu is generated by the multiplying corresponding private key, EPr with generator G. Secret key $Secret$ is generated by scalar multiplication of EPr and common public key Pu. An intermediate HMAC Code for Integrity Verification (CIV) of video content, V_{node} is generated using secret key $Secret$ as salt and BLAKE2b hash function. PBKDF used for strengthening and randomizing the CIV using ephemeral public key stored in prior node, EPu_{prior}. $NestedHMAC$ is generated by finding the HMAC of the $RandomizedCIV$ using $NestedHMAC$ from prior node as key.

Video Verification Code Validation Video verification is done by comparing stored $NestedHMAC$ in each node and corresponding calculated $NestedHMAC$. The verifier holds the common private key Pr provided by the authorized party. The

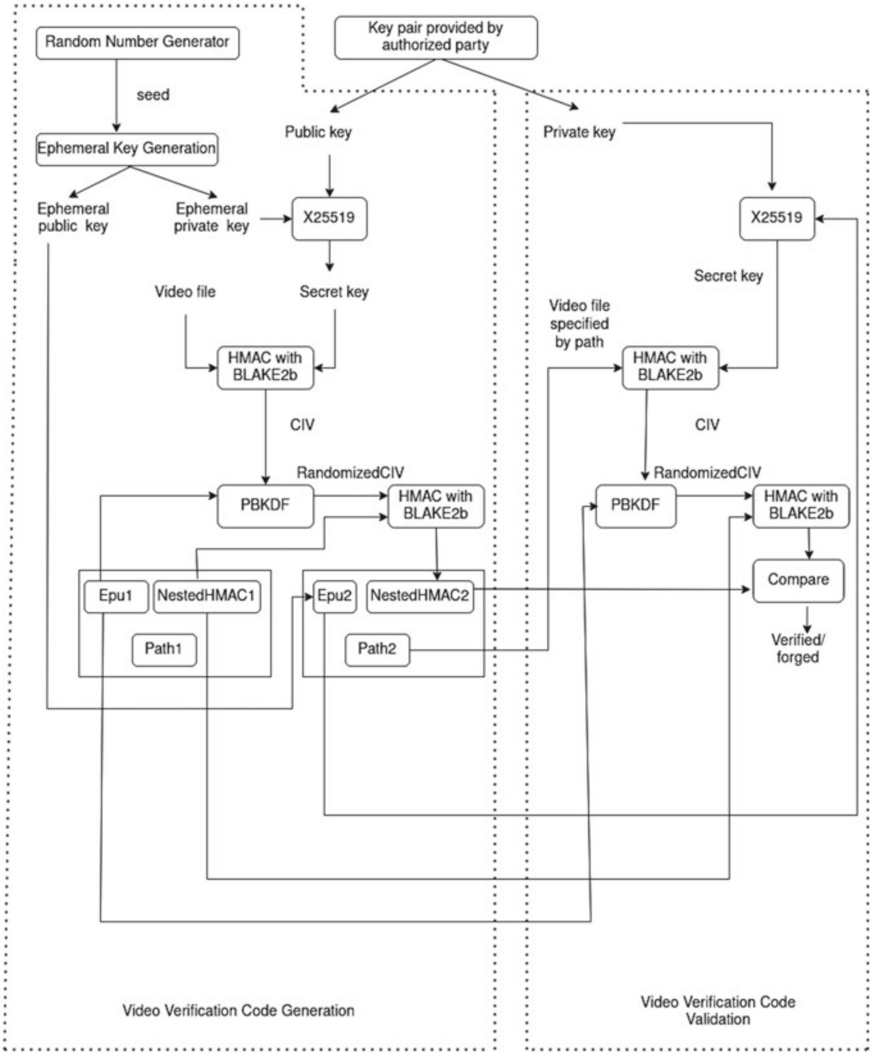


Fig. 1 A schematic block diagram of proposed mechanism

secret key for verification, is generated by scalar multiplication of Pr and Ephemeral public key from each node, EPu. Intermediate HMAC Code for Integrity Verification (CIV) of video content in each node is generated using BLAKE2b hash function with secret key, $Secret$ as salt. CIV is randomized by using PBKDF with EPu_{prior} as salt. $NestedHMAC$ is generated by finding the HMAC of the $RandomizedCIV$ using $NestedHMAC$ from prior node as key. The integrity of the video content is verified by the comparison of $NestedHMAC$ and calculated $NestedHMAC$ in the each stored node. If these two HMAC values are not equal the video is considered to be forged.

Algorithm 1 $Video\_Verification\_Code\_Generation$

```
Input
$E, G, q, V_{node}, Pu.$
Output
$EPu, NestedHMAC.$
repeat
    For each $V_{node}$ do.
    ▷ Generate $seed$ of 256 bits randomly.
    $seed \leftarrow \{0, 1\}^{256}$
    $H(0, ....., 511) \leftarrow SHA512(seed)$
    ▷ Initialization of private key randomly.
    $EPr \leftarrow H(0, ....., 255)$
    ▷ Generation of ephemeral public key.
    $EPu \leftarrow EPr * G$
    ▷ Computation of shared secret.
    $Secret \leftarrow Pu * EPr$
    ▷ Generation of Intermediate Code for Integrity Verification.
    $CIV \leftarrow HMAC(BLAKE2b, Vnode, Secret)$
    ▷ Randomization of CIV using Password Based Key Derivation Function.
    $RandomizedCIV \leftarrow PBKDF(CIV, EPu_{prior})$
    ▷ Generation of NestedHMAC.
    $NestedHMAC \leftarrow HMAC(BLAKE2b, NestedHMAC_{prior}, RandomizedCIV)$
until
```

Algorithm 2 $Integrity\_Verification$

```
Input
$Pr, V_{node}, EPu$
Output: Video data integrity verified or not.
repeat
    For each node do.
    ▷ Computation of shared key
    $Secret \leftarrow Pr * EPu$
    ▷ Generation of Intermediate Code for Integrity Verification at verification side.
    $CIV \leftarrow HMAC(BLAKE2b, V_{node}, Secret)$
    ▷ Randomization of $CIV$ using Password Based Key Derivation Function.
    $RandomizedCIV \leftarrow PBKDF(CIV, EPu_{prior})$
    ▷ Generation of Nested Code for Integrity Verification for comparison.
    $NestedHMAC \leftarrow HMAC(BLAKE2b, NestedHMAC_{prior}, RandomizedCIV)$
    ▷ Comparison of HMAC stored in the node and calculated $NestedHMAC$ at the verifier side.
    if $NestedHMAC$ in the node $\neq NestedHMAC$ then
        Video content is forged
    end if
until
```

4 Experimental Results

In order to evaluate our method, we utilize five distinct video segments, which are publicly available through sources [7–11]. Each video segment has a resolution of 1280 × 720 pixels. Additionally, we evaluate the performance of our method using several videos obtained from benchmark datasets, including VIRAT [16], SULFA [13], and Derf's collections [14]. The frame rate of the videos is set at 30 frames per second. To create tampered videos for testing purposes, we employ AVS Video Editor [20] to remove, copy, or insert frames within the video segments. Our experimental setup involves a PC equipped with an Intel Core i7-45 U CPU@1.8 GHz×4 and 12 GB RAM. The test videos are encoded using the H.264/AVC video codec provided by FFMPEG [21]. The OpenSSL cryptographic library is also utilized in our evaluation process.

4.1 Performance Evaluation

In Table 1, we present a performance comparison between our method and state-of-the-art techniques, ECIESH and CS-ECDSA on various test videos of different sizes. Encoding time is the time taken to generate verification code and verification time is the time taken to verify the verification code in the node. The performance evaluation shows that the proposed method on average is about 53.5% faster than ECIESH and 28.65% faster than CS-ECDSA in terms of encoding time and 61% faster than ECIESH and 40% faster than CS-ECDSA in terms of verification time.

Fig. 2 showcases the comparison of total execution time, encompassing both the encoding time and verification time, for the X25519 algorithm and the Conventional Elliptic Curve Diffie-Hellman key exchange algorithm (ECDH). The evaluation is conducted on different test videos with varying sizes. For all videos, it is evident that the total execution time for ECDH is substantially higher than that of X25519. X25519 uses shorter key sizes compared to traditional algorithms like Diffie-Hellman. Smaller key sizes also contribute to faster computation and reduced storage requirements.

The total execution time for proposed method with SHA-512 and BLAKE2b compared in Fig. 3, on different test videos with varying sizes. The results demonstrate the performance benefits of using BLAKE2b, which offers faster execution times while maintaining similar security strength as SHA- 512. BLAKE2b is about 1.47 times faster than SHA- 512.

Table 1 Comparison of the proposed method with recent state-of-the art methods

Video Name	Encoding time (in milli seconds)			Verification time (in milli seconds)		
	ECIESH	CS-ECDSA	Proposed method	ECIESH	CS-ECDSA	Proposed method
Video1 [11]	12.2	8	5.5894	12	7.7	4.543
Video2 [10]	18.4	11.9	8.55	18.4	11.7	6.903
Video3 [8]	86	56.7	40.738	86	56.5	34.594
Video4 [9]	165	110	79.033	163	109	66.49
Video5 [7]	42.8	27	19.399	42	26.9	16.409
Video6 [13]	32.4	21	14.672	31.6	20.6	12.2013
Video7 [13]	16.7	10.8	7.75966	16.1	10.5	6.195
Video8 [16]	48	31	22.273	47.5	30.5	18.605
Video9 [16]	13.8	9	6.288	13.6	8.8	5.193
Video10 [16]	18.8	12.3	8.8373	18.5	12.1	7.139
Video11 [14]	182.6	120.1	86.2903	182.67	118.6	72.939
Video12 [14]	49.3	32.3	23.2071	48.8	32	19.238

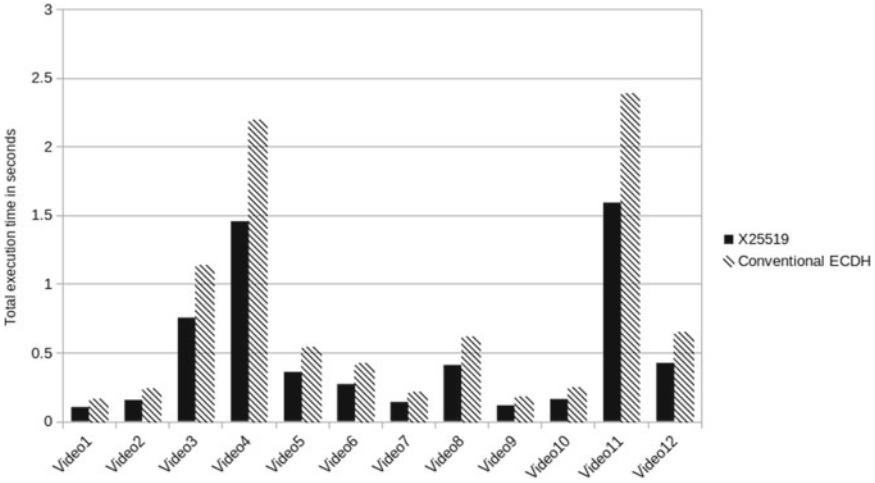


Fig. 2 Comparison of total execution time for conventional ECDH and proposed method with different test videos of varying sizes

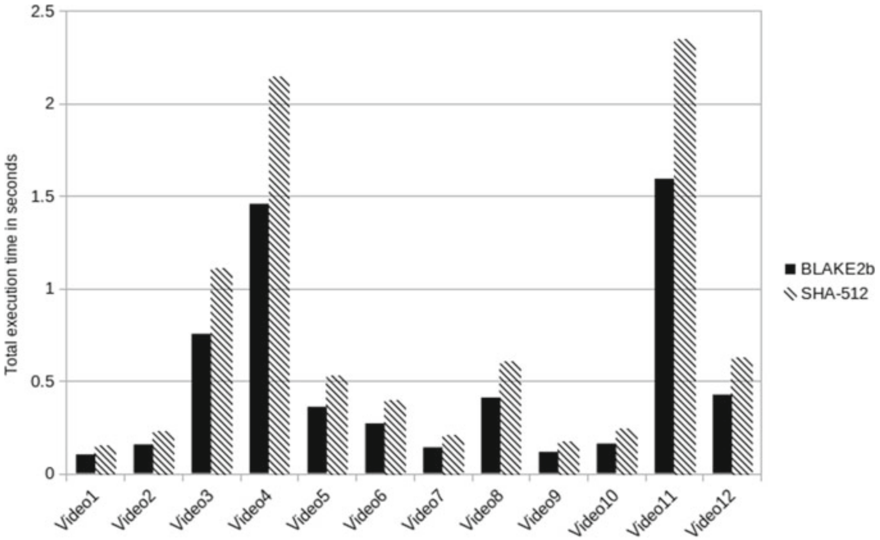


Fig. 3 Comparison of total execution time for SHA-512 and BLAKE2b with different test videos of varying sizes

4.2 Security Analysis

PBKDF used to strengthen and randomize intermediate verification code by making them resistant to brute force attacks. X25519 is resilient against various cryptographic attacks, including those based on the discrete logarithm problem [22]. The specific properties of Curve25519 make it difficult for attackers to calculate the private key from the public key, ensuring the security of the key exchange process [18]. The hardness of the discrete logarithm problem ensures that even if an attacker substitutes a different public key, they would not be able to compute the shared secret key [22]. Thus avoid key substitution attacks [23].

X25519 implementations typically incorporate constant-time techniques, which prevent timing attacks [24]. This ensures that an attacker cannot gain information about the private key through timing variations in the algorithm's execution. X25519 ensures that individual operations, such as additions and multiplications, are independent of secret data. This prevents side channel attacks that attempt to extract information by analyzing correlations between different operations. BLAKE2b, recognized for its exceptional speed, is a cryptographic hash function that generates a 512-bit digest size [19]. Remarkably, it provides an equivalent level of security strength as SHA-512, a widely recognized and adopted hash function.

Our method ensures the prevention of video data modification through the utilization of nested Hash-based Message Authentication Code (HMAC) for each node. To enhance security, a unique ephemeral key is generated for each node, adding an additional layer of protection. Each node is connected to the prior node through two distinct ways: one utilizing the ephemeral key of the prior node, and the other involving the prior node's nested HMAC. This interconnected structure makes it significantly challenging to modify individual nodes without detection or disruption to the overall integrity of the video data.

5 Conclusion

A novel video integrity verification method that combines X25519 and HMAC with BLAKE2b is introduced. The approach involves the generation of nodes corresponding to each video clip, which includes the ephemeral public key of each node and the nested HMAC value. The nested HMAC value is created again during the course of verification and matched with the value that already exists in the node. Experimental findings show how adaptable our proposed method is since it successfully applies to videos of any kind and can identify different kinds of forgeries by authorized individual. Our method utilizes X25519 with BLAKE2b, which has advantages over other methods including lower memory usage due to smaller key size, faster execution, and a higher level of security. Furthermore, we demonstrate through security analysis that our integrity verification approach is resistant to a variety of attacks, including side channel attacks, timing attacks, key substitution attacks, and brute force attacks.

The proposed method exhibits great potential for video integrity verification in devices like CCTVs and Accident Data Recorders, which necessitate rapid verification of integrity and robust protection against tampering attempts. Furthermore, the method's minimal memory requirements make it an appealing and practical solution for implementation in such devices.

Acknowledgements University Grants Commission, Government of India, for granting research fellowship.

References

1. Hays J, Efros AA (2007) Scene completion using millions of photographs. ACM Trans Graph (ToG) 26(3):4–es
2. Kwatra V, Schödl A, Essa I, Turk G, Bobick A (2003) Graphcut textures: image and video synthesis using graph cuts. ACM Trans Graph (tog) 22(3):277–286
3. Patwardhan KA, Sapiro G, Bertalmío M (2007) Video inpainting under constrained camera motion. IEEE Trans Image Process 16(2):545–553
4. Singh RD, Aggarwal N (2018) Video content authentication techniques: a comprehensive survey. Multim Syst 24:211–240
5. Ghimire S, Lee B (2020) A data integrity verification method for surveillance video system. Multim Tools Appl 79:30163–30185
6. Ghimire S, Choi JY, Lee B (2019) Using blockchain for improved video integrity verification. IEEE Trans Multim 22(1):108–121

7. 029-realistic beautiful flower painting timelapse by artistbrownlion-satisfying video-2.5 min. https://www.youtube.com/watch?v=2g8bS-_nNYE&t=7s/. Accessed April 2021
8. 1 min of nature footage-4k (ultra hd). https://www.youtube.com/watch?v=WLKJnHu0GC4/. Accessed April 2021
9. 4k video ultra hd-epic footage. https://www.youtube.com/watch?v=od5nla42Jvc/. Accessed April 2021
10. Nature in 30 seconds. https://www.youtube.com/watch?v=MHna8CzxPLk/. Accessed April 2021
11. Real yellow car. https://www.youtube.com/watch?v=o2YNaYcwdbA/. Accessed April 2021
12. Sowmya K, Chennamma H, Rangarajan L (2018) Video authentication using spatio temporal relationship for tampering detection. J Inf Secur Appl 41:159–169
13. Qadir G, Yahaya S, Ho AT (2012) Surrey university library for forensic analysis (sulfa) of video content
14. Derf's collection. https://media.xiph.org/video/derf/. Accessed 6 Aug 2021
15. Lawrence L, Shreelekshmi R (2021) Chained digital signature for the improved video integrity verification. In: International conference on machine learning and intelligent systems (MLIS), pp 520–526
16. Oh S, Hoogs A, Perera A, Cuntoor N, Chen CC, Lee JT, Mukherjee S, Aggarwal J, Lee H, Davis L et al (2011) A large-scale benchmark dataset for event recognition in surveillance video. In: CVPR 2011. IEEE, pp 3153–3160
17. Dong J, Zheng F, Cheng J, Lin J, Pan W, Wang Z (2018) Towards high-performance x25519/448 key agreement in general purpose gpus. In: 2018 IEEE conference on communications and network security (CNS). IEEE, pp 1–9
18. Bernstein DJ (2006) Curve25519: new Diffie-Hellman speed records. In: Public key cryptography-PKC 2006: 9th international conference on theory and practice in public-key cryptography, New York, NY, USA, April 24–26, 2006. Proceedings 9. Springer, pp 207–228
19. Saarinen MJ, Aumasson JP (2015) The blake2 cryptographic hash and message authentication code (mac). Tech. Rep
20. Avs video editor. https://www.avs4you.com/avs-video-editor.aspx/. Accessed 5 Dec 2020
21. Ffmpeg. http://www.ffmpeg.org/. Accessed 20 Nov 2020
22. Galbraith SD (2012) Mathematics of public key cryptography. Cambridge University Press
23. Bohli JM, Röhrich S, Steinwandt R (2006) Key substitution attacks revisited: taking into account malicious signers. Int J Inf Secur 5(1):30–36
24. Brendel J, Cremers C, Jackson D, Zhao M (2021) The provable security of ed25519: theory and practice. In: 2021 IEEE symposium on security and privacy (SP). IEEE, pp 1659–1676

Cloud-Based Skin Cancer Classification: Training and Deploying a Model on AWS

Challa Koti Reddy, Chava Pavan Kumar, A. R. P. S. Gowtham, Rajkumar Maharaju, and Rama Valupadasu

Abstract Skin cancer has become one of the most dangerous and most common types of cancer in recent years. Skin cancers come in a variety of types, and identifying the type is crucial for treating the condition when it is still treatable. The dermatologist must also distinguish between skin conditions that affect the tissues on the top layer of the skin and cells of the skin cancer that develop in the epidermal layer of the skin. The current methods for identifying or categorizing skin cancer take a long time and can be painful for the patient due to potential side effects. There is extensive research going on in this area but the unavailability of the balanced datasets and small size of the datasets have become a hindrance. There are not many products like web applications which use deep learning models to identify and categorize the type of skin cancer. We have used the ConvNeXt Tiny deep learning model which is pre-trained on ImageNet. Once we had obtained good accuracy, we had used that model and created a web application which is hosted in the AWS Cloud. The dataset we used for our work is ISIC2018. It consists of seven classes of dermoscopic images of skin lesions which are of high resolution. It is an imbalanced data that has images collected from various clinical sites.

Keywords ConvNeXt Tiny · Skin lesion · Skin cancer · AWS · Classification · ISIC2018

1 Introduction

The skin is the outer layer of the body and is the largest organ in the human body. The skin's ectodermal tissues, which can have up to seven layers, protect the internal organs, muscles, bones, and ligaments beneath it. Skin serves as a protection between the outside world and the internals of the human body, it helps in maintaining body temperature and aids us for the perception of touch, cold, and heat. Skin lesions are defined as the areas of the skin that are abnormal in comparison with other areas of

C. K. Reddy (✉) · C. P. Kumar · A. R. P. S. Gowtham · R. Maharaju · R. Valupadasu
NIT Warangal, Warangal, Telangana 506004, India
e-mail: kotireddychalla2002@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_32

the skin. Infections that occur inside or on the skin are the primary causes of skin lesions. Cancer is characterized by abnormal cell growth that has the capacity to grow and spread to different body parts. One of the more harmful and dangerous forms of cancer is skin cancer. Skin cancer patients can only be treated if it is found when it is in the very early stages. By shielding the entire body, including the muscles and bones, skin plays a crucial part in the functioning of the human body. The earliest signs of skin cancer can be found by checking for wary changes in your skin. A lesion area is a medical term for a diseased area of skin. Skin lesions come in a wide range of types. Each skin lesion is divided based on the origin, or the kind of skin cells that gave rise to it. Melanocytic lesions, which, like melanoma, arise from melanocytes, play a crucial role in the production of the protein pigment known as melanin. Other skin cell types, such as basal or squamous cells, are the source of non-melanocytic lesions. Some of the broad classifications of types of skin cancer are as follows:

1. Melanoma
2. Non-melanoma.

The primary test required to diagnose and assess the severity of skin cancer among the available techniques is skin biopsy. A biopsy is required to determine whether a suspected skin lesion is cancerous and if it is cancerous then what type of skin cancer it is. To effectively diagnose the patient, it is necessary to determine the type of skin cancer. The biopsy involves either removing a small sample of tissue or the entire suspect mole. When it comes to identifying the type of skin cancer, even a dermatologist with training has an accuracy rate of less than 80% [5]. In addition to this error, the lengthy procedure, the lack of dermatologists with the necessary training in public healthcare systems, and complications from biopsies can also result in excessive bleeding, infection, skin numbness, etc., which makes it difficult to classify the type of skin cancer and delays diagnosis. With the aid of convolutional neural networks, we aim to develop a straightforward application that can be used as the main tool for accurately classifying the type of skin cancer while also requiring less time and resources. The occurrence of various types of skin cancer is also not equal, and this is impacting the availability of the balanced datasets of the skin cancer images. The size of datasets is also not large enough to train the deep learning models. The above-mentioned scenarios are the reasons that the research is not progressing as much as we intended. Our objectives are to reduce the time taken to detect the type of skin cancer, which will help the patients to get treatment before it advances to later stages. Improving the accuracy of the classification of skin cancer types using an imbalanced dataset. Developing a web application based on the best-performing model and hosting in the cloud so that anyone connected to the Internet can use it if they have the required image input file.

2 Dataset Used

For machine learning problems involving the classification of skin lesions, the ISIC2018 dataset [2] is frequently used. It consists of 10,015 dermoscopic images with high clarity and resolution of skin lesions gathered from various clinical sites. Melanoma, seborrheic keratosis, basal cell carcinoma, squamous cell carcinoma, actinic keratosis, benign keratosis, and vascular lesions are among the seven groups into which the images are divided (Fig. 1). The extensive and reliable tagging of the ISIC2018 dataset is one of its distinguishing qualities. Several skilled dermatologists label each image, assuring high levels of accuracy in the annotation process. Because of the abundant and trustworthy labeling data, the dataset can be used to train sophisticated machine learning models. The dataset also includes a variety of skin lesions, making it difficult to classify and a useful tool for academics and developers working on skin lesion classification tasks.

The ISIC2018 dataset has been used in several studies to produce cutting-edge outcomes for skin lesion categorization tasks. Convolutional neural networks (CNNs), among other deep learning architectures, have been used by researchers to develop precise and effective models. The ISIC2018 dataset has also been used for transfer learning, which involves optimizing previously trained models' performance using the dataset.

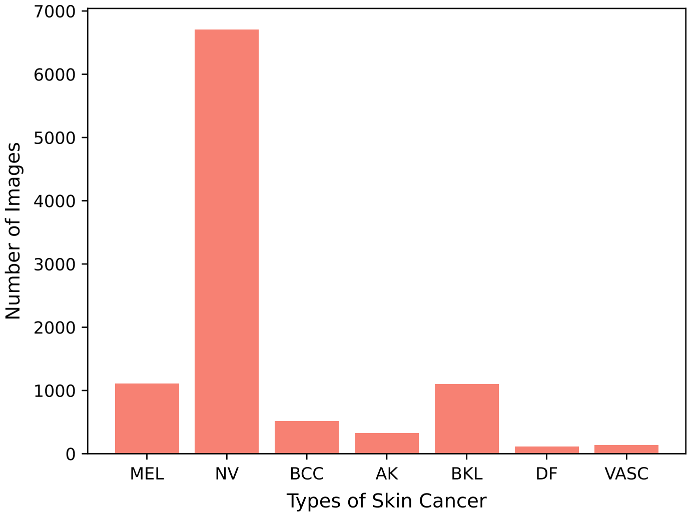


Fig. 1 ISIC2018 dataset distribution [9]

3 Model and Its Architecture

The model used in this paper to obtain the results is ConvNeXt Tiny [7], and the model architecture is given in Fig. 2.

A convolutional neural network (CNN) architecture called the ConvNeXt Tiny model was put forth in the article "A ConvNet for the 2020s" by Jing Yu, Zhe Wang, and Quoc V. Le. The model has been demonstrated to produce state-of-the-art results on a variety of image classification tasks. It is intended to be effective, scalable, and accurate.

The Swin Transformer architecture [6], a hierarchical convolutional transformer that has been demonstrated to be highly effective for image classification, serves as the foundation for the ConvNeXt Tiny model. While using fewer parameters than the Swin Transformer, the ConvNeXt Tiny model still achieves comparable accuracy. A stack of convolutional layers, each of which is followed by a batch normalization layer and a residual connection, make up the ConvNeXt Tiny model. Each of the hierarchically organized convolutional layers oversees processing an image's many levels of detail. In order to learn long-range dependencies in the image, the model additionally employs a hierarchical attention strategy.

The ImageNet dataset, which has over 14 million images in 1000 different classes, is used to train the ConvNeXt Tiny model. The stochastic gradient descent (SGD) optimizer with momentum is used to train the model. Additionally, the model is

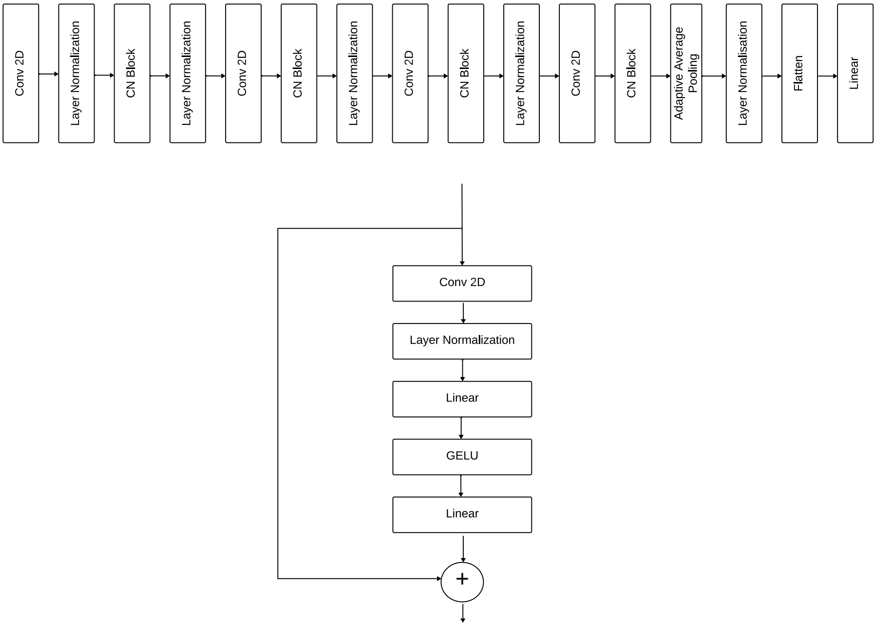


Fig. 2 Model architecture(top), ConvNeXt(CN) Block(bottom) [7]

improved using the ImageNet validation set to increase accuracy. A potential new architecture for image categorization is the ConvNeXt Tiny model. The model can produce even better outcomes in the future and is effective, scalable, and accurate.

4 Loss Function

The paper “Single Model Deep Learning on Imbalanced Small Datasets for Skin Lesion Classification” [9] introduced the multi-weighted new loss (MWNL), a loss function. To address the difficulties of unbalanced datasets and outliers in skin lesion classification, it is a modification of the cross-entropy loss function.

The following formula yields the MWNL:

$$\mathrm{MWNL}(z, y) = -\left(C_y^* \frac{1}{N_y}\right)^{\beta} \sum_{i=1}^{C} \mathrm{Loss}_i \tag{1}$$

$$\mathrm{Loss}_i = \begin{cases} (1 - p_i^t)^r \log(p_i^t) & p_i^t > T \\ G^* & p_i^t \leq T \end{cases} \tag{2}$$

$$G^* = (1 - T)^r \log(T) \tag{3}$$

$$C_y^{\alpha} = C_y \tag{4}$$

$$\beta = \begin{cases} 0 & E \leq E_1 \\ \left(\frac{E-E_1}{E_2-E_1}\right)^2 \alpha & E_1 < E < E_2 \\ \alpha & E \geq E_2 \end{cases} \tag{5}$$

It has been demonstrated that the MWNL is useful for deep learning skin lesion classification model training, particularly when the training set is unbalanced and the dataset contains outliers. It can be used with any deep learning framework and is simple to implement.

The MWNL also has a term intended to concentrate on outliers. Data points known as outliers differ significantly from the dataset’s other data points. Because they can lead to the models learning false patterns, outliers can be a problem for deep learning models. By including a term in the loss function that penalizes the model for making predictions that are too close to outliers, the MWNL solves this issue.

A promising loss function for classifying skin lesions is the MWNL. It can deal with the problems presented by unbalanced datasets and outliers.

5 Augmentation

5.1 RandAugment

An augmentation method called RandAugment [3] applies a set of image transformations at random to a given dataset in deep learning. The technique's goal is to increase the efficiency with which machine learning models perform image classification tasks. The goal of RandAugment is to produce a wide range of augmented images that accurately reflect the diversity found in the real environment. The method was first used in 2019 by Google researchers.

Two hyperparameters, N and M, determine the set of transformations. The amplitude of each transformation is specified by M, and the number of transformations to be applied is specified by N. The transformation is more forceful the larger the value of N and M. Rotation, translation, scaling, shearing, and flipping are just a few of the common image processing operations that are included in the collection of transformations employed by RandAugment. Every transformation is carried out using a magnitude that is chosen at random from a predetermined range.

5.2 MixUp

At the University of California, Berkeley, Hongyi Zhang et al. launched Mixup [11] in 2018 as an augmentation method for training deep neural networks. It is a straightforward method that effectively raises the generalization capacity of models by allowing them to train on input data samples that are mixed up.

The idea behind mixup is that by making the model learn from a variety of diverse examples, it would be better able to generalize to new data. This is because mixup promotes the model to learn more substantial characteristics that are not overfit to samples in the training set. Additionally, mixup reduces the likelihood of overfitting by regularizing the model.

5.3 CutMix

CutMix [10] is a variation of mixup, a popular data augmentation technique that creates new training instances by combining two images and the labels that go with them. By cutting and pasting patches from one image to another, CutMix expands on mixup to produce an even more powerful and varied collection of augmented images.

The CutMix technique forces the model to integrate data from various regions of the input images, which encourages it to learn more robust and discriminative features. Like mixup, it also has a regularizing effect by lowering the possibility of overfitting and enhancing the model's generalization capabilities.

6 TTA and 1Cycle Policy

Test time augmentation (TTA) is a technique commonly used in computer vision tasks, particularly in deep learning-based models. It involves applying data augmentation transformations to the test or inference data during the evaluation phase.

Typically, during the training phase, data augmentation techniques such as random rotations, translations, flips, and scaling are applied to the input images. These augmentations help the model generalize better by exposing it to a wider variety of training samples and reducing overfitting. However, during inference, the model is typically applied to the original, unaltered test data.

With TTA, instead of using the original test data for evaluation, multiple augmented versions of the test data are generated using the same transformations applied during training. The model's predictions are then obtained for each augmented version, and the final prediction is often obtained by averaging or voting over these predictions.

6.1 1Cycle Policy

1cycle policy [8] is a strategy used in training the deep learning models, where we optimize the learning rate and the momentum hyperparameters by changing their values between maximum and minimum in a cyclic way while training in each epoch. It contains two main phases, and they are:

1. **Increasing Phase**: In this phase, the momentum hyperparameters decrease from a higher value to a lower one and the learning rate increases from a lower value to a higher one. This helps the deep learning model to explore a range of learning rates and converge quickly and avoid the poor local optima.
2. **Decreasing Phase**: The learning rate in this phase gradually decreases from the high value to the low while the momentum values increase to the maximum. This helps the model to fine tune and enhances the generalization and stabilizing of the learned weights.

We use the 1cycle policy to optimize our model and help it converge faster, optimizing the momentum hyperparameters and learning rate, and improved generalization. This strategy has been proved to improve the models' performance across different tasks and datasets.

7 Methodology

7.1 Model Training

The model is trained using the strategy in Fig. 3 for 80 epochs with 1cycle policy. As the model is pre-trained on ImageNet, the final fully connected/linear layer is

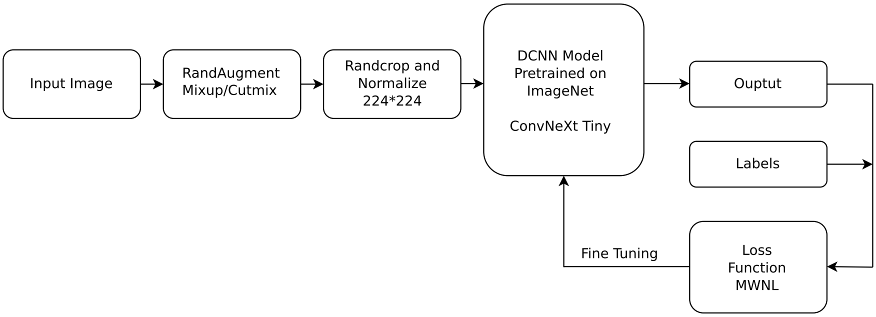


Fig. 3 Overview of the training strategy of the model [9]

not suitable for the task of classifying the types of skin cancer. So, the last layer is replaced with a linear layer with seven output nodes with random weights.

The model training is started by training the last layer, which has random weights, for a single epoch with 1cycle policy using a maximum learning rate of 4e-3. While training the last layer, the remaining layers of the model are frozen; i.e., the layers weights are not updated. After training the last layer, the remaining layers are unfrozen and trained for 80 epochs also using 1cycle policy, but with differential learning rates; i.e., each layer in the model is trained with different learning rates with a maximum learning rate of 2e-3 as part of the 1cycle policy.

In the testing phase, test time augmentation strategy is used, by making predictions on nine different augmentations of the test image, these predictions are then averaged to obtain the final prediction for the given test image.

7.2 Model Deployment

The trained PyTorch model is converted into an ONNX model, which can be easily deployed to the cloud or hardware devices. AWS is used to deploy the ONNX model, and to host the web application which provides an interface for users to get predictions on their images containing skin lesions. An API is also available for users, which can be used on hardware devices with Internet access. Figure 4 shows the AWS architecture used for model deployment. The model is run on Amazon Sagemaker serverless instance, the instance only runs when there is a request from the user, thus saving compute resources and the cost. To deploy the model to Sagemaker, an inference image, built using a docker file, is provided in Amazon Elastic Container Registry (ECR). The inference image contains the inference code and the required packages for running inference.

An AWS Lambda function is created to invoke the model endpoint created in Sagemaker. The return of the Lambda function should also include the necessary CORS headers, to be able to access from a web application. An API gateway endpoint

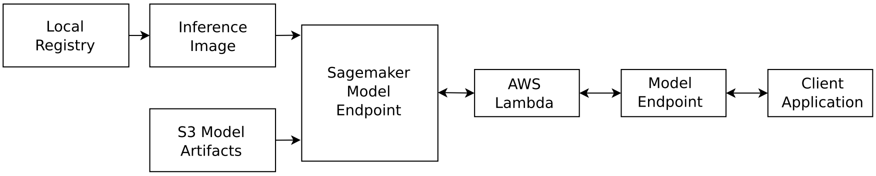


Fig. 4 AWS architecture[1]

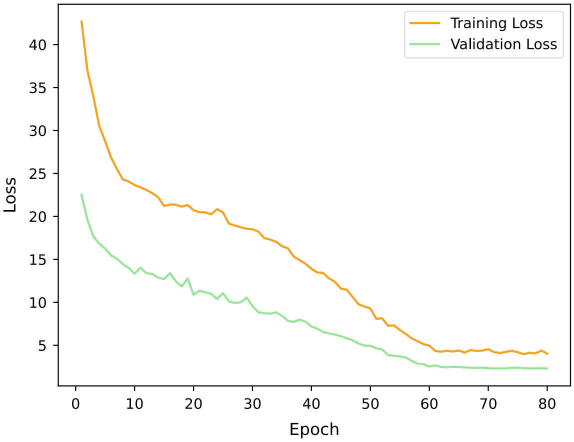


Fig. 5 Training and validation loss

is created to invoke the lambda function, which also acts as the entry point for other applications. A web application, made using React, is hosted on a S3 bucket using AWS Amplify. The web application has the option to upload the image for inference and returns the predicted type of skin cancer. While sending the image to the API endpoint, it is encoded using base64 and decoded at the server to obtain the image.

8 Results

8.1 Training

Figure 5 shows the training and validation loss values for each epoch, one can evaluate the model's ability to generalize and adjust if necessary to optimize its performance. In the above graph, both losses are decreasing which indicates that the model is learning effectively and will be able to generalize well to unseen data.

Fig. 6 Confusion matrix

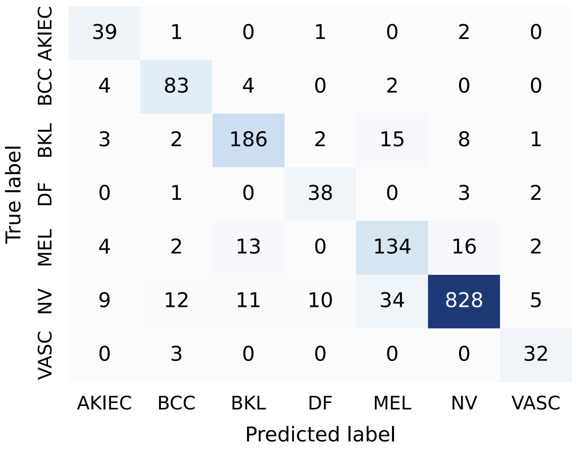


Table 1 Comparison with other models

S. No.	Model name	BACC (%)
1	seresneXt-50 [4]	80.1
2	SENet154 [4]	81.7
3	RegNetY-3.2G [4]	87.5
4	**ConvNext Tiny**	**87.67**

The bold signifies, the model that we used in this paper

8.2 Testing

Figure 6 depicts what is the predicted label of the test image by the model on the *x*-axis and what is the true label of the image on the *y*-axis. The more the images on the diagonal of the matrix the more accurately the model can classify the type of skin cancer.

The metric used to compare a model's performance for ISIC2018 dataset is balanced accuracy (BACC):

$$\text{BACC} = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \tag{6}$$

where TP denotes the true positives, and FN denotes the false negatives and C is equal to the number of classes that we are classifying.

The highest testing BACC obtained with TTA is 87.67%. The other models comparison is shown in Table 1.

9 Conclusion and Future Scope

Most of the models in ISIC2018 challenge leaderboard [4] used an ensemble of models, our model achieved similar performance by just using a single model. The ConvNeXt Tiny model outperformed the RegNet-3.2G model upon using the training strategy that we followed and obtained a balanced accuracy of 87.67%. In our observation, the ConvNeXt Tiny model trained faster than RegNetY-3.2G by using mixed precision training, even though the former is a larger model. Also, the time taken to tune the hyperparameters was lower as we have employed 1cycle policy and learning rate finder. We also deployed the model, and we trained on AWS Sagemaker. The users can get predictions on images using a web application, which is deployed using AWS Amplify.

The future scope includes improving the balanced accuracy of our model by using a better test time augmentation strategy. Instead of directly taking the average of predictions on the augmented images which considers each image equally, we can assign different weights to different augmented images. Also, different sets of transforms can be tried out on the validation set to obtain a better TTA strategy. Further, an ensemble of models can be used to improve the accuracy.

References

1. Analytics T (2020) Deploying a custom machine learning model as REST API with AWS SageMaker. Last accessed 5 Sept 2023
2. Codella N, Rotemberg V, Tschandl P, Celebi ME, Dusza S, Gutman D, Helba B, Kalloo A, Liopyris K, Marchetti M, Kittler H, Halpern A (2019) Skin lesion analysis toward melanoma detection 2018: a challenge hosted by the international skin imaging collaboration (ISIC). https://doi.org/10.48550/arXiv.1902.03368. ArXiv:1902.03368 [cs]
3. Cubuk ED, Zoph B, Shlens J, Le QV (2019) RandAugment: practical automated data augmentation with a reduced search space. https://doi.org/10.48550/arXiv.1909.13719. ArXiv:1909.13719 [cs]
4. ISIC (2023) ISIC challenge. https://challenge.isic-archive.com/leaderboards/2018/. Last accessed 3 Sept 2023
5. Kadampur MA, Al Riyaee S (2020) Skin cancer detection: applying a deep learning based model driven architecture in the cloud for classifying dermal cell images. Inf Med Unlocked 18:100, 282. https://doi.org/10.1016/j.imu.2019.100282. https://www.sciencedirect.com/science/article/pii/S2352914819302047
6. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. https://doi.org/10.48550/arXiv.2103.14030. http://arxiv.org/abs/2103.14030. ArXiv:2103.14030 [cs]
7. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A ConvNet for the 2020s. https://doi.org/10.48550/arXiv.2201.03545. http://arxiv.org/abs/2201.03545. ArXiv:2201.03545 [cs]
8. Smith LN (2018) A disciplined approach to neural network hyper-parameters: part 1—learning rate, batch size, momentum, and weight decay. https://doi.org/10.48550/arXiv.1803.09820. http://arxiv.org/abs/1803.09820. ArXiv:1803.09820 [cs, stat]

9. Yao P, Shen S, Xu M, Liu P, Zhang F, Xing J, Shao P, Kaffenberger B, Xu RX (2022) Single model deep learning on imbalanced small datasets for skin lesion classification. IEEE Trans Med Imaging 41(5):1242–1254. https://doi.org/10.1109/TMI.2021.3136682
10. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y (2019) CutMix: regularization strategy to train strong classifiers with localizable features (2019). https://doi.org/10.48550/arXiv.1905.04899. http://arxiv.org/abs/1905.04899. ArXiv:1905.04899 [cs]
11. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D (2018) mixup: beyond empirical risk minimization. https://doi.org/10.48550/arXiv.1710.09412. http://arxiv.org/abs/1710.09412. ArXiv:1710.09412 [cs, stat]

An Automotive ECU-Based Forward Collision Prevention System

Fariya Islam, Tajruba Tahsin Nileema, Fazle Rabbi Abir, Tasmia Tahmida Jidney, and Kazi A. Kalpoma

Abstract This study presents a software model that identifies vehicles in front of a test vehicle, measures distances, and classifies them as safe, slow speed, and brake. The classification determines which signal should be transmitted to the control unit. A specially tailored dataset of 2162 images from Bangladesh's roadside is used for the software model, which uses transfer learning to identify frontal objects, estimate distances, and classify distances according to control unit signals. Furthermore, two microcontrollers are used for hardware systems, utilizing an ultrasonic sensor to calculate distances, identify frontal objects, and show the expected outputs. The AURIX TC375 microcontroller board-control unit receives signals and triggers the appropriate output. This system can serve as a foundation for autonomous vehicle safety research.

Keywords Transfer learning · Test vehicle · Safe · Slow speed · Brake · Control unit · AURIX TC375

1 Introduction

Advanced Driver Assistance Systems (ADAS) are passive and active safety technologies created to prevent human errors while driving a variety of vehicles. The Forward Collision Warning System and Automatic Emergency Braking System are two features of ADAS technology that make up the Forward Collision Avoidance System. In order to avoid collisions or lessen the consequences, vehicles are designed with forward collision prevention systems. It keeps track of a vehicle's speed, the speed of the vehicle in front of it, and the distance between them so that it may warn the

F. Islam · T. T. Nileema (✉) · F. R. Abir · T. T. Jidney · K. A. Kalpoma
Ahsanullah University of Science and Technology, Tejgaon, Dhaka, Bangladesh
e-mail: ttajruba@gmail.com

T. T. Jidney
e-mail: 180204140@aust.edu

K. A. Kalpoma
e-mail: kalpoma@aust.edu

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_33

driver if the vehicles approach too close and also send the driver the proper control signals, such as slowing down or braking, if necessary. Road safety in Bangladesh is still an issue for which no ideal answer has been developed in this technology-based development period. According to [1] -

1. 2804 transport workers, 666 students, 114 members of law enforcement agencies, 117 teachers, 24 journalists, 31 doctors, 16 freedom fighters, and 133 leaders and workers of different political parties were killed due to road accidents.
2. The highest 28.59% of accidents occurred with the involvement of motorcycles, followed by 24.50% accidents involving with 13.95% buses, 11.42% battery-powered rickshaws, and easy bikes.
3. Out of all accidents, 5.67% took place in Dhaka city and 1.71% took place in Chattogram city.
4. In 2022, the highest death toll in a single day in road accidents was on July 29, when 44 people were killed and 83 injured in 27 road accidents.

The inspiration for this study has been developed as a result of the recent rise in road accidents in our nation. We thus want technology that can assist in controlling the vehicle in potentially deadly circumstances and lower the number of road accidents. In short, we make the following contributions to this paper:

1. We make a dataset consisting of vehicles that are available in Bangladesh, as there is a lack of suitable datasets for the Forward Collision Avoidance System.
2. In the case of the hardware system, we have used Infineon's powerful tricore microcontroller, the AURIX TC375, as the control unit.

2 Related Works

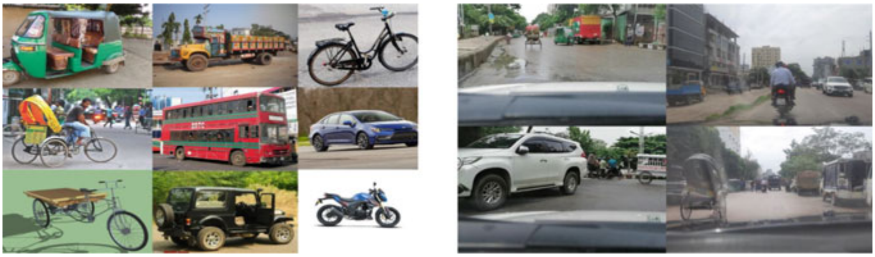
Modern vehicle operations demand safer roads, leading to increased demand for Advanced Driver Assistance Systems (ADAS) in the automobile industry. This section highlights the relevant research that helped us develop our model for vehicle identification, distance estimation, and hardware implementation. Abdelmalek Bouguettaya et al. [2] reviewed various deep learning architectures, datasets, and challenges faced in vehicle detection, offering suggestions for researchers and developers to select the most suitable method for their needs. Hassan Ramadan et al. [3] gave a plan on how their model could assist drivers and prevent auto accidents. The idea of making Yolo more effective at detecting objects on thermal videos and utilizing the findings of object detection to estimate the absolute distance between objects and the source camera was presented by them. Besides, Marek Vajgl et al. [4] proposed a method for improving YOLO's ability to estimate absolute distance using monocular camera information by expanding prediction vectors, sharing backbone weights, and modifying the distance estimation loss function. The research introduced two methods for handling distance: class-agnostic and class-aware, illustrating the relationship between object identification and distance measurement. Shivam

Kumar et al. [5] presented a practical solution for road car accidents using a CNN-based model and a windshield camera. The model monitored preceding vehicles and calculated the distance between them, alerting drivers of impending crashes if they got too close. Besides, Venkateswaran et al. [6] presented a forward collision warning system for Indian roads using YOLO CNN to detect moving vehicles. The system kept track of detected vehicles by assigning unique track IDs using the Hungarian algorithm and Kalman filter and estimated the distance between vehicles by mapping the camera image into a two-dimensional orthogonal top-down view using inverse perspective mapping. Samir A. Elsagheer Mohamed et al. [7] proposed an IoV-based technique that calculated safe driving distance and safe driving speed for connected vehicles, considering factors like weight, tires, length, speed, road type, and weather conditions. The technique calculated a safe driving distance, minimized gaps, avoided collisions, and maximized road utilization. Also, Hyunmin Chae et al. [8] created a model using Deep Reinforcement Learning (DRL) for a scenario in which a car was going to collide with a pedestrian who was crossing the urban road and whose behavior was unpredictable. Most of the time, the car stopped successfully about 5 m in front of the pedestrian. Tommaso Nesti et al. [9] presented a novel Ultrasonic Sensor (USS)-based object detection system for accurate low-speed scenarios, utilizing 3D point clouds, Bird's Eye View images, and deep neural network training. Experiments showed satisfactory performance across classic and custom metrics, bridging the gap between USS and established sensors. Moreover, Mohanad Abdul Hamid et al. [10] developed an automobile collision avoidance system using an Atmega 328p microcontroller and an ultrasonic sensor to detect motor gaps and alert drivers in danger range. Sensors accurately read shorter distances, but the system lacked real-time feedback due to environmental noise. On the other hand, Srinivasa Rao et al. [11] created a vehicle capable of navigating unknown environments and detecting obstacles using the Raspberry Pi 3+ and LIDAR. Without human assistance, this automated obstacle avoidance vehicle operated and prevented collisions.

3 Dataset

3.1 Data Creation

A suitable dataset is the primary prerequisite for this work. In order to do that, we have developed our own dataset. Google has been used as our source for the individual vehicle images (Fig. 1a). However, we have taken videos from our test vehicle of the roadside environment in Dhaka, Bangladesh, for this study (Fig. 1b).



(a) Individual vehicle images (b) Roadside environment images

Fig. 1 Sample images of custom dataset

Table 1 Dataset description

Image category	Image quantity
Bike	21
Bus	22
Car	27
Bicycle	20
Truck	31
Jeep	31
Van	13
Cng	8
Rickshaw	6
Roadside environment	1983
Total	2162

Many real-time videos of the frontal road environment that our test vehicle interacts with have been recorded using a mobile phone camera. The videos are then divided into images in VLC using a 0.10 frame rate interval. We have a total of 2162 images in our final dataset. For the training set, we have utilized 800 images; for the validation set, 200 images, and for the testing set, 1162 images. The final collection consists of 1983 images of real-time road conditions and 179 images of individual automobiles that are listed in Table 1.

3.2 Data Annotation

The image data is labeled using the graphical image annotation tool LabelImg. Using this tool, we have manually labeled 1000 YOLO-formatted images. We have drawn rectangular bounding boxes around objects in an image and then added class labels to the objects that should be identified in the testing phase, indicating the object's type or category. In Fig. 2, the images illustrate the data annotation process and environment.

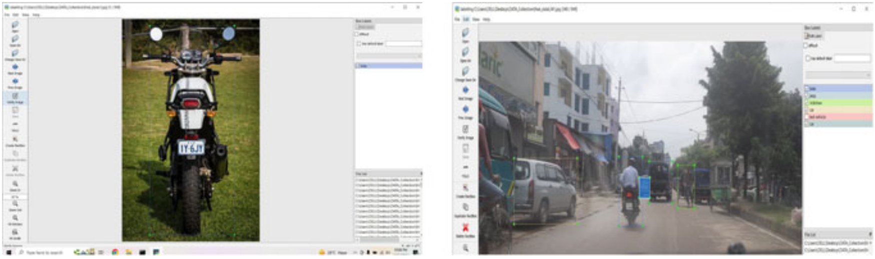


Fig. 2 Sample images of data annotation process of dataset

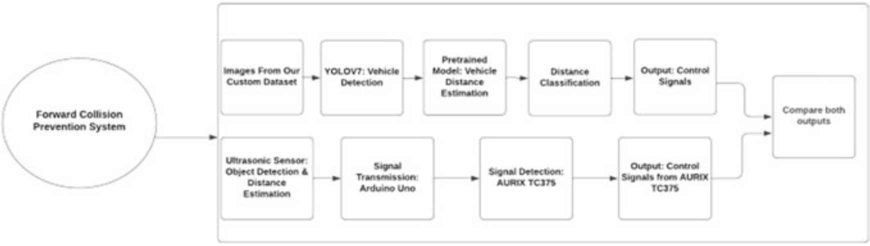

Forward Collision Prevention System
Images From Our Custom Dataset
YOLOV7: Vehicle Detection
Pretrained Model: Vehicle Distance Estimation
Distance Classification
Output: Control Signals
Compare both outputs
Ultrasonic Sensor: Object Detection & Distance Estimation
Signal Transmission: Arduino Uno
Signal Detection: AURIX TC375
Output: Control Signals from AURIX TC375

Fig. 3 Block diagram of proposed system

4 Proposed System

The system is made up of two systems: the software system and the hardware system. In the case of the software system, after data annotation, Yolov7, an object detection model, is being used to identify frontal objects from the test vehicle. The distances of the frontal objects that have been detected are then estimated using a deep learning pretrained model from [3]. The final stage for the software system involves the classification of distances as control unit signals that are comparable with the output of the hardware system.

In our hardware system, we have utilized two microcontrollers (Arduino Uno and AURIX TC375), an ultrasonic sensor, LEDs, jumper wires, and 100 Ω resistors. The Arduino Uno and ultrasonic sensor are coupled to detect frontal objects and measure their distances. We have measured the distances of frontal vehicles in inches for the hardware system because the ultrasonic sensor has a relatively limited detection range. The AURIX TC375 (control unit) and Arduino Uno's digital pins are then used to connect the two microcontrollers. The AURIX TC375 board receives the signals from the Arduino Uno, and the Arduino Uno transmits the encrypted signals to the AURIX TC375 board depending on the distance of the frontal object. The output signals are then shown using red, yellow, and green leds, which denote the signals of the control unit, such as brake, slow down, and safe. The proposed system in this study is shown in a block diagram in Fig. 3, which we have created using Lucidchart [13].

5 Methodology

5.1 Software System

Object Detection with YOLOV7: In this work, the YOLOv7 pretrained model is used to recognize the frontal objects of the test vehicle. Because preventing a forward collision requires the system to first recognize an object, we have used transfer learning to feed our annotated data to the YOLOv7 model, which we have then trained to recognize objects from 10 different classes. The vehicle classes are based on vehicles in Bangladesh. Bicycle, bus, person, car, bike, truck, jeep, van, CNG, and rickshaw are the 10 categories given. The output images of the YoloV7 model, shown in Fig. 4, are identified as objects by the object detection model along with their class and likelihood of belonging to that class. A car and a rickshaw are the frontal objects in Fig. 4a, respectively, with probabilities of 0.93 and 0.92. The image in Fig. 4b has two cngs with probabilities of 0.79 and 0.51. Two rickshaws are frontal objects in Fig. 4c, with probabilities of 0.93 and 0.92. A car is the frontal vehicle in Fig. 4d, with a probability of 0.91.

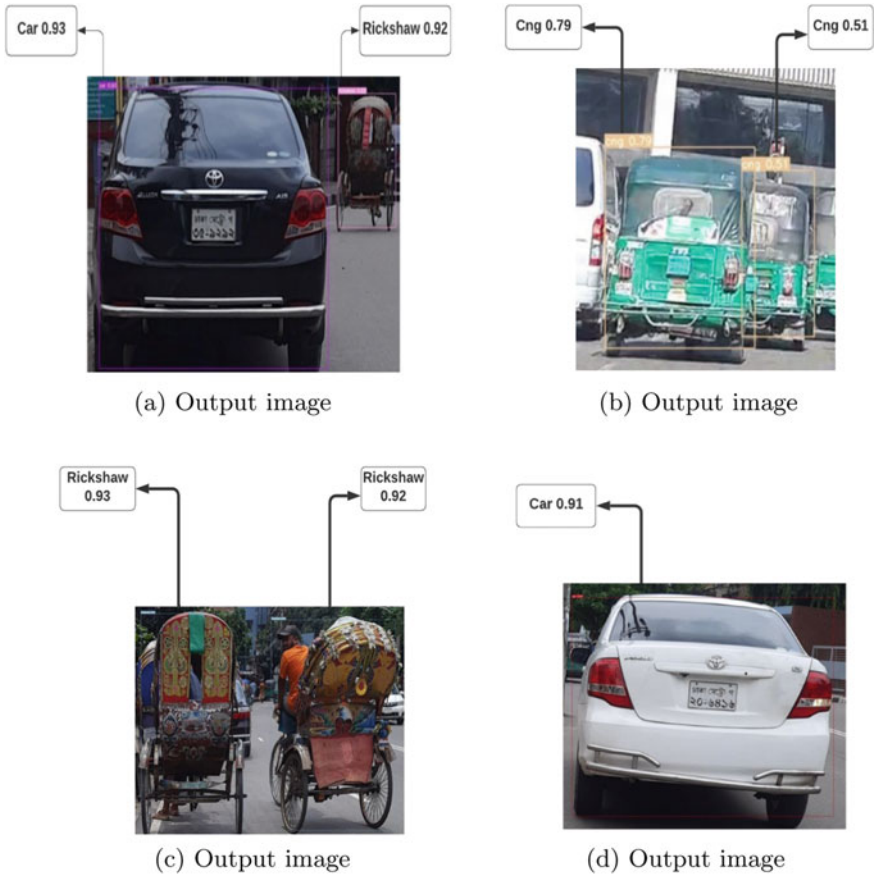
Distance Estimation: The distance is estimated from the camera to the frontal objects of the test vehicle. For distance estimation, a deep learning model has been used from [3] that takes in the bounding box coordinates of the detected objects and estimates the distance to the object. It has been trained on the KITTI dataset. KITTI is developed using a framework for autonomous driving. Figure 5a–d illustrate output pictures that display the class of an object that is detected, the probability of the detected object, and the estimated distance of frontal objects.

Distance Classification: In [8], they described how they used deep reinforcement learning to create a system to prevent a car from colliding with itself in the front. 1000 attempts later, their system often came to a stop within 5 m of the target. We have continued to classify the distances for the control unit after choosing 5 m as our critical distance from [8]. For this work, the classification is carried out based on a few circumstances. They are listed below:

1. The signal should read “brake” if the distance is five meters or less.
2. The signal should read “slow speed” if the distance is higher than 5 m but less than or equal to 15 m.
3. The signal should be “safe” if it is farther away than 15 m.

5.2 Hardware System

Object Detection and Distance Estimation: We have utilized an ultrasonic sensor (HC-SR04) for object detection and distance calculation. Four pins make up the sensor. They are GND, trig, echo, and Vcc. The Arduino Uno board’s GND, Vcc,



(a) Output image

(b) Output image

(c) Output image

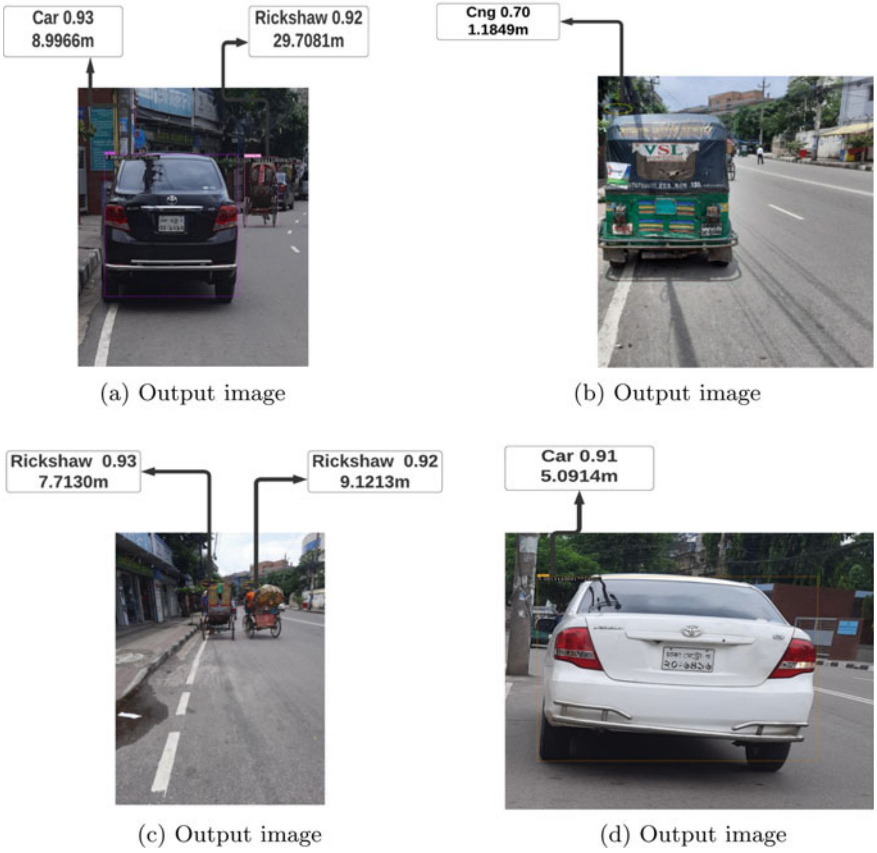
(d) Output image

Fig. 4 Output images of object detection phase

and digital pins 3 and 2 are used to link them. This sensor uses ultrasonic pulses to calculate distance. An ultrasonic wave is sent by the sensor head, which then picks up the wave that the target reflects back to it. The time elapsed between the emission and reception is measured by an ultrasonic sensor to determine the target's distance. The formula (1) is used by this sensor to determine distance.

$$s = \frac{(v * t)}{2} \tag{1}$$

where s is the distance from the frontal object, v is the wave's speed, and t is the amount of time it takes for the wave to return to the sensor. Because the wave must travel twice as far to detect the object and return a response to the sensor, the formula ($s = v * t$) is divided by two.



(a) Output image

(b) Output image

(c) Output image

(d) Output image

Fig. 5 Output images of distance estimation phase

Signal Transmission: The Arduino Uno board is used to transmit the signal. Inches are used as units of distance by the sensor when measuring distance. The Arduino Uno board measures the distances in inches and then sends signals to the AURIX TC375 board based on those distances. The board sends a signal to the AURIX board to turn on the red LED if the sensor detects a distance less than or equal to 5 in. The Arduino board sends a signal to the AURIX board to turn on the yellow LED if the distance is more than 5 in. but less than or equal to 15 in. Finally, the board sends a signal to the AURIX board to turn on the green LED if the distance is more than 15 in. Digital pins 4, 5, and 6 of the Arduino Uno board are used to link the Arduino board to the Aurix board.

Signal Detection: Digital pins 2, 3, and 4 on the AURIX TC375 board are used to detect or receive signals from the Arduino Uno board. The red LED is turned on as an output if the AURIX board receives a “HIGH” signal at digital pin 2. The yellow

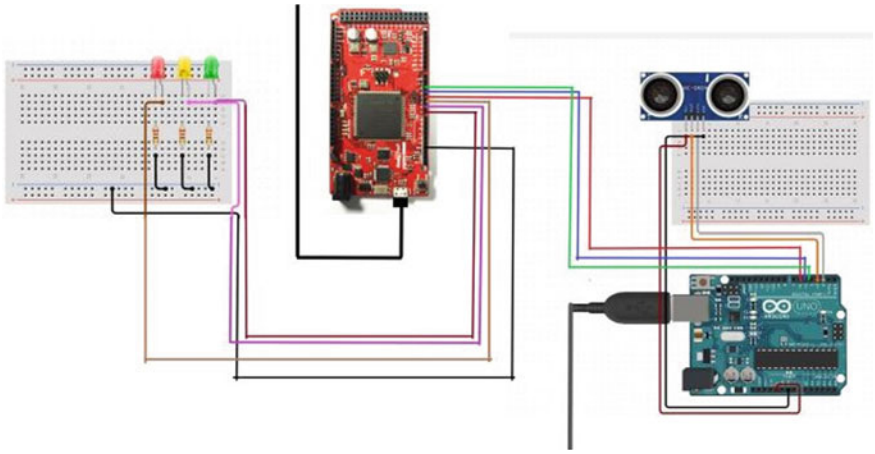


Fig. 6 Circuit design of hardware system

LED is turned on and the other two LEDs are kept off if the AURIX board detects a 'HIGH' signal at digital pin 4. The green LED is turned on as an output by the AURIX board when digital pin 3 receives a 'HIGH' signal.

Output of ShieldBuddy AURIX TC375 Board: Depending on the signal it receives from the Arduino Uno board, the AURIX board's digital pins 5, 6, and 7 are utilized to turn on the LEDs as an output. If the distance is less than or equal to 5 in., it turns on the red LED, signaling that the automobile has to be stopped. If the distance is greater than 5 in. but less than or equal to 15 in., it turns on the yellow LED to indicate that the automobile has to slow down. If the distance is greater than 15 in., it turns on a green LED to indicate that the vehicle may proceed safely.

In Fig. 6, the whole circuit design of the hardware system is shown. Two microcontroller boards (Arduino Uno and AURIX TC375), breadboards, an ultrasonic sensor (HC-SR04), three resistors (100 Ω each), three LEDs (red, yellow, and green), and wires are used to build the hardware system. The circuit design is created using Circuit.io. [12].

6 Results Analysis

6.1 *Software System*

Following object detection, distance estimation, and distance classification, we are able to classify each distance into categories, which are safe, slow speed, and brake. The detailed output data from a software model is shown in Table 2. This output information is taken directly from our code samples. The 'Object' column in Table 2

Table 2 Output of software system

Object	Frame	Xmin	Ymin	Xmax	Ymax	Distance	Output signal
1	1	1386	658	1635	980	14	Slow speed
2	1	1833	544	1920	1030	10	Slow speed
3	1	1710	658	1920	900	17	Safe
4	1	1626	552	1764	1024	11	Slow speed
5	1	0	523	1491	1076	11	Slow speed
–	–	–	–	–	–	–	–
9294	1162	1255	524	1405	874	17	Safe
9295	1162	1707	575	1917	958	14	Slow speed
9296	1162	1379	642	1452	770	3	Brake
9297	1162	901	462	1315	992	13	Slow speed
9298	1162	1448	672	1686	858	23	Safe

denotes each object in the frame. The ‘Frame’ column denotes the image number of the test dataset. The ‘Xmin’ and ‘Ymin’ values correspond to the bounding box’s bottom-left and top-left corners, respectively. The bounding box’s top-right corner has the coordinates ‘Ymax’, whereas the bottom-right corner has ‘Xmax’. The ‘Distance’ column reflects the predicted distance of the distance estimation model, whereas the ‘Output Signal’ depicts the actions that should be conducted depending on distances. Only the first five objects from the first frame and the last five from the last frame are displayed in Table 2. Between the first frame output and the last frame output information, there is additional information about the output of other frames and their objects, which is shown by dots.

6.2 Hardware System

The outcome of the entire hardware system is displayed in Table 3.

From Table 3, we can observe that all the distance values that are less than or equal to 5 in. show the red LED or brake signal as output. On the other hand, all the distance values greater than 5 in. but less than or equal to 15 in. show the yellow LED or slow down signal as output. Finally, when the distances are greater than 15 in., it shows the green LED or safe signal as output.

Table 3 Output of hardware system

Distances of object (inches)	LED output
4.23	Red
3.91	Red
2.08	Red
8.41	Yellow
7.88	Yellow
10.11	Yellow
16.14	Green
18.94	Green
19.04	Green

Table 4 Result discussion (I)

Software system		Hardware system	
Distance range	Output signal	Distance range	Output signal
distance(meters) ≤ 5	Brake	distance(inches) ≤ 5	Red LED
$5 <$ distance(meters) ≤ 15	Slow speed	$5 <$ distance(inches) ≤ 15	Yellow LED
distance(meters) > 15	Safe	distance(inches) > 15	Green LED

Table 5 Result discussion (II)

Software system		Hardware system	
Distance	Output	Output	Distance
3 m	Brake	Red LED	4.23 in.
10 m	Slow speed	Yellow LED	10.11 in.
17 m	Safe	Green LED	16.14 in.

6.3 Discussion

We can see from Sects. 6.1 and 6.2 that, in contrast to hardware systems, software systems have greater distance ranges. The hardware system’s ultrasonic sensor has a limited range for measuring distance. Table 4 displays the distance ranges and their corresponding output signals for both software and hardware systems. The critical distance for a software system is 5 m, but the critical distance for a hardware system is 5 in.. Likewise, the other distance ranges for control unit signals have to be scaled down for hardware systems as well. In the case of output signals from hardware systems, the ‘Red LED’ signifies the “Brake” signal from software systems, together with the “Yellow LED” and “Green LED”, which denote the ‘Slow speed’ and ‘Safe’ signals from software systems.

The relationship between the output signal and distance ranges of both systems is attempted to be seen in Table 5, which can serve as a visual representation of Table 4. The software systems display the signal “brake” at a distance of 3 m, which is less than the critical distance for software systems, which is 5 m. In the case of hardware systems, 4.23 in. display “Red LED” as output, which is less than this system’s 5 in. critical distance. Similar comparisons are shown in the following rows of data that follow those in Table 5’s first row. This analysis uses the information from Tables 2 and 3 for the two systems.

7 Conclusion

In this study, a system that determines whether to prevent a forward accident based on the critical distance between the test vehicle and the target frontal vehicle is constructed. A unique dataset is produced. Images of various automobiles and the surrounding roadside landscape are included in the custom dataset. Transfer learning is utilized to feed our training photos into the YOLOv7, which provides real-time object detection. Following the detection phase, this system determines the actual distance between the test vehicle and other frontal vehicles and applies multiclass classification to come to a control unit decision. Then, using two microcontrollers-an Arduino Uno and an AURIX TC375-we have created a hardware version of this system. The frontal objects have been identified, and their distance from the test vehicle has been measured using an ultrasonic sensor that is connected to an Arduino Uno. When the appropriate action has to be taken, it sends signals to the control unit (AURIX TC375). We have found the expected results for both systems. The distance is measured using a software model that produces precise values. But in the hardware system, the distance varies a little bit more than the real distance since the speed of sound is temperature-dependent and changes by around 0.17% for every degree Celsius. The predicted distance can occasionally fluctuate due to these variations, which also impact the travel time. Perhaps a sensor that is more effective and efficient can be used in place of an ultrasonic sensor to measure distances more precisely. In the future, we’ll utilize a better sensor that can detect larger distances, and we’ll connect the sensor directly to the ShieldBuddy Aurix TC375 Board. The software system serves as a representation of the theoretical analysis for this particular study field, where the custom dataset has been used. The hardware system displays the Forward Collision Prevention System in real time with real objects as test data. The development of this study field may be encouraged by the results presented in this work.

References

1. New Age Bangladesh: Road accidents in 2022. https://www.newagebd.net/article/190603/road-accidents-kill-9951-in-2022. Last accessed 2023-01-20
2. Bouguettaya A, Zarzour H, Kechida A, Taberkit AM (2021) Vehicle detection from UAV imagery with deep learning: a review. IEEE Trans Neural Netw Learn Syst 33(11):6047–6067. IEEE
3. Ramadan H, AbdElkader A, Khaled A, AbdElalem A, Mohamed W, Tawfik H (2021–2022) Vehicle detection and distance estimation. In: The 1st student conference
4. Vajgl M, Hurtik P, Nejezchleba T (2022) Dist-YOLO: fast object detection with distance estimation. Appl Sci 12(3):1354. MDPI
5. Kumar S, Shaw V, Maitra J, Karmakar R (2020) FCW: a forward collision warning system using convolutional neural network. In: 2020 International conference on electrical and electronics engineering (ICE3). IEEE, pp 1–5
6. Venkateswaran N, Jino Hans W, Padmapriya N (2021) Deep learning based robust forward collision warning system with range prediction. Multim Tools Appl 80:20849–20867
7. Mohamed E, Samir A, Alshalfan KA, Al-Hagery MA, Othman MTB (2022) Safe driving distance and speed for collision avoidance in connected vehicles. Sensors 22(18):7051
8. Chae H, Kang CM, Kim B, Kim J, Chung CC, Choi JW (2017) Autonomous braking system via deep reinforcement learning. In: 2017 IEEE 20th International conference on intelligent transportation systems (ITSC). IEEE, pp 1–6
9. Nesti T, Boddana S, Yaman B (2023) Ultra-sonic sensor based object detection for autonomous vehicles. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 210–218
10. Abdulhamid M, Amondi O (2020) Collision avoidance system using ultrasonic sensor. Land Forces Acad Rev 25(3):259–266
11. Baras N, Nantzios G, Ziouzios D, Dasygenis M (2019) Autonomous obstacle avoidance vehicle using lidar and an embedded system. In: 2019 8th International conference on modern circuits and systems technologies (MOCAST). IEEE, pp 1–4
12. Circuit.io. https://www.circuito.io/. Last accessed 2023-05-03
13. Lucidchart. https://www.lucidchart.com. Last accessed 2023-05-03

Credit Card Fraud Detection by Using Ensemble Method of Machine Learning

Nihar Ranjan, G. S. Mate, A. J. Jadhav, D. H. Patil, and A. N. Banubakode

Abstract Online transactions have become an essential aspect of life as universe becomes more technological and every industry leverages the web to grow enterprises. Online transactions have been increasing steadily, and this trend is expected to continue. Credit cards are a popular form of internet transaction, but with their widespread use comes a significant drawback: credit card fraud. Since banks are unable to screen every transaction, machine learning is essential to identifying credit card fraud. In our research, we used Kaggle to gather a dataset of 2,844,808 credit card transactions from a European Bank Dataset. There are 492 fraudulent transactions in it; to balance the dataset, we proposed hybrid resampling method; and for the detection of credit card fraud, Random Forest Algorithm is used. The assessment of the model is assessed based on accuracy, precision, recall, and F1-score. Our model shown fairly good results of 97.66, 98.85, 95.94, 97.37% for accuracy, precision, recall, and F1-score, respectively.

Keywords Machine learning · Credit card · Fraud detection · Random forest · Decision tree

N. Ranjan (✉) · G. S. Mate · A. J. Jadhav · D. H. Patil
Department of Information Technology, Rajarshi Shahu College of Engineering, Pune, India
e-mail: nihar.pune@gmail.com

G. S. Mate
e-mail: gsmate_it@jspmrscoe.edu.in

A. J. Jadhav
e-mail: ajjadhav_it@jspmrscoe.edu.in

D. H. Patil
e-mail: dhpatil_it@jspmrscoe.edu.in

A. N. Banubakode
MET COE, Mumbai, India
e-mail: abhijitsiu@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_34

1 Introduction

Over the last few years, online transactions have taken on a major significance on persons' life. From small-scale merchants to large corporations, all must unquestionably develop their online presence to reach a larger audience with their goods and services. Customers prefer online transactions over in-person purchases since the distance is no longer an issue. Online transactions often offer additional benefits, such as the ability for the client to manage their investments and create a purposeful budget. Eighty percent of Americans prefer cards versus cash, according to statistics [1]. By offering a variety of credit card features, such as easy credit, EMI availability, incentives and offers, flexible credit, purchase protection, and the ability to make international transactions fee-free, banks also hope to boost client engagement [2].

Like every coin has two sides, with more people depending on credit cards for their day-to-day transactions there arises a problem of credit card fraud. Data from the National Crime Records Bureau (NCRB)'s Crime In India 2020 report indicates that during the pandemic, instances of online financial theft involving credit or debit cards have surged by almost 225%, from 367 in 2019 to 1194 in 2020 [3]. According to the Reserve Bank of India (RBI) data, fraudsters stole a total of 615.39 crores from more than 1.17 lakh cases of credit and debit card theft over a ten-year period (April 2009–September 2019) [4].

2 Literature Survey

There have been various studies and many methods have been implemented for the detection of credit card fraud. Here, we are going to discuss the various algorithms and methods that have been studied and implemented in earlier studies. In [5], Logistic Regression, Decision Tree, Random Forest, Naïve Bayes, and the neural network approach ANN were employed as machine learning techniques. They concluded that the ANN model had the best precision and accuracy. In [6], Logistic regression, Decision Tree, and Random Forest were implemented. The Random Forest classifier outperforms the logistic regression and Decision Tree when all three methods are compared. In [7], the emphasis is mostly on real-world credit card fraud detection. The accuracy of identifying fraud can be enhanced with the proposed technique, which employs random forest algorithm. The precision, specificity, sensitivity, and accuracy of the procedures are used to assess their performance. Predictive models like random forest, XGBoost, and logistic regression have all been utilized in [8] along with a variety of resampling techniques. The trials' findings showed that random forest performed better than other models when combined with a hybrid resampling technique that included SMOTE and Tomek Links' removal. SVM, NB, KNN, Logistic Regression, and Random Forest algorithms are used in [9] to determine which algorithm is more effective at detecting credit card fraud. They decided on the Random Forest and KNN algorithms based on accuracy or the best value of MCC. The aim of [10] is to address the challenges posed by card fraud databases. They operate on datasets that are balanced and unbalanced. By using SMOTE to balance the dataset, Matthews Correlation Coefficient was the superior parameter to use. In [11], it has been stated how various machine learning techniques are applied to identify fraudulent transactions. It provides a general overview of machine learning techniques like SVM, Random Forest, Logistic Regression, Neural Network, and Decision Tree, as well as a quick overview of several fraud kinds. Additionally, they mentioned how more accurately forgeries may be detected using random forests. They have put forth a machine learning-based system in [12]. In addition to discussing logistic function and how data should be prepared for logistic regression, they concentrated on logistic regression techniques.

3 Proposed Model

The project's major goal is to detect credit card fraud, and machine learning algorithms are very helpful for the same as they are more accurate than the traditional methods. Machine learning algorithms are also self-learning and can identify patterns with minimal human intervention. Our proposed model architecture is shown in Fig. 1.

3.1 Data Collection

We are using a dataset obtained from Kaggle for this project. The dataset includes September 2013 credit card transactions made by cardholders across Europe. Out of 284,807 transactions over the course of two days, 492 scams are included in this dataset. The positive class (frauds) accounts for only 0.172% of all transactions in the extremely skewed dataset. Figure 2 shows pie chart of class distribution where classes 0 and 1 represent fraud and non-fraud classes. Figure 3 shows the bar graph of classes 0 and 1.

3.2 Data Analysis

Clarity of the data being used is very important for working on any project. Graphical illustrations of the data give much clear information of the same. Here are the graphical illustrations of the data.

The above pie chart and bar graph show the distribution of fraudulent and legitimate transactions. Legitimate transactions are labeled as '0', while fraudulent transactions are labeled '1'.

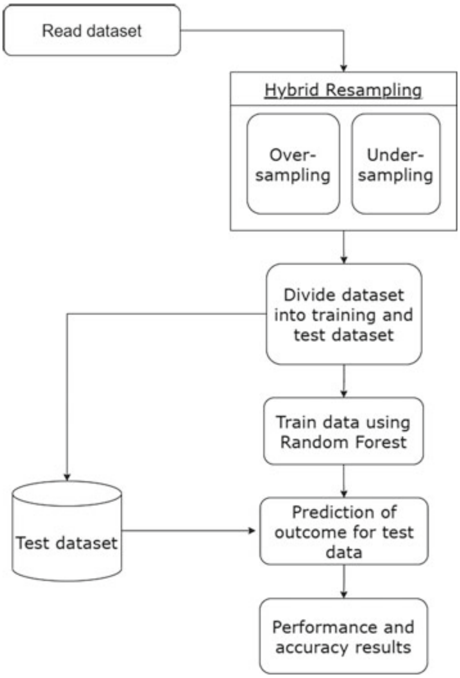

Read dataset
Hybrid Resampling
Over-sampling
Under-sampling
Divide dataset into training and test dataset
Train data using Random Forest
Test dataset
Prediction of outcome for test data
Performance and accuracy results

Fig. 1 System architecture

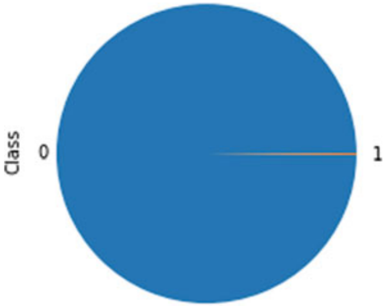


Fig. 2 Pie chart of class distribution

The percentage distribution obtained is as follows: non-fraud transaction is 99.83% of the dataset, and fraud transaction is 0.17% of the dataset.

3.3 Data Processing

PCA Transformation: The Principal Component Analysis reduces noise and decreases requirements for memory. It is a dimensional reduction method that reduces

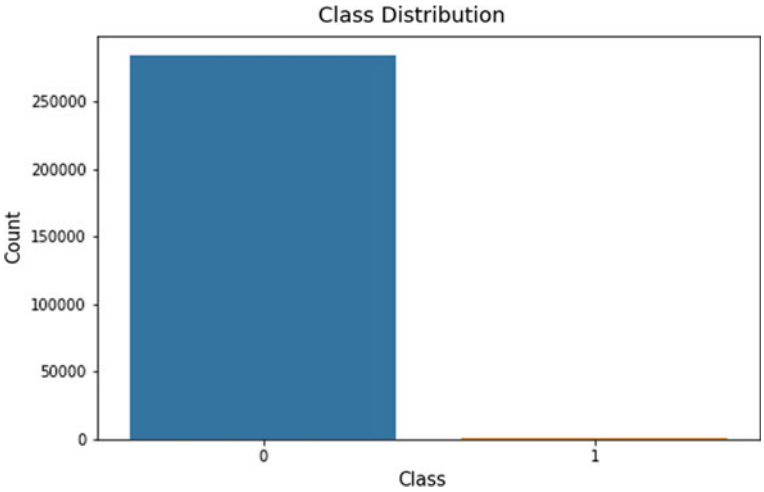


Fig. 3 Bar graph depicting class distribution

the dimensionality of a large dataset, by transforming a larger variable into a smaller one but still contains most of the information of the larger dataset.

Formatting: The dataset obtained is PCA formatted.
Cleaning: This step involves removing data or fixing the data. The data that is incomplete or missing is removed.
Sampling: The dataset obtained is highly imbalanced. So, to make the data balanced and to get a better accuracy, we need to perform hybrid sampling. It is a technique that combines two sampling techniques to create a balanced dataset, such as random oversampling and random undersampling.

3.4 Training the Model

The model is built using the Random Forest algorithm. The Random Forest algorithm is chosen as it has many advantages as compared to other classification algorithms. Using the scikit learn library, the dataset is split into training and testing data. During the training phase, 80% of the data is used to train the model, while 20% is utilized to test it later.

3.5 Model Evaluation

The trained model is tested using the test dataset. Using the metrics module on scikit learn, the results are evaluated using several evaluating factors such as accuracy (the fraction of correct predictions made by the model), recall (the number of correct positive results divided by the total number of relevant samples), precision (the number of correct positive results divided by the number of positive results predicted by the classifier), and so on.

4 Proposed Methodology and Algorithm

The project's primary purpose is to detect credit card fraud, for which the data set utilized was obtained from Kaggle. This dataset consists of features from V1, V2, …, V28. Due to concerns about confidentiality, these features have been PCA transformed. Amount, time, and class are the only features that are not PCA transformed. The "Time" feature displays the time difference in seconds between the dataset's first and current transactions. The feature "Amount" represents the transaction amount, while the "Class" feature indicates whether the particular transaction is legitimate or fraudulent. Legitimate transactions are indicated with a '0'. Fraudulent transactions are indicated with a '1'. The PCA transformation reduces noise and the data is cleaned. Clean data is data that does not contain any missing value or the data of a wrong data type (here data of other than numerical data type). The dataset obtained is highly imbalanced. This dataset has 492 frauds of 284,315 transactions. The ratio of legitimate transactions to that of fraudulent transactions is huge making the dataset highly imbalanced. This imbalanced data is very difficult to work upon as it contains noise and different dataset properties. To make the dataset balanced, hybrid resampling is used. Hybrid resampling consists of two types of sampling techniques: random undersampling and random oversampling.

4.1 Undersampling

Undersampling is a technique to make an imbalanced dataset into a balanced dataset. It is a method of reducing the number of data in the majority class while keeping the number of data in the minority class constant. Random Undersampler is being used for undersampling in this project. Random Undersampler is a class used to perform undersampling. The majority class which is the number of transactions is being reduced from 284,315 to 1578. But still, the number of transactions to that of fraudulent transactions is not balanced.

4.2 Oversampling

Another technique for turning an unbalanced dataset into a balanced dataset is oversampling. It is a technique in which the majority class is maintained while the minority

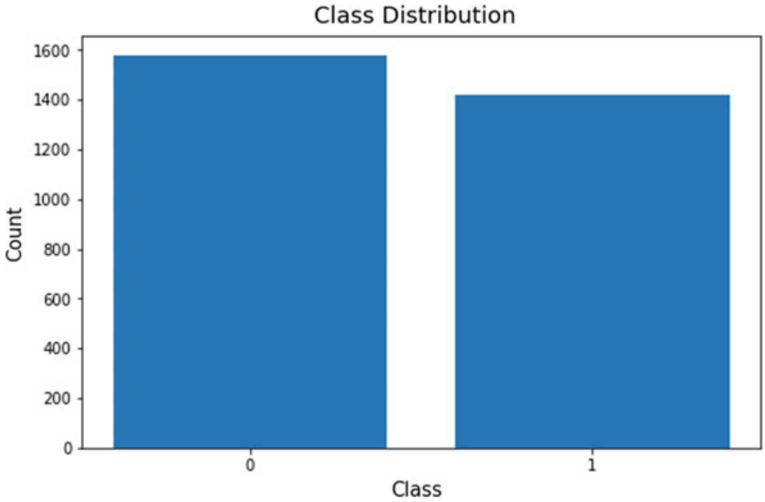


Fig. 4 Bar graph depicting the class distribution after hybrid resampling

class is supplied and grown with data. The most popular oversampling tool, Synthetic Minority Oversampling Technique (SMOTE), was utilized in this project. SMOTE begins by randomly selecting an instance of a minority class and then searches for the k closest neighbors of that minority class. Next, a line segment is created in the feature space by joining a and b, which is one of the k nearest neighbors, b, at random. This creates the synthetic instance. The two selected cases, a and b, are combined in a convex manner to construct the synthetic instances. After performing oversampling, the fraudulent transactions have been increased to 1421 which were earlier 492. After performing undersampling and oversampling, the majority class now has 1578 transactions, while the minority class has 1421 fraudulent transactions. Now the dataset is quite balanced and much easier to work upon. Performing both random oversampling and random undersampling, also known as hybrid resampling, makes the data well balanced. Here is a graphical illustration in Fig. 4 depicting the distribution of legitimate and fraudulent transactions after the data has been resampled.

The training and testing datasets are separated from the main dataset. The Random Forest algorithm is used to train the data.

4.3 Ensemble Learning

Individual models are combined in ensemble learning to improve the model's stability and predictive capacity. This technique increases the model's prediction ability. It integrates several machine learning models into a single prediction model that is a powerful classifier with a low error rate.

As stated earlier, machine learning algorithms play an important role in fraud detection as they work on both classification and regression problems. These algorithms are extremely quick to adapt to variations in normal behavior and can quickly spot fraud transaction patterns.

Random Forest is an ensemble learning system that relies on the bagging approach. It is one of the most popular techniques used for detecting fraud. This is because it works for both classifications as well as regression problems. It is a group of Decision Trees together which are termed 'forest'. Each Decision Tree makes a different prediction. The one with the maximum votes is considered. This method is known as the bagging technique, and it is used to create a forest of Decision Trees. The steps of working of random forest algorithm is—the algorithm chooses N records at random from the dataset and uses these records to build Decision Trees. According to the number of trees specified, above steps are repeated for different sets of N random records, each Decision Tree is used to predict which category the test record belongs to, and the record is assigned to the category with the most number of votes.

4.4 Performance Parameters

Evaluating the performance of an algorithm is very important. Accuracy is the most important parameter to judge a model. Other characteristics to consider while evaluating a model are precision, recall, and F1-score. The accuracy, precision, recall, and F1-score are used to evaluate the results of this research.

Accuracy is defined as the proportion of correct predictions to total input samples.

$$\text{Accuracy} = \frac{\text{Number of correct Predictions}}{\text{Total number of predictions made}}.$$

The accuracy obtained by our machine learning algorithm—Random Forest—is 0.9766666666666667.

Precision is dividing the true positives by the number of positive results.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}.$$

The precision obtained by our machine learning algorithm—Random Forest—is 0.9885931558935361.

When the number of true positives is divided by the total number of positives, the recall is determined.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}.$$

The recall obtained by our machine learning algorithm—Random Forest—is 0.959409594095941.

It is a consonant mean between precision and recall. The better the model performed, the higher the F1-score.

$$F1 = 2 \times \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}.$$

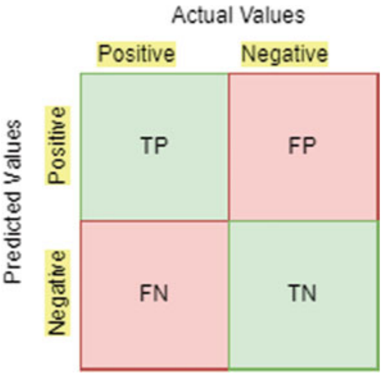
The F1-score obtained by our machine learning algorithm—Random Forest—is 0.9737827715355806.

5 Results and Discussion

The Confusion Matrix as shown in Fig 5 provides a comprehensive assessment of the model in terms of the matrix. True negative, true positive, false negative, and false positive are the four sections. True positive (TP)—actual true values. False positive (FP)—values that are not true but are mentioned as true. False negative (FN)—values that are not false but are mentioned as false. True negative (TN)—actual false values.

According to the Confusion Matrix above in Fig. 6, 326 transactions are true positive, which means that these are the transactions that are legitimate and have also been predicted as legitimate. Three transactions are false positive, which means that these are the transactions that are not legitimate yet have been predicted as legitimate. Eleven transactions are false negative, which means that these are the transactions that are fraudulent but have not been predicted so. Two hundred and sixty transactions are true negative, which means that these are the transactions that

Fig. 5 Confusion Matrix representation



are fraudulent and have been predicted as the same. Random Forest Errors: 13, accuracy score: 0.9783333333333334, classification report is projected in Table 1.

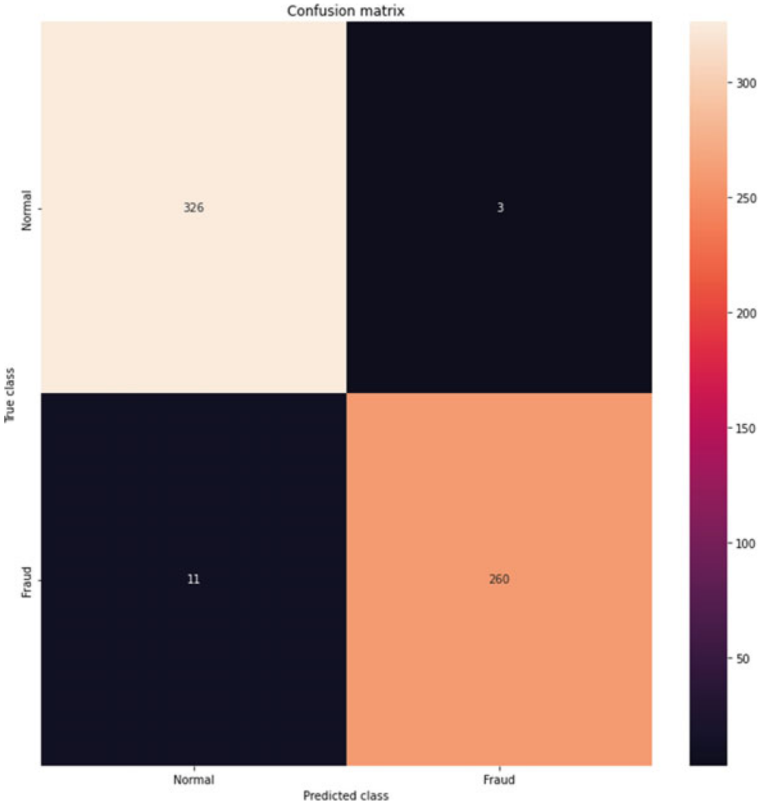


Fig. 6 Confusion matrix obtained

Table 1 Classification report

	Precision	Recall	F1-score	Support
0	0.97	0.99	0.98	329
1	0.99	0.96	0.98	271
Accuracy	0.97	0.98	0.98	600
Macro avg.	0.98	0.98	0.98	600
Weighted avg.	0.98	0.98	0.98	600

Table 2 Comparative analysis of various machine learning algorithms

Model	Accuracy	Precision
Logistic regression	93.84	96.58
Decision tree	92.88	99.48
Random forest	97.66	98.85
XGBoost	90.76	89.96
Naïve bayes	91.62	97.09
Support vector machine	94.99	95.98

6 Comparative Analysis

Various algorithms have been implemented for credit card fraud detection. The Support Vector Machine (SVM) algorithm does not work effectively with large datasets and requires more preprocessing. This makes the system inefficient as the speed of the model gets affected. Logistic regression might have over-fitting issues for large datasets and is particularly susceptible to data imbalances. Decision Trees make use of a single tree which can become biased based on the dominating classes. Random forest uses multiple Decision Trees which in turn gives higher accuracy. Though XGBoost has great precision and accuracy, its main drawback is that it is time-consuming which is not desirable for fraud detection as transactions need to be real-time. Table 2 shows comparative analysis of various machine learning algorithms for detecting fraud of credit card.

As Random Forest depends on multiple Decision Trees to produce an output, the output is unbiased making it more reliable. Random Forest can be used for a large dataset as it only works on a random subset of the dataset at a time. As Random Forest algorithm deals with multiple Decision Trees, the processing time may be more sometimes depending on the power of the system used.

7 Conclusions

Credit card fraud has always been a major concern. Speedy detection of fraud has become the need of the time. With the world contributing daily to become more and more digitalized even in the sector of banking, credit card fraud remains a problem. With the huge advancement in technology, fraud detection has also become more powerful in recent years. The use of machine learning algorithms has further strengthened the fraud detection system. The Random Forest algorithm performs very well with a large amount of data. The use of balanced data, obtained from hybrid resampling, helps in much more efficient working of the system as the processing time is lessened. With the headway in innovation, fraudsters can also become more advanced to commit crimes. The credit card fraud detection system needs to be constantly updated to be able to work efficiently. From the proposed system, we would like to conclude that fraudulent cases cannot completely be eradicated but can surely be lessened with early detection.

References

1. Maddie S (2021) Cash vs credit card spending statistics (2021)—Fundera ledger. Fundera: compare your best small business loan and credit card options, 30 July 2019.
2. George S, Pathade P, Ranjan N et al (2022) Implementation of machine learning algorithm to detect credit card frauds. Int J Comput Appl 184(1)
3. NCRB Report (2020) Debit, credit card fraud online climbs steeply, 225% more cases when compared to 2019. Edex Live
4. Midhun C, Nihar R (2021) Evolutionary and ıncremental text document classifier using deep learning. Int J Grid Distribut Comput 14(1)
5. Meenakshi D, Gayathri J, Indira M (2019) Credit card fraud detection using random forest. Int Res J Eng Technol (IRJET) 6(3)
6. Shakya R (2018) Application of machine learning techniques in credit card fraud detection. UNLV theses, dissertations, professional papers, and capstones, pp 3454
7. Ranjan N, Prasad R (2013) Author ıdentification in text mining for used in forensics. Int J Res Adv Technol 1(5):568–571
8. Lakshmi SVS, Selvani DK (2018) Machine learning for credit card fraud detection system. Int J Appl Eng Res 13:ISSN 0973–4562
9. Nayan U et al (2021) Literature review of different machine learning algorithms for credit card fraud detection. Int J Innov Technol Explor Eng 10(6):101–108
10. Vaishnavi Nath D, Geetha S (2019) Credit card fraud detection using machine learning algorithms. Procedia Comput Sci 165:631–641
11. Paruchuri H (2017) Credit card fraud detection using machine learning: a systematic literature review. ABC J Adv Res 6(2):113–120
12. Gowthami K, Praneetha KVLE, Vinitha G, Kumari R, Sandhya Krishna P (2020) Credit card fraud detection using logistic regression. J Eng Sci (JES) 11(4)

APiCroDD: Automated Pipeline for Crop Disease Detection

Pawan K. Ajmera, Sanchit M. Kabra, Anish Mall, Ankur Lhila, and Aaryan Agarwal

Abstract This research paper proposes APiCroDD: automated pipeline for crop disease detection, an automated framework for early detection of plant diseases using multispectral imagery from drones. Current frameworks for disease detection are labor and time-consuming. They do not leverage the richness of multispectral imagery for feature extraction and perform vanilla manipulation of agriculture indices. Our framework comprises two stages: data acquisition and disease identification. We find that the use of multispectral imagery in the proposed framework provides several advantages over traditional RGB imagery, including better spectral resolution and increased sensitivity to subtle changes in plant health. The multispectral data enables the identification of specific spectral bands associated with diseased regions of the plant, improving the accuracy of disease detection. The proposed framework utilizes a combination of CNNs and segmentation techniques to identify the plant and its disease. Experimental results demonstrate that the proposed framework using EfficientNet is highly effective in identifying a range of plant diseases achieving state-of-the-art performance on manually collected dataset and validated on the PlantVillage dataset.

Keywords Machine learning · Precision agriculture

1 Introduction

In recent years, precision agriculture has become increasingly popular due to its potential to improve crop yields while minimizing the use of pesticides and other chemicals [1]. Nevertheless, the utilization of pesticides and chemicals has negative impacts on human health and increases the costs of production [2]. In this study, we focus on the early detection of plant diseases affecting the leaves of these crops by employing deep learning models. The PlantVillage dataset is utilized for training and

P. K. Ajmera · S. M. Kabra (✉) · A. Mall · A. Lhila · A. Agarwal
Birla Institute of Technology and Science, Pilani, India
e-mail: kabrasanchit@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_35

testing, limiting the diseases to those included in the dataset [3]. We also manually collect a dataset and run extensive experiments to develop a model more tailored for specific regions [4].

In the pursuit of precision agriculture, sensor networks, remote sensing, and robotics find implementation in agricultural fields. However, these tools have limitations as they cannot detect plant diseases with the same level of expertise as humans. [5]. To overcome this limitation, the utilization of deep learning techniques, specifically convolutional neural networks (CNNs), has become increasingly popular in plant disease detection. In this study, we employ CNN models, namely GoogLeNet [6], EfficientNet [7], AlexNet [8], and also compare them with transformer-based model Vision Transformer [9], for the identification of diseased plant leaves. The CNN models were trained on a large dataset, with a focus on achieving high performance in plant classification and disease detection, as measured by the F1-score [10].

Moreover, the proposed framework utilizes multispectral imagery, which provides several advantages over traditional RGB imagery, including better spectral resolution and increased sensitivity to subtle changes in plant health [11]. This enables the identification of specific spectral bands associated with diseased regions of the plant, improving the accuracy of disease detection. Experimental results unequivocally demonstrate the efficacy of the proposed framework in identifying a spectrum of plant diseases [12].

By harnessing the power of deep learning and multispectral imagery, the proposed framework has the potential to revolutionize plant disease management. It could serve as a valuable tool in preventing crop loss by enabling early detection methods. Plus, its completely automatic nature would require minimal human intervention, making it highly attractive to farmers. As a result, this framework could usher in a new era in agriculture, allowing for reduced production costs and lowered use of harmful chemicals.

2 Related Works

In recent years, machine learning has seen significant advancements, particularly in the field of deep learning. Deep learning techniques have proven to be highly effective in categorization tasks, such as image classification, speech recognition, and natural language processing [13]. The objective of this research is to identify the most efficient model for classification tasks. Impressive results have been achieved by AlexNet, GoogleNet, and EfficientNet—all of which are popular deep learning models for image classification. GoogleNet and AlexNet had rather unfavorable results of 6.7% and 15.3%, respectively, while EfficientNet outperformed them exceptionally well, achieving an error rate of only 2.8% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). However, the larger size and higher computational requirements of GoogleNet and AlexNet make them unfavorable for mobile applications. Instead, EfficientNet and DenseNet—with their smaller sizes and lower computational requirements—present a better option for deployment in settings with resource restrictions or mobile applications. Recently, Vision Transformer (ViT), a new deep learning model based on the transformer architecture, has gained popularity due to its state-of-the-art performance on image classification benchmarks [9]. The self-attention mechanism employed in ViT aids in feature extraction from images. Overall, choosing the most appropriate model for classification tasks depends on factors such as model size, computational capabilities, and accuracy [14].

Transfer learning is another vital technique for deep learning-based classification tasks. It enables pre-trained models to be repurposed for new tasks, facilitating rapid deployment of models with comparatively limited data. One of the most recognized transfer learning approaches is fine-tuning, involving the retraining of a pre-trained model on a new dataset with a few additional layers [15]. Recent research has shown that fine-tuning pre-trained models is highly effective for image classification tasks. For instance, in a study by Khan et al., a pre-trained VGG16 model was fine-tuned on a dataset of tomato plant images to identify disease symptoms, resulting in an accuracy of over 97% [16]. Similarly, a pre-trained ResNet-50 model was fine-tuned to identify disease symptoms in apple plant images in a study by Mao et al., resulting in an accuracy of over 96% [17].

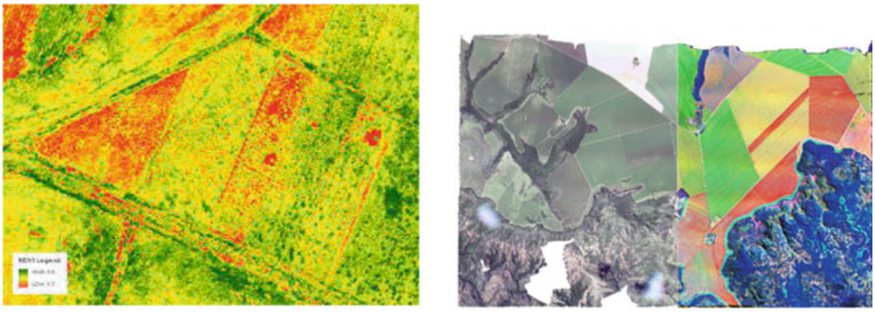
Data augmentation is another crucial component of deep learning-based classification tasks. Data augmentation involves generating new training data from existing data by applying various transformations, such as rotations, flips, and scaling. Data augmentation can assist in improving the generalization ability of deep learning models and mitigating overfitting. In a study by Jin et al., data augmentation was employed to improve the accuracy of a deep learning-based model for detecting tomato leaf diseases, resulting in an accuracy improvement of over 5% [18].

3 Methodology

Data Acquisition Equipped with a multispectral camera, our drone soared above the field, capturing images that held crucial information about the crops. Five bands—Red, Green, Blue, Near-Infrared, and Red-Edge—were utilized to record the spectral reflectance of the crops. We maintained a height of 50 m so as to produce images with a resolution of 5 cm/pixel. In doing so, we were able to discern unique growth patterns, health, and other defining characteristics of the crops in great detail.

3.1 Multispectral Analysis

Multispectral analysis plays a vital role in the proposed framework for early detection of plant diseases. To ensure accurate and reliable results, we followed a rigorous methodology for conducting the multispectral analysis. The acquired images underwent preprocessing to correct any radiometric and geometric distortions. The radio-



(a) NDVI image of an agriculture field

(b) Snapshot of collected multispectral dataset

Fig. 1 Multispectral dataset

metric corrections included calibration, normalization, and conversion to reflectance values. These corrections were necessary to ensure that the images accurately represented the spectral reflectance of the crops. The geometric corrections included orthorectification and georeferencing, which removed any distortions caused by the terrain and aligned the images with the real-world coordinates. These corrections were necessary to ensure that the images were properly aligned with each other and with the ground truth data. The preprocessed images were then used to extract features related to the spectral reflectance of the crops. Normalized Difference Vegetation Index (NDVI) is widely used for estimating vegetation cover and is a good indicator of plant growth and health. Green Normalized Difference Vegetation Index (GNDVI) is specifically sensitive to changes in the green vegetation, and Soil-Adjusted Vegetation Index (SAVI) is designed to reduce the soil background noise and improve the sensitivity of the index to vegetation cover. These indices are crucial for disease detection in the proposed framework as they allow us to identify abnormal vegetation patterns that may be indicative of disease or stress. Diseases can cause changes in the spectral reflectance of crops, which can be detected by analyzing the vegetation indices. For example, if a crop is infected with a disease, its chlorophyll content may decrease, resulting in a lower NDVI value (Fig. 1a). Thus, by computing and analyzing vegetation indices (Fig. 1b), we can identify areas of the field that may be affected by disease and can further investigate them for diagnosis. Several studies have shown the effectiveness of vegetation indices in disease detection. For instance, a study by Mahlein et al. [19] demonstrated that NDVI was effective in identifying early signs of disease in sugar beet crops. Another study by Yang et al. [20] showed that GNDVI was useful in detecting bacterial leaf blight in rice plants. These studies and many others highlight the importance of vegetation indices in disease detection and validate their use in the proposed framework.

In summary, the multispectral analysis in our proposed framework involves data acquisition using a drone equipped with a multispectral camera, image preprocessing to correct for radiometric and geometric distortions, and feature extraction to capture the spectral reflectance, health, and structure of the crops. This approach allows us to gather detailed information about the crops, which can be used to identify crop types and detect changes in crop health.

3.2 Plant Classification and Disease Identification

Deep learning is a way for learning through representation. Convolutional layers store the results of how filters or kernels interact with the layer below them. These filters, or kernels, are made up of weights and preferences that need to be taught. The goal of the optimization function is to make these cores that correctly represent the data. Pooling layers are used for downsampling to lessen the size of each cell and stop it from becoming too exact. Max pooling, which takes the highest number in the pooling area, is the most common type of pooling. Activation function levels are used to make the network not work in a straight line. ReLU is the most-used activation function. Dropout layers are used to prevent overfitting. Dropout layers turn off neurons in the network at random. To figure out the class odds or scores, we use layers that are all linked to each other. The output of the layers that are all linked to each other can be fed into the classifier. Softmax classifier is a well-known and widely used classifier.

AlexNet Krizhevsky et al. entered in the ImageNet image classification competition in 2012, presenting their AlexNet (Refer Fig. 2) network architecture, which achieved a remarkable success. The AlexNet architecture served as the initial foundation for the emerging trend of convolutional neural networks (CNNs). The AlexNet architecture incorporates rectified linear units (ReLU), local response normalization, and overlapping pooling techniques. The architecture of AlexNet is depicted in Fig. 2, situated on the left-hand side. The AlexNet architecture comprises five convolutional layers, each of which is subsequently followed by a rectified linear unit (ReLU) layer. The inclusion of normalization layers serves to facilitate the process of generalization, as stated in reference [9]. The characteristics present in the fifth convolutional layer are transmitted to a fully connected network subsequent to pooling. As previously stated, fully connected layers are responsible for computing the probability of

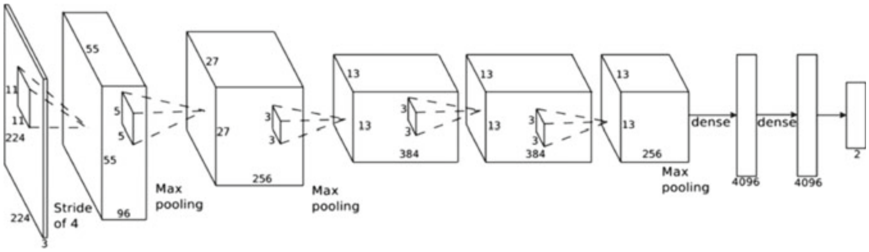


Fig. 2 Architecture of AlexNet

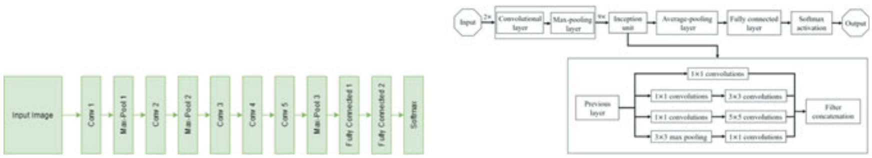


Fig. 3 Architecture of GoogleNet

a given class. Dropout layers are incorporated into fully connected layers to mitigate the issue of overfitting. The final fully connected layer, denoted as FC8, contains the predicted class probabilities for a given input image. The probabilities are categorized using a softmax classifier.

GoogleNet is a deep neural network architecture that was developed by Google researchers in 2014. It is designed for image classification and object recognition tasks. The GoogleNet architecture, as shown in Fig. 3, is based on a deep convolutional neural network with 22 layers, including multiple inception modules. One of the main advantages of the GoogleNet architecture is its efficiency in terms of both memory usage and computational cost. This is achieved through the use of 1×1 convolutions, which reduce the dimensionality of the input feature maps, as well as the incorporation of global average pooling, which replaces the traditional fully connected layers and significantly reduces the number of parameters. This auxiliary classifier consists of a small convolutional network followed by a fully connected layer and a softmax classifier, and its outputs are combined with the main classifier during training.

Vision Transformer Vision Transformer (ViT) Fig. 4 is a deep neural network architecture that has shown remarkable performance in image classification tasks. It is based on the transformer architecture introduced in natural language processing, which has been adapted to the vision domain. Unlike convolutional neural networks (CNNs), ViT processes the input image as a sequence of patches and uses a self-attention mechanism to capture the long-range dependencies among them. The ViT architecture consists of a sequence of transformer blocks, each of which has a multi-head self-attention mechanism followed by a position-wise feedforward network. The self-attention mechanism allows the network to attend to different parts of the image, while the feedforward network applies nonlinear transformations to the features. The output of the transformer blocks is passed through a classification head, which maps the features to class probabilities. During pre-training, the model is trained to predict the missing patch in an image, given the rest of the patches. This pre-training step helps the model to learn useful features that can be transferred to downstream tasks, such as image classification. However, ViT requires more computational resources during training and inference due to the use of self-attention mechanisms. Therefore, it is typically trained on large-scale distributed systems, such as GPUs or TPUs, to speed up the training process.

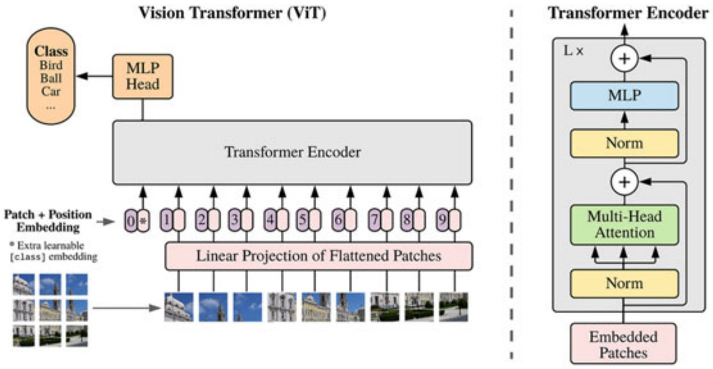


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Fig. 4 Vision transformer architecture

EfficientNet EfficientNet (Fig. 5) performs image classification tasks while being economical with memory and computing resources. It accomplishes this by compound scaling the network's width, depth, and resolution in a way that balances the trade-off between accuracy and efficiency. In particular, this architecture combines inverted bottleneck blocks, which boost the network's representational strength while reducing the number of parameters, with depth-wise separable convolutions, which lower the number of parameters and boost computing efficiency. The foundation of EfficientNet is a backbone network made up of numerous convolutional layers with different widths, depths, and resolutions. The network starts off by extracting low-level features from the input image using a stem convolutional layer, which is followed by a series of repeated blocks that gradually improve the features. Each level in which the blocks are arranged has a variable width, depth, and resolution. Finally, a head network is made up of a few fully connected layers and a softmax classifier receives the output of the backbone network and outputs the anticipated class probabilities. One of the main advantages is its capacity to perform on image classification problems with significantly smaller networks than previous state-of-the-art approaches. EfficientNet is thus well suited for applications with limited resources, such as those found in embedded systems and mobile devices. It has also been demonstrated to generalize effectively to other computer vision applications, like object detection and segmentation, with only small adjustments.

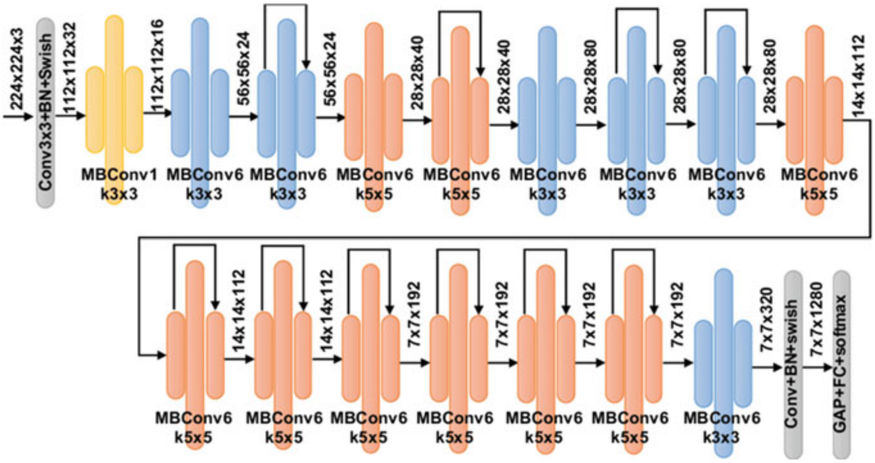


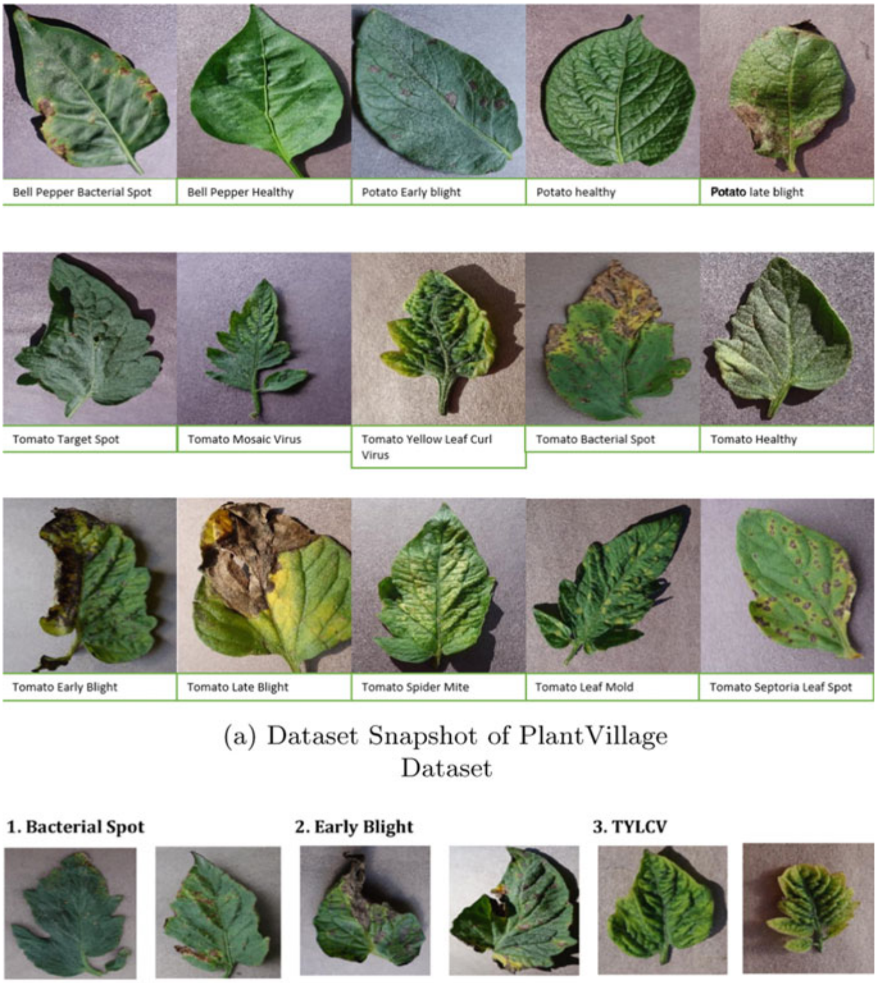
Fig. 5 EfficientNet architecture

3.3 Dataset

To conduct our research, we utilized the publicly available PlantVillage dataset (Fig. 6a), which consists of images of healthy and unhealthy plant leaves. The dataset contains 54,306 images of 14 crop species and has been annotated by plant pathology experts for the identification of diseases and conditions. We took all the images of potatoes, tomatoes, and bell peppers, both healthy and diseased, for training and testing. Along with this dataset, we curated a custom manually collected dataset(Figure 6b) to capture more regional context. The dataset consists of images collected from various agricultural fields, encompassing five different crop types: tomato, bajra (pearl millet), potato, rice, and bell peppers. The dataset was specifically curated to support our investigation into plant disease detection and classification using deep learning techniques.

The images were captured using high-resolution cameras mounted on unmanned aerial vehicles (UAVs) or ground-based imaging systems. Multiple images were taken from different angles and viewpoints to capture various aspects of the crops, including leaves, stems, and fruits. The dataset includes images captured at different growth stages and under varying lighting and environmental conditions to ensure the representation of real-world scenarios.

To ensure the accuracy and reliability of the dataset, rigorous data collection protocols were followed. The crop samples were carefully selected to represent healthy plants as well as plants exhibiting different disease symptoms. In collaboration with agricultural experts, we conducted thorough visual inspections of the crops to identify and annotate instances of diseases and pests. Each image in the dataset is labeled with the corresponding crop type and disease condition, allowing for supervised training and evaluation of deep learning models.



(a) Dataset Snapshot of PlantVillage Dataset

(b) Snapshot of manually collected dataset

Fig. 6 Disease datasets snapshot

4 Results

Each model was trained until the change in training loss became negligible, resulting in models that reached a stable state. The training loss for each model was monitored, and the point at which it exhibited minimal change was identified. The following subsections provide detailed information about the hyperparameters and performance of each model.

4.1 AlexNet

The AlexNet model was trained for 50 epochs based on observations from the training loss graph. The loss decreased steadily during the initial epochs and continued to decrease at a slower rate until convergence. A batch size of 64 was used, along with a learning rate of 1e−5 and a momentum of 0.5. The final accuracy achieved on the testing set was 99.42%, and the $F1$-score was 0.982.

4.2 GoogleNet

The GoogleNet model was trained for 30 epochs based on observations from the training loss graph. During the initial epochs, the loss decreased significantly, but after 5 epochs, it reached a plateau. A batch size of 16 was used, along with a learning rate of 2e-3 and a momentum of 0.9. The final and best accuracy achieved on the testing set was 99.56%.

4.3 EfficientNet

The EfficientNet model was trained for 25 epochs. Similar to GoogleNet, the loss curve showed a significant decrease in the initial epochs, but after the last 10 epochs, the loss stagnated. A batch size of 32, learning rate of 3e −3, and momentum of 0.75 were used for training. The final and best accuracy achieved on the testing set was 99.73%.

4.4 Vision Transformer

The Vision Transformer model was trained for 30 epochs. The training loss stabilized after the 25th epoch. The model was trained using a batch size of 32, learning rate of 2e−5, and momentum of 0.9. The final and best accuracy achieved on the testing set was 95.51%.

These results indicate that the GoogleNet and EfficientNet models achieved excellent accuracy on the testing set, surpassing 99%, while the Vision Transformer model achieved a slightly lower accuracy of 95.51%. Further analysis revealed that all models performed particularly well on certain classes as tomato and potato to name a few, while showing variations in performance on others.

It is important to note that the training and evaluation of these models involved data preprocessing techniques such as normalization and augmentation. We hypothesize the reason for Vision Transformers drop in performance is due to lack of training

Table 1 Comparison of test accuracy and F1-score of different models

Model name	Parameters	Test accuracy	F1-score
GoogleNet	7 Million	99.56%	0.977
EfficientNet	**5 Million**	**99.73%**	**0.985**
AlexNet	67 Million	96.73	0.982
Vision transformer	85 Million	95.51%	0.839

data. Transformers require vast amounts of data to live up to the state-of-the-art results promised by them. Since our data was rich in covering different classes, it points in the direction that more data gathering of each class would lead to increase in the performance of the transformer models.

Overall, the experiments demonstrated the effectiveness of the EfficientNet model, highlighting their potential for accurate image classification tasks. It has the least computational size and expense and hence would be ideal for deployment on any device with bare minimum computational features. We recommend to use EfficientNet in our final stages of the pipeline for the same reason. Refer Table 1 for details.

5 Conclusion

In this research paper, we have presented APiCroDD, an automated pipeline for early detection of plant diseases using multispectral imagery from drones. The proposed framework has demonstrated its effectiveness in improving the accuracy of disease detection compared to traditional RGB imagery. By leveraging the advantages of multispectral data, including superior spectral resolution and increased sensitivity to subtle changes in plant health, APiCroDD has shown promising results in identifying specific spectral bands associated with diseased regions of the plant. The combination of convolutional neural networks (CNNs) and segmentation techniques utilized in APiCroDD has proved to be a robust approach for identifying both the plant and its associated disease. Through training on a manually collected dataset and validation on the widely used PlantVillage dataset, the framework with EfficientNet has achieved state-of-the-art performance in detecting a wide range of plant diseases. In conclusion, APiCroDD offers a promising automated pipeline that leverages multispectral imagery and advanced deep learning techniques. With its potential to revolutionize disease management practices, APiCroDD paves the way for more efficient and proactive approaches to ensure crop health, increase agricultural productivity, and contribute to global food security.

Acknowledgements This work was supported by the Short Term Project Scheme of Technology Development Programme (DRISHTI CPS/TDP-ST/SL/2022/003) from IITI DRISHTI CPS Foundation.

References

1. Schramski JR, Dell AI, Grady JM, Eason TN, Kull CA (2015) Precision agriculture: challenges and opportunities for land use and food security. Science 349(6250):514–518
2. Chander S, Kumar P (2016) Precision agriculture in India: an overview. J Saudi Soc Agric Sci 15(1):1–9
3. Hughes DPJ, Salathe M (2015) An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060
4. Khamparia A, Singh G, Patel DK, Singh D, Singh V (2021) Deep learning based leaf disease detection model for tomato crop and its performance evaluation for diverse geographical locations. Sensors 21(1):215
5. Wang N, Nuyttens D, Luo Y, Cao Z (2016) An overview of precision agriculture technologies. J Integr Agric 15(11):2565–2584
6. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
7. Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946
8. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
9. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
10. Saito K, Jha S, Singh S, Singh P, Singh P (2015) Precision agriculture using machine learning and advanced data analytics. In: 2015 IEEE International conference on engineering and technology (ICET). IEEE, pp 1–4
11. Zhao C, Zhang D, Li Z, Wang S (2019) Multispectral imaging for plant phenotyping: a review. Comput Electron Agric 165:104961
12. Fuentes A, Yoon S, Kim SC, Park DK, Oh SI (2021) Deep learning for plant disease detection: a comprehensive review. Comput Electron Agric 183:106019
13. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
14. Mahajan D, Girshick R, Ramanathan V, He K, Paluri M, Li Y, Bharambe A, van der Maaten L (2018) Exploring the limits of weakly supervised pretraining. arXiv preprint arXiv:1805.00932
15. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in neural information processing systems, pp 3320–3328
16. Khan A, Sohail A, Ahmad S, Khan A (2018) Deep learning based tomato plant diseases classification. In: 2018 International conference on frontiers of information technology (FIT). IEEE, pp 68–73
17. Mao Q, Liu W, Liu S, Hu J, Chen H (2021) Efficient identification of apple diseases using a fine-tuned ResNet-50 model. Comput Electron Agric 184:106042
18. Jin J, Yu C, Zeng Y, Liu S (2021) Tomato leaf diseases detection using improved convolutional neural network. Plant Methods 17(1):13
19. Mahlein AK, Kuska MT, Thomas S, Wahabzada M, Behmann J, Rascher U (2012) Development of spectral indices for detecting and identifying plant diseases. Remote Sens Environ 126:132–142
20. Yang G, Liu C, Li Z, Zhao C, Li X, Wang Z (2017) Detection of bacterial leaf blight in rice using a UAV-mounted multispectral imaging system. Remote Sens 9(1):64

TimeGAN for Data-Driven AI in High-Dimensional Industrial Data

Felix Neubürger, Yasser Saeid, and Thomas Kopinski

Abstract The availability of historical process data in predictive maintenance is often insufficient to train complex machine learning models. To address this issue, techniques for data augmentation and synthesis have been developed, including the use of Generative Adversarial Networks (GANs). In this paper, the authors apply the GAN-based approach to synthesize simulated time-series data. Experiments are carried out to find a trade-off between the amount of labeled data needed and the accuracy of the synthetic data for downstream tasks. The authors find that using 40% of the original data for training the GAN results in synthetic data containing the same information for downstream tasks as the original data, leading to an estimated speedup of 60% in the initial computing time. The results of the evaluation for the authors' own FEM simulation data, as well as for the Tennessee-Eastman benchmark dataset, are presented, demonstrating the potential of GANs in reducing time and energy in process development, while additionally interpolating a fixed parameter grid that is subsequently used for simulation purposes. This work demonstrates the feasibility of using GANs for the generation of high-dimensional time-series data in industrial applications as a supplementary method to classical FEM simulations.

Keywords Generative Adversarial Networks · Industrial AI · Time-series generation · FEM simulation · Data-driven ai · Low data

F. Neubürger (✉) · Y. Saeid · T. Kopinski
South Westphalia University of Applied Sciences, 59872 Meschede, Germany
e-mail: neubuerger.felix@fh-swf.de

Y. Saeid
e-mail: saeid.yasser@fh-swf.de

T. Kopinski
e-mail: kopinski.thomas@fh-swf.de

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_36

1 Introduction

In the field of predictive maintenance, the availability and quality of historical process data are often insufficient to train complex machine learning models. Because of this recurring problem with deep learning, techniques are being developed focusing on the available data. Data augmentation and synthesis methods are successfully used to improve image recognition problems [13]. However, augmentation of data describing physical reality in a complex industrial process is not as easy as it is for images, since the physical reality must be preserved. One possible solution is to apply deep learning-based methods to synthesize simulated time-series data to subsequently compare different partitions of the training data with the goal of finding the amount of labeled data needed to generate accurate synthetic data for later use. In manufacturing, the modeling of new processes is often realized via dedicated finite element method (FEM) software, often requiting lengthy computation for the entire process. The use of certified numerical simulation modeling software is a very important step in the development of new complex industrial processes since accurate mathematical and physical conditions must be taken into account. When simulating a complex manufacturing process, a so-called process window is often defined within which the parameters for the simulation are set and then the simulation is performed. The high-dimensional parameter grids would however take a lot of computing resources with specialized software to yield sufficiently large datasets for machine learning model training in downstream tasks. In addition to the number of resources used for those calculations, there are cases where only very specific parts of the simulated data are needed for the specific downstream task. The computation time and energy resources may not be fully maximized, potentially resulting in the final simulated data not being employed. The computation time and required resources can be reduced by utilizing Generative Adversarial Networks (GAN) to synthesize the required parts of the simulated data within this parameter grid. The GAN can then be used as an additional tool to quickly generate important data for a specific use case analysis. Generating sequences with the help of generative methods does not replace thorough use of numerical simulation software. However, it would be a powerful complementary tool to a simulation and process development toolbox. After a GAN has been trained on a limited set of numerically modeled data, synthesizing new data points or sequences in the context of time-series data is not a very resource-intensive process, as the trained generator model only needs to perform logical output tasks. This inference step typically takes $O(seconds)$ time compared to $O(hours)$ for numerical modeling, hence improving the overall process by several orders of magnitude. This time saving is also a cost and energy saving in process development and may enable or enhance the use of deep learning in subsequent tasks. Additionally, a generative neural network can indirectly interpolate a parameter grid that is used for an FEM simulation by producing variations in the output data that translate into variations in the input parameter grid, leading to increased robustness of newly developed algorithms. This paper is structured as follows: Sect. 2 describes similar work that has been done with generative neural networks. In Sect. 3, we explain the data, preprocessing steps and models being used for the experiments. Section 4 summarizes the findings of the experiments while discussing the trade-offs that can be done to maximize synthesizing power while reducing the time and energy consumed in the process. Section 5 gives an overview of the findings and an outlook for the applicability of GANs within the domain of low-data problems.

2 Related Work

The TimeGAN architecture introduced by Yoon et al. [18] is a generative time-series model trained antagonistically and collectively through a learned embedding space with monitored and unsupervised losses. This approach overlaps several research directions and combines topics such as autoregressive sequence prediction models, GAN-based methods for sequence generation, and time-series representation learning [1, 2, 10]. The transformer architecture, based on multiple levels of self-awareness [5], has recently become a widely used deep learning model architecture. [11] has been shown to outperform many other popular neural network architectures, such as convolutional neural networks (CNN) on images and recurrent neural networks (RNN) on sequential data, and even to represent properties of a universal computing machine [6, 8, 11]. Certain studies attempt to incorporate the transformative model into the GAN model architecture in order to enhance the quality of synthetic data or optimize the training process [7, 9, 11] for tasks involving image and text generation. Evaluation of generative models can be done by performing a principal component analysis (PCA) [17] and t-distributed stochastic neighbor embedding (t-SNE) [12] which are used to map the multidimensional output sequence vectors into two dimensions. This is done to visually observe and qualitatively evaluate the similarity in the distribution of the synthetic data and real data instances. For a more quantitative comparison, several known signal properties are measured and compared to the transformer-generated ones as well as RNN-generated sequences with real sequences of the same class. Li et al. [11] proposed several heuristics to more effectively train a transformer-based GAN model on time-series data and quantitatively compared the quality of the generated sequences with real and with sequences generated by other state-of-the-art time-series GAN algorithms. The architecture of GAN is similar to that of DCGAN for generating auxiliary samples. Basically, it takes significant effort to prepare a dataset for machine learning pipelines with high-quality data points. Nonetheless, one can expect to find invalid states, including those that are misclassified, ambiguous, or entirely unrelated. If the number of such samples is small compared to the core items in the dataset, their presence has little impact on behavior. There are two reasons for this: Firstly, the gradients are averaged with respect to model parameters in each micro-batch. This reduces the effect of calculating an undesirable set of gradients for an invalid entry making the training more robust against such outliers. Secondly, the effect on average gradients is more limited and is determined by the learning rate. Thus, if the vast majority of the data is correct and related to the prediction target, the deleterious effect of invalid samples becomes insignificant. However, when the dataset is small, the percentage of invalid data points tends to be higher and the effect of these outliers on the model training increases. Because of this relationship, the generative approach to increasing the dataset size can lessen this effect. In such situations, taking additional steps to improve the quality of the dataset has a significant impact on accuracy.

3 Data and Methodology

This section describes the analyzed data and methods used for this study. The Tennessee-Eastman dataset is briefly introduced. We describe the self-generated data in the context of our press-hardening use case. Additionally, we give a short explanation of the TimeGAN architecture and the evaluation methods. The data used for the evaluation of the TimeGAN data synthesis experiments consists of two distinct datasets. Firstly the well-studied Tennessee-Eastman (TE) dataset serves as a benchmark for high-dimensional time-series classification tasks. The Tennessee-Eastman dataset is introduced by [15] and consists of "fault-free" and "faulty" datasets. Each data frame contains 55 columns ("fault number," "simulation-Run," "sample," and the other columns contain the process variables). These 52 usable variables are numerical with a description of the meaning of each variable available under [4]. The complexity of this dataset is comparable to the complexity of industrial processes. In the case of complex control systems, this work can be transferred to a specific case. This dataset is used to evaluate the methodology on a large and complex dataset to demonstrate its applicability to other industrial datasets. The second dataset under investigation was generated by the authors through self-generated finite element method (FEM) simulations of the pressing process. This process is commonly employed in the automotive industry to manufacture high-rigidity steel parts for car bodies. The complete simulation dataset uses associated FEM models calculating the heat transfer from a hot metal board to a die for molding, as well as numerically calculating the mechanical shifts of the metal. In this article, we will focus on the temperature data of this process, since this is the critical characteristic which needs to be monitored and/or correctly evaluated but cannot be measured directly. The temperatures gathered from the FEM simulation are representative of the real-world temperature sensors to ensure comparability between real and simulated data. Figure 1 shows the location of the thermocouples in the molding tool. From the point of view of a mechanical engineer, there is a process window, within which parameters outside this window can never give satisfactory results. But within this multidimensional window of the process, there are many possible sets of parameters that can produce feasible output results. We modeled a parameter grid with four parameters, resulting in a total number of simulation attempts using the ABAQUS [16] simulation software totaling $20,412$ simulation runs. The total simulation time of all sequences was about 30 weeks of single core processor time. Although we were able to parallelize this process via several processors, not all simulation runs converged due to the numerical instability of the process. These failed simulation runs are not included in the training

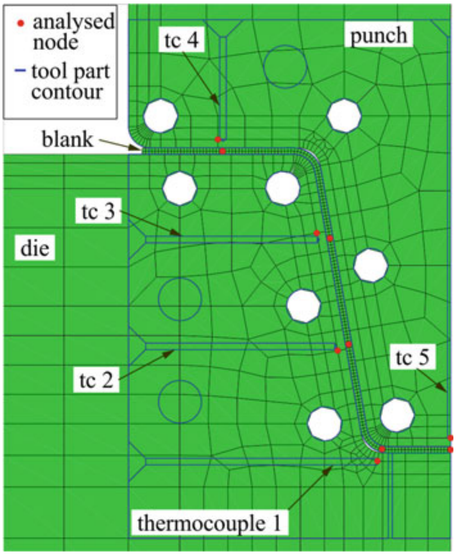


Fig. 1 Positions of the thermocouples (red dots) and nodes of the FE-model. The blank and contact areas are modeled more finely than the outer parts of the punch and die to save computation time. The white circular areas are the cooling pipelines being used for transferring heat from the tool out of the system in the real-world process

Table 1 Parameter grid used for the FEM simulation

Parameter	Minimum value	Maximum value	Number of steps
Oven temperature	800 °C	1000 °C	21
Forming die temperature	60 °C	170 °C	12
Pressing force	10 MPa	25 MPa	16
Holding time	4 s	10 s	4

set. The numerical instabilities leading to those failures are however not present in the suggested GAN approach, making the method a reliable substitution method for these cases. A more detailed description of this modeling process and the subsequent follow-up task can be found in [14].

The parameter grid we used to generate the training data can be found in Table 1. Based on this dataset, we posed the question as to how much FEM modeling data would actually be required to train a GAN model capable of interpolating between already modeled data grid points. Such a data synthesis would provide more variation in the training data for subsequent tasks. To generate synthetic sequences, we pre-process the training data resulting in unique fixed-length sequences which are subsequently fed to the GAN. Standard scaling is applied to the training data leading to better processing of the feature values by the neural network. After preprocessing,

Table 2 Hyperparameters used for the experiments

Parameter	Tennessee-Eastman	FEM-Data
Embedder epochs	500	5000
Supervisor epochs	500	5000
GAN epochs	500	500
Max sequence length	25	367
Batch size	10,000	10,000
Hidden dim	20	20
Num layers	3	3
Optimizer	adam	adam
Learning rate	0.001	0.001

we use the TimeGAN architecture described in [18] for data synthesis. The code used for this work is an adaptation of the TimeGAN [3] PyTorch implementation, and it can be found at https://github.com/DataScienceLabFHSWF/WiTraPresGAN. The parameters for training the timeline for different datasets are presented in the Table 2. By trying to minimize the number of parameters in the network itself to prevent overfitting and demonstrate the strength of the architecture itself. Hidden units are long-term short-term memory (LSTM) cells. When evaluating the effectiveness of TimeGAN, we are guided by the evaluation metrics given in [18]. In order to assess the similarity between the original and synthetic data, we employ the “Train on Real, Test on Synthetic" (TSTR) method. This technique, initially introduced in the original TimeGAN paper by Yoon et al., is commonly used to evaluate the performance of generative models specifically designed for time-series data. The basic idea behind the TSTR method is to train a generative model on real data and then use it to generate synthetic data that is similar to the real data. The synthetic data can then be used to evaluate the quality of the generative model. The TSTR method consists of two phases: a training phase and a testing phase. During the training phase, the generative model is trained on real data. Specifically, in the case of TimeGAN, a generator and a discriminator are trained using an adversarial training process. The generator is responsible for generating synthetic time-series data that is similar to the real data, while the discriminator is responsible for distinguishing between the real and synthetic data. Once the training phase is complete, the generative model is used to generate synthetic data that is similar to the real data. This synthetic data is then used in training a secondary RNN doing a regression for all the variables contained in the dataset. Specifically, the secondary model trained on synthetic data is compared to the one trained on real data using the mean squared error (MSE) metric. If the model that is trained on synthetic data behaves similar on real testing data when compared to the model trained on real data, then the generative model is considered to be successful. We measure this similarity by calculating the difference between the test RMSEs of both models. The TSTR method has several advantages over other methods for evaluating generative models for time-series data. One advantage is that it allows for a direct comparison between the real and synthetic data, which can be useful for determining how well the generative model is able to capture the underlying patterns and structures in the real data. Another advantage is that it can be used to evaluate the performance of generative models in a variety of applications, including anomaly detection, forecasting, and simulation. In addition to this metric, we visualize the data that is generated together with the real data. For that we apply, analogously to the original TimeGAN paper, t-SNE [12] and PCA [17] to the time-flattened sequences and plot the distributions into a single image. When the distribution of real and synthetic data is similar and closely clustered together, we can assume that a good distinction between these two classes is not possible for simple discriminative models. This means that the generated data can be used for training downstream task models without introducing an unwanted bias. As the generative model, we use the TimeGAN proposed by Yoon et al. [18] The TimeGAN architecture consists of two main components: a generator and a discriminator. The generator is responsible for creating synthetic time-series data that resembles the real data. The discriminator is trained to distinguish between the real and synthetic data. The generator consists of four components: an embedding network, a generator network, a masking network, and a recovery network. The embedding network maps the original time-series data into a latent space, where the generator network generates synthetic data. The masking network is used to randomly mask some of the time steps in the generated data to improve its realism, and the recovery network is used to fill in the masked time steps. The discriminator is a recurrent neural network that processes time-series data and classifies it as either real or synthetic. The generator and discriminator are trained in an adversarial way, where the generator generates synthetic data, while the discriminator tries to correctly classify the data as real or synthetic.

4 Experimental Results

Running the experiments, we find that not all of the simulation data is needed to train a GAN model synthesizing data that closely resembles the original data. This means that a significant speedup from the order of CPU-weeks to the order of a single day in model training and minutes in new sample generation. We tested fractions from 10% to 100% of the original FEM simulation data for training our TimeGAN. It is important to note that this data is randomly sampled from the full training dataset. After training as many synthetic sequences as original sequences are generated by the GAN to evaluate the similarity between original and synthetic data. We evaluate the trained models via the Train-Synthetic-Test-Real method to quantify the difference in information contained in the synthetic data compared to real data. We trained the LSTM to predict all of the features in the data with the other features as given parameters to simulate a regression downstream task. We then averaged the RMSE of the test predictions on the real data to aggregate the performance over all predictable features. We aim for the difference of the scores trained on original and synthetic

Table 3 Results of the Train-Synthetic-Test-Real evaluation of the FEM simulation data

Amount of training data (%)	RMSE original data	RMSE synthetic data	RMSE difference
10	0.33 ± 0.2	0.29 ± 0.1	0.04 ± 0.3
20	0.51 ± 0.2	0.57 ± 0.3	0.06 ± 0.5
30	0.38 ± 0.1	0.35 ± 0.1	0.03 ± 0.2
40	**0.37 ± 0.1**	**0.33 ± 0.1**	**0.04 ± 0.2**
50	0.41 ± 0.1	0.32 ± 0.1	0.08 ± 0.2
60	0.37 ± 0.1	0.32 ± 0.1	0.05 ± 0.2
70	0.37 ± 0.1	0.33 ± 0.1	0.04 ± 0.2
80	0.37 ± 0.1	0.32 ± 0.1	0.04 ± 0.2
90	0.22 ± 0.1	0.46 ± 0.2	0.24 ± 0.3
100	0.23 ± 0.1	0.47 ± 0.2	0.24 ± 0.3

RMSE of the helper LSTM trained on the given data for the feature regression task. A lower difference hints at a closer match between synthetic and original data

data to be small while the score itself should also be small on the LSTM trained on synthetic data. The results of the evaluation for our own FEM simulation data can be seen in Table 3. We find that the difference in predictive scores stagnates when using 40% of the original data as training data. The RMSE score of the LSTM also stagnates at that percentage of used training data. Second the performance of the LSTM drops significantly on the original data test set with a higher amount of training data used for the GAN training. This can be interpreted as an artifact of overtraining of the downstream model on the synthetic data. We conclude that a percentage of 40% of the original data could be used for training a GAN yielding synthetic data containing the same information for downstream tasks as the original data. This in turn results in an estimated speedup in initial computing time of 60%. Another, more qualitative method of showing the similarities between the original and synthetic data is transforming the sequences into a latent space with the t-SNE method and comparing the structures arising from that latent space representation in a two-dimensional space. A lack of cluster-like structures that can be used to discriminate the original and synthetic data shows that the information in the two classes is very similar and a clear distinction between them is not possible. This visualization can be seen in Fig. 2.

To benchmark our results against a well-known dataset, we run the same test procedure on the Tennessee-Eastman dataset. The results for the Train-Synthetic-Test-Real evaluation can be found in Table 4. We find that the performance of the GAN does not significantly depend on the percentage of used training data. This might be due to the initial size of the dataset and the information contained in the fractions used in the GAN training. When evaluating the generated sequences qualitatively with the help of the t-SNE embedding of the sequences, as shown in Fig. 3, one can

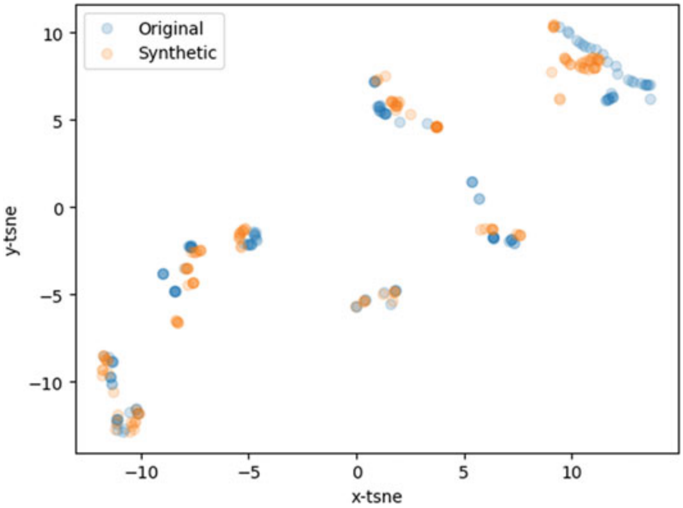


Fig. 2 t-SNE-plot for the visualization of the similarity of the original and synthetic data. This plot uses the synthetic data from the TimeGAN model that has been trained on 100% of the original FEM simulation data. It can be seen that the representations of the original and synthetic data are closely clustered together while still maintaining variation

Table 4 Results of the Train-Synthetic-Test-Real evaluation of the Tennessee-Eastman dataset

Amount of training data%	RMSE original data	RMSE synthetic data	RMSE difference
10	0.03 ± 0.03	0.09 ± 0.08	0.06 ± 0.11
20	0.04 ± 0.05	0.09 ± 0.07	0.05 ± 0.12
30	0.03 ± 0.04	0.17 ± 0.17	0.14 ± 0.21
40	**0.04 ± 0.06**	**0.09 ± 0.10**	**0.05 ± 0.16**
50	0.04 ± 0.05	0.09 ± 0.09	0.05 ± 0.14
60	0.03 ± 0.04	0.08 ± 0.09	0.06 ± 0.17
70	0.04 ± 0.05	0.08 ± 0.08	0.04 ± 0.13
80	0.04 ± 0.06	0.07 ± 0.09	0.03 ± 0.15
90	0.03 ± 0.06	0.08 ± 0.07	0.04 ± 0.10
100	0.03 ± 0.05	0.07 ± 0.08	0.04 ± 0.13

RMSE of the helper LSTM trained on the given data for the feature regression task. A lower difference hints at a closer match between synthetic and original data

see a close match between the original and synthetic data with some variations. This underlines the time-series generation capabilities of the TimeGAN trained in this work even for complex datasets. In addition to the faster sequence generation and resulting time and energy savings, the method has an added benefit. When simulating a rigid parameter grid, one can only generate a sparse representation of the whole process. The use of a generative model allowing for variation in the output data also

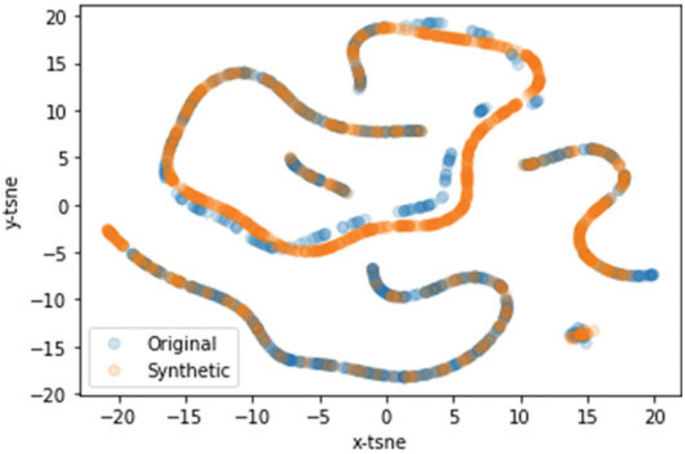


Fig. 3 t-SNE-plot for the visualization of the similarity of the original and synthetic data. This plot uses the synthetic data from the TimeGAN model that has been trained on 60% of the original Tennessee-Eastman dataset. It can be seen that the original and synthetic data representations are closely clustered together and follow the same shape but still have some variations between them

gives a possibility for an indirect interpolation of the parameter space. A schematic illustration of this grid interpolation is shown in Fig. 4. This interpolation makes the parameter space less sparse and might lead to improvements in performance of downstream analyses, because of a more completely available process space.

5 Discussion and Outlook

The results of this work show that Generative Adversarial Networks (GANs) have the potential to significantly improve the use of deep learning models in predictive maintenance. In particular, we demonstrate that GANs can be used to synthesize simulated industrial time-series data, which can then be used as training data for downstream tasks. The experiments carried out in this study show that the use of GANs can reduce the amount of labeled data needed for these tasks by around 60 %, as well as reduce the computing time and resources required for this process. Using GANs for synthetic data generation can lead to a significant speedup in the initial simulation computing time and due to that a reduction in energy consumption. These are important considerations in the industrial sector. The trade-off between the amount of labeled data required and the usefulness of the synthetic data for downstream tasks is an important aspect to consider in future research. Further experiments can be carried out to determine the optimal amount of labeled data needed to produce synthetic data meeting the desired level of accuracy in specific downstream tasks. Additionally, we show a more general approach to test for the information contained

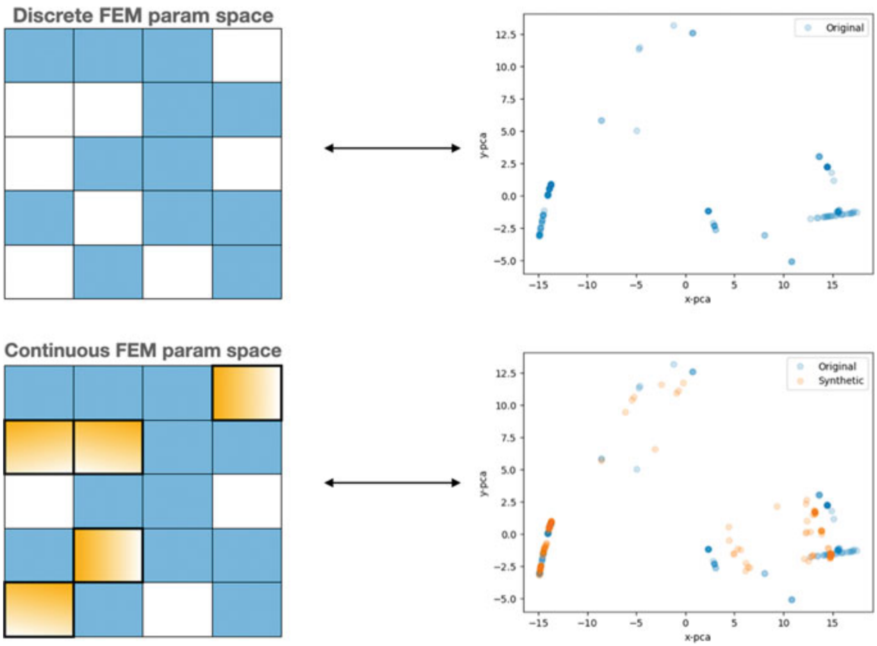


Fig. 4 Schematic illustration of a parameter grid used in FEM simulation and an indirect interpolation with a generative neural network. This indirect interpolation can be visualized using a PCA of the time-series data. The more space is filled in the PCA latent space between the original data, the more complete the process window is

in the synthetic data. Additionally, the use of GANs in other types of time-series data, such as those generated by other complex industrial processes, can be investigated to determine the generalizability of this approach. An end-to-end solution could be developed with industry partners for important use cases.

References

1. Bahdanau D, Brakel P, Xu K, Goyal A, Lowe R, Pineau J, Courville AC, Bengio Y (2016) An actor-critic algorithm for sequence prediction. CoRR abs/1607.07086 http://arxiv.org/abs/1607.07086
2. Bengio S, Vinyals O, Jaitly N, Shazeer N (2015) Scheduled sampling for sequence prediction with recurrent neural networks. CoRR abs/1506.03099 http://arxiv.org/abs/1506.03099
3. birdx0810: Implementation of the timegan in pytorch (2022). https://github.com/birdx0810/timegan-pytorch
4. Chen X (2019) Tennessee eastman simulation dataset. https://dx.doi.org/10.21227/4519-z502
5. Chernyavskiy A. Ilvovsky D, Nakov P (2021) Transformers: “the end of history” for nlp? CoRR abs/2105.00813 https://arxiv.org/abs/2105.00813
6. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North

American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186. https://aclanthology.org/N19-1423

7. Diao S, Shen X, Shum K, Song Y, Zhang T (2021) TILGAN: transformer-based implicit latent GAN for diverse and coherent text generation. In: Findings of the association for computational Linguistics: ACL-IJCNLP 2021. Association for Computational Linguistics, pp 4844–4858 (Online). https://aclanthology.org/2021.findings-acl.428
8. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2020) An image is worth 16 × 16 words: transformers for image recognition at scale. CoRR abs/2010.11929 https://arxiv.org/abs/2010.11929
9. Jiang Y, Chang S, Wang Z (2021) Transgan: two transformers can make one strong GAN. CoRR abs/2102.07074 https://arxiv.org/abs/2102.07074
10. Lamb A, Goyal A, Zhang Y, Zhang S, Courville A, Bengio Y (2016) Professor forcing: a new algorithm for training recurrent networks
11. Li X, Metsis V, Wang H, Ngu AHH (2022) TTS-GAN: a transformer-based time-series generative adversarial network. CoRR abs/2202.02691 https://arxiv.org/abs/2202.02691
12. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(86):2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html
13. Motamedi M, Sakharnykh N, Kaldewey T (2021) A data-centric approach for training deep neural networks with less data. arXiv preprint arXiv:2110.03613
14. Neubürger F, Arens J, Vollmer M, Kopinski T, Hermes M (2022) Coupled finite-element-method-simulations for real-time-process monitoring in metal forming digital-twins. In: 2022 10th International conference on control, mechatronics and automation (ICCMA), pp 260–265
15. Rieth CA, Amsel BD, Tran R, Cook MB (2017) Additional tennessee eastman process simulation data for anomaly detection evaluation. https://doi.org/10.7910/DVN/6C3JR1
16. Smith M (2009) ABAQUS/standard user's manual, version 6.9. Dassault Systèmes Simulia Corp, United States
17. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemom Intell Lab Syst 2(1): 37–52. https://www.sciencedirect.com/science/article/pii/0169743987800849 (Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists)
18. Yoon J, Jarrett D, van der Schaar M (2019) Time-series generative adversarial networks. In: Wallach H, Larochelle H. Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems. vol 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf

A Deep Learning Framework for Assamese Toxic Comment Detection: Leveraging LSTM and BiLSTM Models with Attention Mechanism

Mandira Neog and Nomi Baruah

Abstract As social media platforms grow in popularity, this research piece discusses the significance of creating a secure and positive online environment. The major goal is to protect users by detecting objectionable language in Assamese social media comments. The ultimate goal is to create a very effective mechanism for detecting toxic comments in Assamese, supporting a safe online environment. To address the lack of available datasets, a well-curated dataset was manually assembled for the experiment. Deep learning models such as LSTM and bidirectional LSTM (BiLSTM) were used to capture the contextual intricacies of user-generated comments. Notably, the BiLSTM model beats the LSTM model by including an attention mechanism, attaining a promising accuracy rate of 86.9% in successfully identifying toxic comments. Using the capabilities of the LSTM and BiLSTM models, a more robust and efficient approach for recognizing toxic phrases in Assamese is developed, aligned with the goal of building a secure, respectful, and toxic-free online environment.

Keywords LSTM · BiLSTM · Attention mechanism · Assamese · Toxic

1 Introduction

The emergence of social media platforms and the Internet has revolutionized how we exchange information and interact. By bringing together people from different communities, establishing relationships, and facilitating the dissemination of concepts, businesses, and causes, these platforms have become an essential part of our everyday lives. But the growing use of online interactive communication devices has also led to a mass of information [1], including highly toxic messages that may trigger bullying, harassment, and personal attacks. These toxic comments not only put someone's emotional and mental health in danger but also restrict people's ability to voice a variety of viewpoints.

M. Neog (✉) · N. Baruah
Dibrugarh University, Dibrugarh, Assam, India
e-mail: mandira.neog@gmail.com

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_37

Recognizing [2] the need to address concerns about offensive and hateful content on the Internet, social media administrators have resorted to manual review techniques to find and remove such information. However, this method requires a lot of work, takes an extended amount of time, and is ultimately unscalable and unsustainable. As a result, the presence of inappropriate content continues to be a serious problem for online platforms, frequently forcing communities to impose restrictions on or prohibit user comments entirely.

In this context [3], it has become clear that spotting and removing toxic comments is essential to maintaining a secure and encouraging online community. Although there is content filtering software, its effectiveness is limited to locating and blocking online pages and paragraphs with explicit language. Therefore, it is imperative to provide reliable and effective techniques to identify toxic comments [4], especially in languages with inadequate resources like Assamese. Despite the growing importance of toxic comment identification, relatively little research has been done in this field, especially for Indian languages and Assamese in particular. This research gap emphasizes the significance of focused research into toxic comment identification in Assamese, with the objective of developing a viable approach that can contribute to a safer online environment. The major objective of this research article is to fill in the previously mentioned gap by outlining an effective method for finding toxic Assamese comments on social media. Deep learning techniques, specifically [5] long-short-term memory (LSTM) and [6] bidirectional LSTM (BiLSTM), are used in research because of their success in natural language processing tasks such as sentiment analysis and emotion recognition. In this research, we use the power of LSTM and BiLSTM models, supplemented with an [7] attention mechanism, to capture the contextual significance of Assamese comments. Through deep learning techniques, this strategy tries to address the issues caused by the language's intrinsic ambiguity and increase text understanding. The goal of the research is to overcome the inherent ambiguity of the Assamese language by utilizing the strengths of the LSTM and BiLSTM models to capture the contextual meaning of Assamese comments. By utilizing these deep learning approaches, we want to lessen the difficulties involved with comprehending and interpreting Assamese text, resulting in a more accurate and nuanced analysis of the comment's contextual importance.

The development of a manually curated dataset [8] that was specially designed for the experiment is a significant contribution to this research. This dataset, which was gathered from numerous public domains, ensures the data's relevance and applicability, greatly improving the research's impact and credibility. The overall goal of this research article is to close the toxic comment identification gap for Assamese social media users by utilizing deep learning techniques and a properly crafted dataset.

A thorough analysis of the research findings is included in the paper's systematic methodology. It starts with an introduction in Sect. 1 that emphasizes the importance of identifying toxic comments in Assamese and the necessity of closing the current research gap. Section 2 discusses the toxic comment notion, which is important in the context of toxic/non-toxic classification. The literature review in Sect. 3 sheds light on the current state of research on the topic. Section 4 describes the technique, emphasizing the LSTM and BiLSTM models for optimal context acquisition. Section 5 examines the model performance for toxic comment detection in depth. Section 6 results include experimental outcomes and evaluation standards. The conclusion of Sect. 7 emphasizes the necessity of toxic comment identification for a safer online environment by summarizing major findings and suggesting future research.

2 Domain Overview

Hate Speech: Toxic comments and hate speech [5] are inextricably linked, and knowing what constitutes hate speech is critical in deciding whether a comment is toxic or non-toxic. According to the Cambridge Dictionary, hate speech is a public speech that expresses hatred or supports violence against individuals or groups based on numerous qualities such as race, religion, gender, sexual orientation, nationality, social class, or religious beliefs. As hate speech frequently involves the use of disparaging language and seeks to cause damage or discrimination against specific individuals or communities, this definition serves as a foundation for recognizing and assessing the toxic nature of statements. We can successfully recognize and handle toxic comments on online platforms by knowing the meaning of hate speech, fostering a safer and more inclusive environment for users.

Verbal and written communication plays an important part in our lives, impacting numerous areas and serving as frequently used mediums of expression. Words have a lot of contextual and emotional weight. As a result, toxic speech occurs when a term is accompanied by negative and offensive feelings, such as the use of harsh words in a discourse. While criticism is a normal aspect of communication, it is crucial to promote constructive criticism as a preferred manner of expressing unhappiness over insults.

3 Related Work

This section offers a thorough analysis of different research studies on the subject of recognizing toxic comments. The application of deep learning methods for efficient toxic comment recognition is the key area of focus. The research takes into account recent publications over the last 5 years, providing insights into trends, limitations, and potential future directions in this field. While the research's main emphasis is on deep learning techniques, it also considers how they may be applied to other languages besides Indian, providing insights into innovative methods to use deep learning for the detection of toxic comments. Table 1 provides a succinct breakdown of the key findings and the methods used for the literature review.

Dubey et al. [5] propose employing LSTM neural networks for text mining to identify hostile and abusive remarks with 94.49% precision, 92.79% recall, and 94.94% classification accuracy.

Xu et al. [6] improve sentiment analysis of online comments with emotion-enriched TF-IDF weighted word vectors, outperforming RNN, CNN, LSTM, and NB with exceptional accuracy using their BiLSTM model.

An LSTM-based technique for sentiment categorization is introduced by Murthy et al. [9] to address difficulties in analyzing in-depth user opinions. With enhanced training data, the deep learning system LSTM performs well in sentiment analysis.

Tripathi et al. [10] examine the NB, SVM, and LSTM algorithms for consumer sentiment analysis, discovering that LSTM excels in whole phrase analysis and Bernoulli NB excels in aspect-based classification, implying that larger datasets could enhance accuracy.

AEC-LSTM, a deep neural network model for sentiment classification, is introduced by Huang et al. [11]. It combines emotion psychology with topic attention, convolutional ELSTM, and an enhanced LSTM cell to succeed in typical sentiment tasks. Long et al. [12] use BiLSTM networks with Multi-head Attention (MHAT) to assess sentiment in Chinese social media. Their method outperforms existing techniques by handling complexity and capturing context with ease. In order to address difficulties and ambiguities, Elfaik et al. [13] introduce BiLSTM for sentiment analysis in Arabic text. Studies show that F1 is 92.39% better than current deep learning and conventional methods.

A recurrent attention LSTM is suggested by Zhang et al. [14] for objective document-level sentiment analysis. With the help of a joint loss function, attention is focused on important sentimental words. For the IMDB, Yelp, and Amazon datasets, the model outperforms cutting-edge techniques.

Muhammad et al. [15] use Word2Vec and LSTM for sentiment analysis of Indonesian hotel reviews, attaining an accuracy variance of 8.56% on 2500 reviews. Skip-gram, Hierarchical Softmax, and 300 dimensions are optimal Word2Vec parameters.

Gandhi et al. [16] use word recognition, Word2Vec, and deep learning models to improve Twitter sentiment analysis. On the IMDB dataset, their technique achieves 87.74% and 88.02% testing accuracy with CNN and LSTM, respectively.

For the sentiment analysis of tweets, Srivastava et al. [17] presented CNN-BiLSTM and BiLSTM models, all without the need for specialized word embeddings. With a success rate of 0.84 and 0.80, respectively, these models outperformed standard classifiers in demonstrating the effectiveness of deep neural networks in sentiment analysis on social media. For the purpose of analyzing the sentiment of code-mixed writings in Dravidian and English, Anusha et al. [18] presented a BiLSTM model. The model performed 13th, 14th, and 14th in the FIRE 2021 task, with F1 scores of 0.563, 0.604, and 0.365 for Tamil, Malayalam, and Kannada language pairs, respectively.

Wei et al. [19] offer a BiLSTM model with multi-polarity orthogonal attention for better implicit sentiment analysis, which captures sentiment variations in word orientation effectively.

For sentiment categorization, Hameed et al. [20] provide a computationally efficient single-layered BiLSTM model for real-time sentiment analysis, attaining a high accuracy of 90.585% across varied datasets.

Table 1 Comparison of various sentiment analysis work done in deep learning methods

Refs.	Language	Platform	Dataset	Methods	Results
[5]	English	Twitter	56,745	LSTM	Acc—94.94%
[6]	Chinese	Review	15,000	BiLSTM	F1—92.18%
[8]	English	TripAdvisor, Amazon and IMDB	1m	LSTM	Acc—85%
[9]	English	IMBD	50,000	LSTM	Acc—92.8%
[10]	Nepali	Twitter	4035	NB, SVM, LSTM	Acc—79%
[11]	Chinese	Review	19,465	BiLSTM+Multi head+Attention mechanism	F1—95.33%
[12]	English	IMDB, Yelp datasets, Amazon	_	Recurrent attention LSTM, BiLSTM	Acc—71.4%
[13]	Arabic	Health services, reviews, Twitter	61,582	BiLSTM	F1—92.39%
[14]	English	IMDB, Yelp and Amazon	–	Recurrent attention LSTM, BiLSTM	Acc—71.4%
[15]	Indonesian	Hotel reviews	2500	LSTM	Acc—85.96%
[16]	English	IMBD	50,000	CNN , LSTM	Acc—88.02%
[17]	English	Twitter	_	CNN, BiLSTM	Acc—84%
[18]	Dravidian language	YouTube	71,309	BiLSTM	F1—0.56%
[19]	Chinese	SMP2019	_	BiLSTM	_
[20]	_	Review	70,275	BiLSTM	Acc—90.585%
[21]	English	–	–	BERT, DistilBERT, BiLSTM, TNC	ACC-85.13%
[22]	English	Public dataset	81,122	BERT , BiLSTM	F1—96.23%
[23]	Serbian	YouTube, Newspaper portals	118,876	BiLSTM	Precision-97.00%

Lin et al. [21] present a hybrid technique for Indonesian sentiment analysis that uses BERT, DistilBERT, BiLSTM, and TCN to achieve above 85% accuracy by mixing contextual and semantic data.

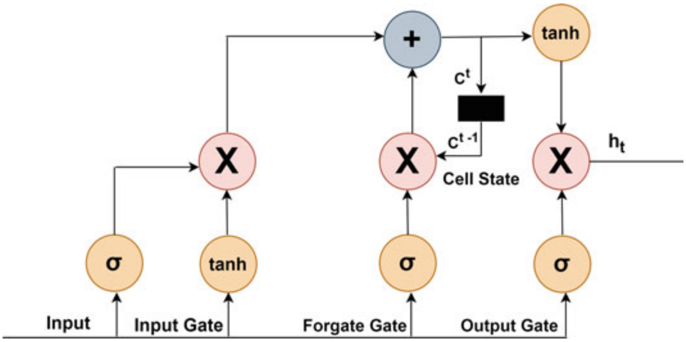


Fig. 1 LSTM unit

Saleh et al. [22] improve hate speech detection accuracy by using domain-specific word embeddings and deep learning models like BiLSTM, getting up to a 96.23% F1-score.

Stankovi et al. [23] created a BiLSTM deep neural network for detecting hate speech in sports on social media with great precision (96 and 97%).

4 System Design

The LSTM architecture is made up of memory blocks that are similar to computer memory chips. Recurrently coupled memory cells and three gates—input, output, and forget—are included in these blocks. Figure 1 shows [24] the structure of the LSTM unit. In a series, these gates determine what information is retained, deleted, and revealed. Memory cells are refreshed by integrating past information with the outputs of the input and forget gates, as demonstrated in Fig. 1.

4.1 LSTM

The processed data created by applying an output gate to the updated memory cell is represented by the hidden state. Training entails fine-tuning the learned parameters in order to increase LSTM performance.

These gates control the information flow for the current time step. The cell [24] is defined in formulas (1), (2) and (3) as follows:

Formula:

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

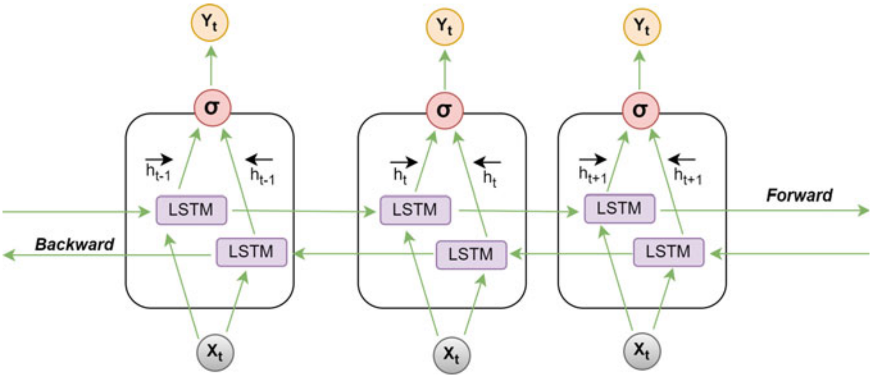


Fig. 2 BiLSTM unit

$$f_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_f\right) \tag{2}$$

$$o_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_o\right) \tag{3}$$

To select the input gate i_t, apply the sigmoid function to the weighted sum of the current input x_t at the current time step, biases b_i, and the prior LSTM output $h_t - 1$ at the previous step in time t-1. This design also allows for the usage of other gates, such as the forget and output gates. The LSTM is composed of two successive layers, each with 128 units. However, a bidirectional LSTM—another term for a bidirectional encoder—can also be created by stacking two LSTMs. Bidirectional LSTM generates a number of hidden outputs, which are discussed in more detail below.

4.2 BiLSTM

The Bidirectional Long Short-Term-Memory(BiLSTM) is an RNN that processes input data in both directions, making it useful for tasks such as sentiment analysis and text categorization. It blends forward and backward layers, which improves context understanding and task comprehension. The architectural layout of BiLSTM is depicted [25] in Fig. 2.

Here, BiLSTM is used to increase dataset generalization and the influence of fitting network properties. Equations (4), (5) and (6) give the calculation formula and illustrate the basic concept of Fig. 2.

$$\overrightarrow{h_t} = \sigma\left(W_{[xh]}x_t + W_{[h][h]}\overrightarrow{h_{t-1}} + b_{[h]}\right) \tag{4}$$

$$\overleftarrow{h_t} = \sigma\left(W_{[xh]}x_t + W_{[h][h]}\overleftarrow{h_{t-1}} + b_{[h]}\right) \tag{5}$$

$$H_t = W_{[xh]}\overrightarrow{h_t} + W_{[hy]}\overrightarrow{h} + b_y \tag{6}$$

Formulas (4), (5) and (6) depict the BiLSTM [25] expression, which stands for the activation function. The hidden layer's input is denoted by the symbol h_t, and the output is created by updating both the forward and backward structures ($\rightarrow h_t$ and $\leftarrow h_t$), respectively. These formulas represent the information transformation and flow operations used by the BiLSTM to enable the bidirectional processing of the input sequence.

4.3 Data Preprocessing

Dataset Due to the lack of an appropriate dataset, we manually gathered a dataset of 20,000 comments from several public domains, including Facebook, YouTube, and news portals. 17,500 of these comments are labeled as non-toxic, while 2500 are labeled as toxic. This distribution shows that 87.5% of the comments are non-toxic, whereas 12.5% are toxic.

Data preprocessing We manually collected Assamese language data from social media networks to ensure the reliability and quality of the data used in our experiment. However, we noticed issues with data quality and representativeness. To address these concerns, we carried out the necessary preparation procedures, such as data cleansing, filtering, and normalization. We gave binary classification tags to the data and combined them into a single CSV file encoded in Assamese script and UTF-16 LE format. In addition, to improve the content, we used techniques such as eliminating URLs, special characters, "@ symbols," stop words, stemming, and lemmatization. These preprocessing efforts increased the accuracy and effectiveness of our analytical and modeling processes dramatically.

Data Embedding In text analysis and the processing of natural languages (NLP), word embedding is developing into a helpful technique. It represents words with numerical vectors in order to find significant patterns and contextual relationships. AI models can grow more accurate and efficient by understanding the syntactic and semantic links between words and minimizing the dimensionality of text input. We must select both the vocabulary size and the vector dimensions for the layer that embeds to map individual words to fixed-size 100-dimensional vectors. The inherent length of an input sequence defines its maximum length. The embedding layer's output is then further processed by dense layers, dropout layers, and neural network layers, all of which increase the overall performance of the AI model.

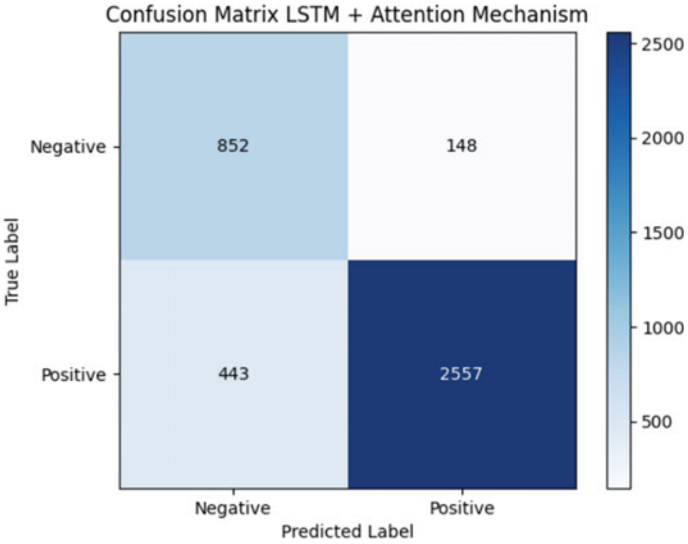


Fig. 3 Confusion matrix LSTM + Attention mechanism model

5 Implementation of the Models

This section evaluates the performance of various models used to detect toxic Assamese comments. We examine the strengths, shortcomings, and overall usefulness of the LSTM and BiLSTM models for their accuracy in toxic comment detection.

LSTM with Attention Mechanism
The LSTM model for binary classification first attains 81.97% accuracy, 86.5% F1 score, 79.0% recall, and 95.3% precision by means of tokenization, padding, and splitting. By adding the attention mechanism, accuracy is improved to 85.22%. This technique addresses the equal treatment of sequence components by allocating attention selectively. The model targets negative comments with 64 hidden units and 0.2 dropout regularization. The confusion matrix of the LSTM with the attention mechanism is shown in Fig. 3. The following outcomes were obtained from the model's evaluation using a dataset that had 4000 comments: There were 443 false negatives, 148 false positives, 852 genuine negatives, and 2557 true positives. Reducing the number of false positives and false negatives is important to improve the sentiment prediction accuracy of the model, especially when it comes to identifying toxic statements. This enhancement calls for additional research and improvement.

BiLSTM with Attention Mechanism The BiLSTM model detected toxic comments with a noteworthy accuracy of 86.87%. It made use of a BiLSTM layer with 64 hidden units, a dense layer with sigmoid activation, and 0.2 dropout regularization. Training used the Adam optimizer for 20 iterations with a batch size of 32 and an embedding

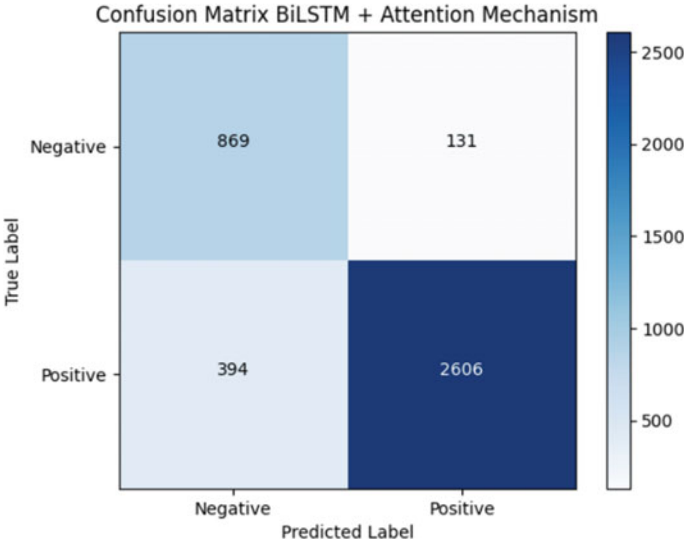


Fig. 4 Confusion matrix LSTM + Attention mechanism model

layer based on tokenized data. The presence of an attention mechanism increased the accuracy of the refined BiLSTM model to 86.9% or 0.1%. This improvement can be attributed to the attention mechanism's capacity to rank critical input segments. The model is composed of three layers: bidirectional LSTM, embedding, and attention mechanism. The context vector formed from the LSTM outputs is shaped in part by the attention scores. The confusion matrix in Fig. 4 illustrates the functioning of the BiLSTM with attention mechanism. When this model was tested on a dataset of 4000 comments, it found 2606 true positives, 869 true negatives, 131 false positives, and 394 false negatives. Sentiment prediction accuracy will rise with a decrease in false positives and negatives.

6 Results and Discussion

The research compared the efficiency of two deep learning models LSTM and BiLSTM with attention mechanisms in detecting toxic comments. The BiLSTM with attention mechanisms model scored an amazing 86.9% accuracy, 87.50% precision, 98.00% recall, and 92.45% F1 score. The LSTM model achieved 81.97% accuracy, 95.3% precision, 79.0% recall, and an F1 score of 86.5%. Table 2 gives an in-depth analysis of the outcomes of the suggested models. The addition of the attention mechanism increased LSTM accuracy to 85.22%, precision to 95.57%, recall to 85.26%, and F1 to 89.67%. The suggested BiLSTM attention mechanisms model outperformed the baseline model, reaching 86.9% accuracy versus the baseline's 86.87%.

Table 2 Comparing performance of the proposed models

Methods (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
LSTM	81.97	95.30	79.00	86.50
LSTM + Attention mechanism	85.22	95.57	85.26	89.67
BiLSTM	86.87	87.58	98.00	92.45
BiLSTM + Attention mechanism	86.90	87.50	98.00	92.45

Table 3 Contrasting the proposed model's performance with existing LSTM and BiLSTM models

Refs.	Author	Language	Methods	Results (%)
[10]	Milan Tripathi	Nepali	LSTM	79.00
[15]	Muhammad et. al.	Indonesian	LSTM	85.96
Proposed approach		Assamese	LSTM + Attention mechanism	85.22
[18]	Anusha et. al	Dravidian	BiLSTM	56.00
Proposed approach		Assamese	BiLSTM + Attention mechanism	86.90

Table 3 compares the innovative BiLSTM + Attention mechanism model to the traditional LSTM and BiLSTM models, as well as other existing ones. The evaluation has been carried out within the framework of the SOV language structure and two deep learning models LSTM and BiLSTM. Although the model's performance varies depending on language structure, the LSTM model outperforms previous research in sentiment analysis for languages with subject-object-verb (SOV) structure, obtaining an accuracy of 85.22%. In comparison, the BiLSTM model shows even more promising findings, with an outstanding accuracy of 86.90% in recognizing toxic comments in Assamese. By effectively distinguishing between true positives and negatives, the BiLSTM model's prediction performance is enhanced by the addition of an attention mechanism. Although there is room for development, it showed excellent generalization and 86.9% accuracy. It might function even better if data quality and quantity are increased. The approach may be able to identify toxic comments, suggesting further research directions for its development.

7 Conclusion

In conclusion, this research concentrated on finding potentially toxic Assamese comments on social media. Deep learning approaches such as LSTM and BiLSTM use attention mechanisms to improve their performance. The BiLSTM method with the attention mechanism performed better than average in detecting toxic content. The importance of this research depends on the immediate requirement to recognize and delete inappropriate comments in order to promote a polite and safe social media environment, as such content may have detrimental effects. A problem was that there was not a suitable digital dataset available in Assamese. A dataset of 20,000 sentences that were manually collected and annotated from various Internet sources was used to address the issue. The accuracy of toxic comment detection is expected to significantly increase with the quantity of the dataset. In order to improve detection skills in this particular environment, future research should concentrate on larger datasets and investigate different deep learning models.

References

1. Neelakandan S, Sridevi M, Saravanan C, Murugeswari K, Singh Pundir AK, Sridevi R, Lingaiah TB (2022) Deep learning approaches for cyberbullying detection and classification on social media. Computat Intell Neurosci 11:1–13
2. Vaidya A, Mai F, Ning Y (2020) Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. In: ICWSM. 2020 May 26, vol 14(1). pp 683–9
3. Maslej-Krešňáková V, Sarnovský M, Butka P, Machová K (2020) Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification. Appl Sci 10(23):8631
4. Deka RR, Kalita S, Bhuyan MP, Sarma SK (2020) A study of various natural language processing works for assamese language. In: Dawn S, Balas V, Esposito A, Gope S, (eds) Intelligent techniques and applications in modern science and technology. ICIMSAT 2019. Learning and analytics in intelligent systems. vol 12. Springer, Cham, pp 6–15
5. Dubey K, Nair R, Khan MU, Shaikh S (2020) Toxic comment detection using LSTM. ICAECC. 3rd edn. IEEE Xplore, pp 1–8
6. Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access 7:51522–32
7. Liu G, Guo J (2019) Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–38
8. Appidi AR, Srirangam VK, Suhas D, Shrivastava M (2020) Creation of corpus and analysis in code-mixed kannada-english twitter data for emotion prediction. In: Proceedings of the 28th international conference on computational linguistics; 2020 Dec 8–13; Barcelona, Spain (Online): International Committee on Computational Linguistics, pp 6703–9
9. Murthy GS, Allu SR, Andhavarapu B, Bagadi M, Belusonti M (2020) Text based sentiment analysis using LSTM. Int J Eng Res Technol 9(5):299–303
10. Tripathi M (2021) Sentiment analysis of Nepali Covid19 tweets using nb svm and LSTM. J Artif Intell 3(03):151–68
11. Huang F, Li X, Yuan C, Zhang S, Zhang J, Qiao S (2021) Attention-emotion-enhanced convolutional LSTM for sentiment analysis. IEEE Trans Neural Netw Learn Syst 33(9):4332–45
12. Long F, Zhou K, Ou W (2019) Sentiment analysis of text based on bidirectional LSTM with multi-head attention. IEEE Access 20(7):141960–9
13. Elfaik H, Nfaoui EH (2020) Deep bidirectional LSTM network learning-based sentiment analysis for Arabic text. J Intell Syst 30(1):395–412
14. Zhang Y, Wang J, Zhang X (2021) Conciseness is better: recurrent attention LSTM model for document-level sentiment analysis. Neurocomputing 28(462):101–12
15. Muhammad PF, Kusumaningrum R, Wibowo A (2021) Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews. Proc Comput Sci. 1(179):728–35

16. Gandhi UD, Malarvizhi PK, Chandrababu G, Karthick G (2021) Sentiment analysis on twitter data by using convolutional neural network (CNN) and long short term memory (LSTM). Wireless Personal Commun 17:1–10
17. Srivastava T, Arora D, Sharma P (2023) Sentiment analysis of COVID-19 Tweets Using BiLSTM and CNN-BiLSTM. ICRTC 2022. In: Proceedings of international conference on recent trends in computing. Lecture notes in networks and systems, Singapore, Springer Nature, Mar 21 2023, pp 523–35
18. Anusha MD, Shashirekha HL (2021) BiLSTM-sentiments analysis in code-mixed Dravidian Languages. FIRE 2021. In: Proceedings of forum for information retrieval evaluation, 13-17 Dec 2021, India, CEUR-WS vol 3159. pp 6–13
19. Wei J, Liao J, Yang Z, Wang S, Zhao Q (2020) BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis. Neurocomputing 28(383):165–73
20. Hameed Z, Garcia-Zapirain B (2020) Sentiment classification using a single-layered BiLSTM model. IEEE Access 17(8):73992–4001
21. Lin CH, Nuha U (2023) Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy. J Big Data 10(1):1–19
22. Saleh H, Alhothali A, Moria K (2023) Detection of hate speech using BERT and hate speech word embedding with deep model. Appl Artif Intell 37(1):384–405
23. Vujici'c Stankovi'c S, Mladenovi'c M (2023) An approach to automatic classification of hate speech in sports domain on social media. J Big Data 10(1):1–6
24. Naqvi U, Majid A, Abbas SA (2021) UTSA: Urdu text sentiment analysis using deep learning methods. IEEE Access 12(9):114085–94
25. Yang M, Wang J (2022) Adaptability of financial time series prediction based on BiLSTM. Proc Comput Sci 1(199):18–25

Security in VANETs with Insider Attack Resistance and Signature Aggregation

Vijaya Lode, Kekhelo Lasushe, and Anil Pinapati

Abstract Vehicular Ad-hoc Networks (VANETs) are a type of network in which vehicles communicate with one another and exchange information so as to provide quality-of-life improvements to the vehicle users as well as the people belonging to the area. In VANET, vehicles share information such as the current status of the vehicle, the status of the traffic or the status of the road conditions to the other vehicles. All this information requires the network to be highly secure. Therefore various schemes have been proposed to secure the network. However, they suffer when a receiver has to verify multiple incoming message signatures. To reduce the verifying time and size of the signature The proposed scheme provides an efficient pairing-free aggregate signature, it will verify multiple messages by combining multiple signatures into a single signature called aggregate signature, and it is also resistant to insider attacks without using the tamper-proof device.

Keywords Vehicular ad-hoc networks (vanets) · Elliptic curve cryptography (ecc) · Aggregate signature

1 Introduction

With the advancement in wireless technology and electronics becoming cheaper, the possibility of connected vehicles and intelligent transportation systems has expanded. These new technologies have provided many benefits [1]. However, these same technologies have also introduced new vulnerabilities to the security system that did not

V. Lode (✉) · K. Lasushe · A. Pinapati
Department of Computer Science and Engineering, National Institute of Technology, Calicut, Kerala 673601, India
e-mail: vijaya_p220102cs@nitc.ac.in

K. Lasushe
e-mail: kekhelo_m210376cs@nitc.ac.in

A. Pinapati
e-mail: anilpinapati@nitc.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_38

exist before, including privacy breaches and personal information thefts [2], which may result in the misuse of false information for personal benefits [3]. For all these reasons, it is important to devise a security scheme for the vehicles participating in Vehicular Ad-hoc Networks (VANETs) and also to reduce the time taken to verify multiple messages. This can be done with the help of signature aggregation. Signature aggregation is the process of combining multiple signatures into a single signature, and the final aggregated signature is taken for verification and security is tested with different attack resistances.

1.1 VANET Architecture

In VANETs, there are three important components namely trusted authority (TA), roadside unit (RSU) and onboard unit (OBU).

The TA is the top tier among these components, and it is responsible for managing the entire network. The role of RSU is to host an application that connects with other networking devices. These RSUs are connected to one another and finally connected to the TA by a wired network. The OBU is equipped with the vehicles, and it is responsible for collecting information about the vehicle such as traffic, speed, fuel, etc. [1]. They are briefly described as follows:

(1) Trusted Authority: The Trusted Authority (TA) manages the entire system by including the registration of OBUs and RSUs. Also, the TA is responsible for ensuring the security of the system. This can be done by authenticating the vehicles to avoid any harm to the other vehicles. The TA utilizes a large memory size and a high amount of power and it has the ability to reveal the OBU's identities in case of any malicious behaviors.

(2) Roadside Unit: RSUs are middle-level nodes fixed on nearby roads. RSUs are computing devices, and they provide local connectivity to the vehicles in the area.

(3) Onboard Unit: OBUs are lowest-level GPS-based tracking devices mounted on vehicles that allow the vehicles to share information with the other OBUs and the RSUs. The main function of the OBU is to connect with the RSU and the other OBUs through wireless links. The OBUs take input power from the batteries of the vehicle.

1.2 VANET Characteristics

Compared to MANETs, VANETs have unique characteristics. These characteristics are discussed below:

(1) High Mobility: VANETs are very mobile compared to other ad-hoc networks as the vehicles keep on moving in random directions and change their location

constantly. This makes it difficult to estimate the topology of the network. Due to this high mobility, the communication time between the nodes in VANETs is very short.

(2) Dynamic Network Topology: The topology of VANET is rapidly changing and not constant because of the high mobility of the vehicles.
(3) Wireless Communication: The way the nodes are connected and the way they communicate are all achieved through a wireless medium.
(4) Driver Safety: VANETs can improve the safety of drivers, and the quality of life, enhance the comforts of the passengers and also improve the flow of traffic.
(5) Network Strength: The strength of the network in VANETs depends on the flow of traffic on the street. The strength is low in the case when there are fewer vehicles on the road and high when there are traffic jams.
(6) Large Network: The networks are larger in highways and entry and exit points of a city.
(7) Volatility: The volatility in VANETs is because of the high mobility of the vehicles.

1.3 VANET Security

Compared to other IoT systems, the most important feature of VANETs is high mobility. This high mobility causes group membership to change at a higher rate than the other IoT systems [4]. In addition, the wireless connection and the group membership are easily accessible to the public so participants in a VANETs are considered untrusted.

(1) Integrity: Message Integrity is essential for securing the message exchange system. The system must be able to detect and prevent malicious messages both intended or accidental. Message integrity also ensures that the message has not been modified in the process of transfer between the nodes.
(2) Authentication: Integrity alone is not sufficient for ensuring security in VANET. It is also essential to ensure that only verified or authorized users are able to participate in generating messages in the system. To achieve message authentication, message signatures are used to sign messages. Members with non-valid signatures are not authorized.
(3) Tracing: The term tracing means matching a message to the message sender. With tracing it is also possible to reveal the identity of the sender. This term where the identity of the sender may be revealed whenever necessary is termed as conditional privacy.
(4) Privacy: In VANET, privacy means protecting the identity of the users/drivers. This real-world identity can be registration information which can be used to identify the user.

(5) Non-Repudiation: By non-repudiation, a vehicle that sent a particular message cannot hide or deny the fact that it sent the message. This is essential to identify the misbehaving vehicles and stop them from interfering with the network.

(6) Revocation: Revocation is used to remove access to malicious vehicles. This ensures that the revoked vehicles can no longer prove their group membership with whatever information they possess.

(7) Insider Attack Resistance: VANET is a dynamic membership system, any vehicle can move in and out anytime. This allows an attacker to freely join the system by forging or stealing an identity certificate. Considering this, it is worth looking into these risks of insider attacks [8].

(8) Man-in-the-middle-Attack: An attacker successfully establishes the connection with the communicating parties and takes the control of entire communication, and the communicating parties will think that they are talking to each other through a secure channel.

2 Related Works

In most VANET schemes, vehicles are grouped together based on their area. A vehicle joins a group by requesting group membership from one of the upper levels. After being authenticated, the vehicle is granted a signature key which is used to sign messages so as to communicate with other vehicles. These keys can be group keys or symmetric and asymmetric keys. In VANET broadcasting messages is usually desirable; however, there are some schemes that don't support it. One-to-one symmetric keys are usually used in these schemes [5–7]. In a one-to-one symmetric key scheme, the vehicles in the same group share a unique key with each other. However, this scheme is impractical as this will require a large number of keys. Some other schemes are proposed where RSU handle all communications, but this is not desirable as this will insert an extra hop of communication [8].

Most of the VANETs require message broadcasting, due to this almost all the scheme uses group keys or group signatures. In all these schemes, a single symmetric key is shared among all the group members to communicate with each other [9–11]. However, the problem with these schemes is that they are vulnerable to insider attacks.

In [11] Vijayakumar et al. proposed a scheme based on elliptic curve pairings which provides message integrity, and authenticity of the sender by using identities, non-repudiation and tracing. However, the TA knows all secret values, so the compromised TA will falsify signatures to launch an insider attack and also it is using complex pairing operations.

In 2021 Funderburg et al. [12] proposed a scheme without pairing that is resistant to insider attacks, i.e., the attacks not only by other vehicles but also in situations when the TA is compromised and the information is vulnerable to the attackers. In this scheme, each vehicle signature key certification is generated by the TA, but with this signature key certification, the TA still will not be able to generate a valid message signature as it requires the vehicle's private key. The scheme also provides tracing and

non-repudiation. While this is an acceptably great scheme, it is not efficient enough when handling enormous incoming messages as the receiver needs to validate all the messages one by one. The scheme proposed by Han et al. [13] provides signature aggregation, i.e., signatures are aggregated so it is more efficient for the receiver to validate multiple messages. However, this scheme mostly considers the RSUs to manage the aggregation thus inserting an extra hop of communication.
In our proposed scheme we used signature aggregation to validate multiple signatures in a single verification and our scheme was meeting important security requirements like privacy, authentication, nonrepudiation resistance to insider attacks, and man-in-the-middle attacks.

3 Proposed System

The typical TA-RSU-OBU hierarchy is used in the proposed scheme. The RSUs are communication relay nodes, the TA will authenticate the OBUs, and the OBUs will communicate by sharing information with RSUs and other OBUs. The process of the proposed scheme is shown in Fig. 1 [13].

3.1 Initialization

All the parameters which are used in this paper are listed in the following Table 1.

For a vehicle to join the system, firstly the vehicle owners contact the appropriate government authority and obtain a signed vehicle registration certificate. Then the vehicle joins the system. The vehicle registration certificate contains the vehicle

Fig. 1 Process of the proposed scheme

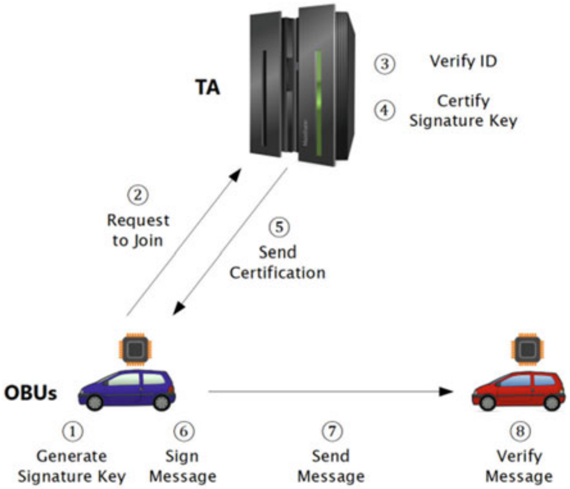


Table 1 Initialization

Symbol	Definition
G	Elliptic curve generator point
α	Private key of group
G_{pub}	Public key of group
β	Private key of OBU i
V_{pub}	Signature/public key of OBU i
$\sigma_{V_{pub}}, X$	Signature key certification of OBU i
M	Message
t	Timestamp
σ_M, Y	Message signature parameters
H	Hash function that maps ecc point to $\mathbb{Z}_p^*$

owner's identity. The vehicles in the system are grouped together based on their geographic area.

For each group, the TA will generate a private/public key pair as:

$$\alpha \in \mathbb{Z}_p^*$$

$$G_{pub} = \alpha * G$$

For each group, the public parameters are G, G_{pub}, and H where H is a secure hash function that maps to $\mathbb{Z}_p^*$.

3.2 Vehicle Joins a Group

Whenever a vehicle wants to enter a new group, it generates a public/private key pair as:

$$\beta \in \mathbb{Z}_p^*$$

$$V_{pub} = \beta * G$$

Then, in order to join a group, the vehicle sends a request to the TA. The TA will verify the identity of the vehicle, and if it is valid, the TA will map the vehicle's public key to its registration information that is stored by the TA to allow tracing if required in the future. But, to prevent a vehicle from changing its public key and thwarting the tracing, the TA chooses a random number $a \in \mathbb{Z}_p^*$ and sign as follows:

$$X = a * G$$

$$\sigma_{V_{pub}} = \alpha + a * H(V_{pub}||X)$$

Finally, vehicle i receives $\sigma_{V_{pub}}$ and X from TA.

3.3 Vehicle Sends a Message

In order to send a message, a vehicle signs the message using the signature key certification as well as its private key. This signature is verified by the receiver and therefore only messages with valid signatures are accepted. To sign a particular message, a vehicle chooses a random number $b \in \mathbb{Z}_p^*$ and generates a timestamp t. Then the signature is generated as shown:

$$Y = b * G$$

$$\sigma_M = \sigma_{V_{pub}} + \beta + b * H(M||t||Y)$$

After the signature is generated, vehicle i will broadcast message M. Along with the message some other parameters are also broadcasted. These are σ_M, V_{pub}, X, Y and t.

3.4 Receiver Validates the Message

Firstly, the receiver checks the timestamp and compares it with $t_{now} - t > t_{replay}$ to avoid replay attacks. If $t_{now} - t > t_{replay}$, the receiver will assume the message as a replay attack and discard the message. If $t_{now} - t < t_{replay}$, the receiver proceeds the message for verification as shown:

$$\sigma_M * G = G_{pub} + X * H(V_{pub}||X) + V_{pub} + Y * H(M||t||Y)$$

The proof of the equation is given as shown:

$$\begin{aligned}
&\sigma_M * G \\
&\qquad = (\sigma_{V_{pub}} + \beta + b * H(M||t||Y)) * G \\
&\qquad = ((\alpha + a * H(V_{pub}||X)) + \beta + b * H(M||t||Y)) * G \\
&\qquad = (\alpha * G + a * G * H(V_{pub}||X) + \beta * G + b * G * H(M||t||Y)) \\
&\qquad = G_{pub} + X * H(V_{pub}||X) + V_{pub} + Y * H(M||t||Y)
\end{aligned}$$

3.5 Revocation

When malicious vehicles are detected, they are revoked by the TA. For this, a new group key is chosen by the TA and the signature key certifications of all the vehicles except the malicious vehicles are regenerated. The signature validation from these revoked vehicles will fail as shown:
Firstly, the vehicle generates a signature σ'_M using its old signature key certification and its signature key:

$$\sigma'_M * G = (\sigma'_{V_{pub}} + \beta' + b * (H(M||t||Y)) * G$$

When the receiver receives the message, it validates the signature using the new group key:

$$\sigma'_M * G = G_{pub} + X' * H(V_{pub}'||X') + V_{pub}' + Y * H(M||t||Y)$$

But,

$$\begin{aligned}\sigma'_M * G\\ &= ((\alpha' + a' * H(V_{pub}'||X')) + \beta' + b * H(M||t||Y)) * G\\ &= (\alpha' * G + a' * G * H(V_{pub}'||X') + \beta' * G + b * G * H(M||t||Y))\\ &= G_{pub}' + X' * H(V_{pub}'||X') + V_{pub} + Y * H(M||t||Y)\end{aligned}$$

Therefore for revoked vehicles, the validation will fail as the new group key G_{pub} is not equal to the old group key G_{pub}'

$$G_{pub}' + X' * H(V_{pub}'||X') + V_{pub} + B * H(M||t||Y)$$

$$\neq G_{pub} + X' * H(V_{pub}'||X') + V_{pub} + B * H(M||t||Y)$$

3.6 Signature Aggregation

When a vehicle receives multiple messages $\{M_i, \sigma_{Mi}, V_{pub_i}, t_i, X_i, Y_i\}$, $i \in$ (1,2,3, .., n), from different vehicles (V_1, V_2, .., V_n), the receiver takes all the signatures (σ_{M1}, σ_{M2}, ... , σ_{Mn}) and sum them up to get

$$\sigma_M = \sum_{i=1}^{n} \sigma_{Mi}$$

The validity of the signatures is verified by the receiver as shown in the equation:

$$\begin{aligned}&\sigma_M * G\\&\quad= \sum_{i=1}^{n} G_{pub} + \sum_{i=1}^{n} (X * H(V_{pub_i}||X)) + \sum_{i=1}^{n} V_{pub_i} + \sum_{i=1}^{n} (Y_i * H(M_i||t_i||Y_i))\end{aligned}$$

The proof of the equation is given as shown:

$$\begin{aligned}&\sigma_M * G\\&\quad= \sum_{i=1}^{n} \sigma_{Mi} * G\\&\quad= \sum_{i=1}^{n} (\sigma_{V_{pub_i}} + \beta_i + b_i * H(M_i||t_i||Y_i)) * G\\&\quad= \sum_{i=1}^{n} G_{pub} + \sum_{i=1}^{n} (X * H(V_{pub_i}||X)) + \sum_{i=1}^{n} V_{pub_i} + \sum_{i=1}^{n} (Y_i * H(M_i||t_i||Y_i))\end{aligned}$$

When there are invalid signatures in the batch, the aggregate signatures are verified by using a binary search technique. The signatures are divided into two parts. These two parts are reaggregated and verified. If either one of the two parts fails again, then identical operations performed on the invalid batch repeatedly. This process is repeated until there is only one signature left.

4 Results

The security features satisfied by the proposed method are discussed below along with the performance of the scheme.

4.1 Scheme Security

Integrity Message integrity is ensured in the proposed scheme as the hash of the message is included in the hash used in the signature. If there was a manipulation in the message, $H(M'||\mathrm{t}||Y)$ will not be equal to $H(M||\mathrm{t}||Y)$ therefore the validation of the final signature will fail. Furthermore, a timestamp is included in the hash to prevent replay attacks.

Authentication The proposed method makes sure that vehicles that have not been verified by TA are not been able to deliver messages. This is accomplished by making sure that the vehicle cannot generate its own signature and that the only TA can provide a legitimate signature. Obtaining α from G_{pub} is a difficult task as it cannot be extracted without solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), $G_{pub} = \alpha * G$. The signature validation for a forged signature key by a malicious vehicle will fail as shown:

$$\sigma'_M * G = (\sigma'_{V_{pub}} + \beta + b * H(M||t||Y)) * G$$

where by substituting:

$$\begin{aligned}
\sigma'_M * G & \\
&= ((\alpha' + a' * H(V_{pub}||X')) + \beta + b * H(M||t||Y)) * G \\
&= (\alpha' * G + a' * G * H(V_{pub}||X') + \beta * G + b * G * H(M||t||Y)) \\
&= {G_{pub}}' + X' * H(V_{pub}||X') + V_{pub} + Y * H(M||t||Y)
\end{aligned}$$

Here, ${G_{pub}}'$ from the forged signature is not equal to the group key and therefore validation fails.

$${G_{pub}}' + X' * H(V_{pub}||X') + V_{pub} + Y * H(M||t||Y)$$

$$\neq G_{pub} + X' * H(V_{pub}||X') + V_{pub} + Y * H(M||t||Y)$$

Privacy Privacy is ensured in the proposed scheme by hiding their real-world identities from the outside world. The signature keys are used to identify a vehicle and there is no relation between a vehicle's signature key and its registration information as the signature keys are generated by a random number chosen by the vehicle.

Tracing In the proposed scheme, tracing is possible as the mapping of the vehicle's signatures to their registration information is stored in the TA. For this, the TA can trace the real-world identities of the vehicles suspected of malicious behaviors.

Non-Repudiation The proposed scheme ensures that a vehicle cannot use a forged signature key certification that is generated by the TA. Also, a vehicle cannot steal the signature key of other vehicles. The messages are sent along with σ_M, V_{pub}, X, Y and t. Only the message signer knows the values of $\sigma_{V_{pub}}$, β, and b. Besides the value of β and b cannot be calculated without solving ECDLP. Also $\sigma_{V_{pub}}$ cannot be generated from σ_M. So, the signature validation for a vehicle that tries to generate a signature using a stolen V_{pub} with guessed values of $\sigma_{V_{pub}}$ and β will fail as shown:

$$\begin{aligned}
&\sigma'_M * G\\
&\quad = (\sigma'_{V_{pub}} + \beta' + b' * H(M||t||Y')) * G\\
&\quad = \sigma'_{V_{pub}} * G + \beta' * G + b' * G * H(M||t||Y')\\
&\quad = (\alpha' + a' * H(V_{pub}||X)) * G + {V_{pub}}' + Y' * H(M||t||Y')\\
&\quad = {G_{pub}}' + X' * H(V_{pub}||X) + {V_{pub}}' + Y' * H(M||t||Y')
\end{aligned}$$

And it can be seen that

$$\begin{aligned}
&{G_{pub}}' + X' * H(V_{pub}||X) + {V_{pub}}' + Y' * H(M||t||Y')\\
&\neq G_{pub} + X * H(V_{pub}||X) + V_{pub} + Y' * H(M||t||Y')
\end{aligned}$$

Insider Attack Resistance Insider attacks include the attacks by the theft of crucial material possessed by the TA. The TA is considered fully trustworthy in many schemes and possesses all the private keys of the vehicles in the system. As a result these schemes being highly vulnerable to attacks when the TA is compromised. In the proposed scheme, the TA cannot use the vehicle's signature key certification to generate a valid message signature as it requires the vehicle's private key. This restricts the masquerading attacks even with the assistance of the TA. If a malicious vehicle tries to generate a signature by compromising TA, the signature validation will fail as shown:

$$\sigma'_M = \sigma_{V_{pub}} + \beta' + b * H(M||t||Y)$$

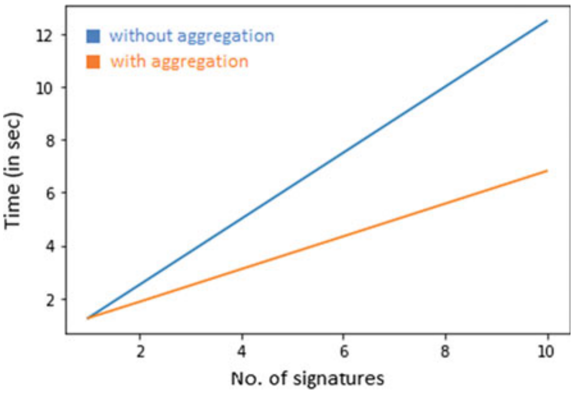
The proof of failure for the signature validation is shown:

$$\begin{aligned}
&\sigma'_M * G\\
&\quad = (\sigma_{V_{pub}} + \beta' + b * H(M||t||Y)) * G\\
&\quad = ((\alpha + a * H(V_{pub}||X)) + \beta' + b * H(M||t||Y)) * G\\
&\quad = (\alpha * G + a * G * H(V_{pub}||X) + \beta' * G + b * G * H(M||t||Y))\\
&\quad = [G_{pub} + X * H(V_{pub}||X) + {V_{pub}}' + Y * H(M||t||Y)
\end{aligned}$$

Validation will fail because ${V_{pub}}' \neq V_{pub}$.

Man in the Middle Attack The proposed scheme ensures that there is no possibility that any other vehicle outside the group can not prove its identity as an authorized member of the group. The scheme can resist man-in-the-middle attacks.
The proof for signature verification will fail as follows.

Fig. 2 Time comparison between scheme with signature aggregation and scheme without signature aggregation



$$\begin{aligned}
&\sigma'_M * G\\
&\qquad = (\sigma_{V_{pub}} + \beta' + b' * H(M'||t'||Y') * G\\
&\qquad = G_{pub}{}' + X' * H(V_{pub}{}'||X') + V_{pub}{}' + Y' * H(M'||t'||Y')
\end{aligned}$$

And it can be seen that

$$\begin{aligned}
&\sigma'_M * G\\
&\qquad \neq G_{pub} + X * H(V_{pub}{}'||X) + V_{pub}{}' + Y' * H(M'||t'||Y')
\end{aligned}$$

4.2 Scheme Performance

The timings were measured on an Intel Core i3-7020U processor at 2.30 GHz with 4.0 GB of RAM using the Windows 10 operating system. The comparison between the performance of the proposed with signature aggregation and the scheme without signature aggregation is shown in Fig. 2.

As can be seen, the change in performance varies by a huge amount as we increase the number of signatures. The graph shows the comparison between the two schemes (with and without signature aggregation) up to ten signatures. When there are ten signatures, the time taken by the scheme without signature aggregation is almost twice the amount of time taken by the scheme with signature aggregation.

5 Conclusion and Future Scope

In this paper, an efficient certificate-less aggregate signature was proposed based on Elliptic Curve Cryptography for Vehicular Ad-hoc Networks without pairings. The proposed scheme reduces the signature verification time and improves the efficiency when multiple signatures are received at the receiver end. This new scheme is analyzed for various security parameters and shows that it satisfies all the security and privacy requirements of the VANET. For future work, more security features can be tested along with various attacks in VANETs.

References

1. Sheikh MS, Liang J, Wang W (2019) A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets). Sensors 19(16):3589. https://doi.org/10.3390/s19163589
2. Parkinson S, Ward P, Wilson K, Miller J (2017) Cyber threats facing autonomous and connected vehicles: future challenges. IEEE Trans Intell Transp Syst 18(11):2998–2915. https://doi.org/10.1109/TITS.2017.2665968
3. Othmane LB, Weffers H, Mohamad MM, Wolf M (2015) A survey of security and privacy in connected vehicles. In: Wireless sensor and mobile ad-hoc networks. Berlin, Germany, Springer, pp 217–247. https://doi.org/10.1007/978-1-4939-2468-4_10
4. Mejri MN, Ben-Othman J, Hamdi M (2014) Survey on vanet security challenges and possible cryptographic solutions. Veh Commun 1(2):53–66. https://doi.org/10.1016/j.vehcom.2014.05.001
5. Xiong W, Tang B (2017) A cloud-based three key management scheme for vanet. In: Proceedings GSKI, Chiang Mai, Thailand, pp 574–587. https://doi.org/10.1007/978-981-13-0896-3_57
6. Gao T, Qi J (2018) An anonymous access authentication scheme for vanets based on id-based group signature. In: Proceedings BWCCA, Taichung, Taiwan, pp 490–497. https://doi.org/10.1007/978-3-030-02613-4_43
7. Li Q, Hsu CF, Raymond Choo KK, He D (2019) A provably secure and lightweight identity-based two-party authenticated key agreement protocol for vehicular ad hoc networks. In: Security in Communication Networks, vol 2019. Art. no. 7871067. https://doi.org/10.1155/2019/7871067
8. Maria A, Pandi V, Lazarus JD, Karuppiah M, Christo (2021) Bbaas: blockchain-based anonymous authentication scheme for providing secure communication in vanets. In: Security in Communication Network, vol 2021. pp 1–11. https://doi.org/10.1155/2021/6679882
9. Liu L, Wang Y, Zhang J, Yang Q (2019) A secure and efficient group key agreement scheme for vanet. Sensors 19(3):482. https://doi.org/10.3390/s19030482
10. Paliwal S, Chandrakar A (2019) A conditional privacy-preserving authentication and multi-party group key establishment scheme for real-time application in vanets. Cryptol ePrint Arch 1–27. [Online]. Available: http://ia.cr/2019/1041
11. Mansour A, Malik KM, Alkaff A, Kanaan H (2021) Alms: asymmetric lightweight centralized group key management protocol for vanets. IEEE Trans Intell Transp Syst 22(3):1663–1678. https://doi.org/10.1109/TITS.2020.2975226
12. Vijayakumar P, Azees M, Kozlov SA, Rodrigues JJ (2021) Anonymous batch authentication and key exchange protocols for 6g enabled vanets. IEEE Trans Intell Transp Syst Early Access. https://doi.org/10.1109/TITS.2021.3099488

13. Funderburg LE, Ren H, Lee IY (2021) Pairing-free signatures with insider-attack resistance for vehicular ad-hoc networks (vanets). IEEE Access 9. https://doi.org/10.1109/ACCESS.2021.3131189
14. Han Y, Song W, Zhou Z, Wang H, Yuan B (2022) Eclas: an efficient pairing-free certificateless aggregate signature for secure vanet communication. IEEE Syst J 16(1)

Comparative Study of LevelDB and BadgerDB Databases on the Basis of Features and Read/Write Operations

Pragya Vaishnav, Linesh Raja, and Aniket Bhange

Abstract Due to advent of huge complex datasets, key-value databases have achieved great demand for accessing data quickly and efficiently in comparison with relational databases. There are several key-value databases available, e.g., Level DB, Badger DB, MongoDB, etc. The data stores as a key—value pair that is so these databases perform all the operations on the dataset rapidly. In this paper the author's purpose is to focus on two most famous key-value databases: Badger DB and Level DB and analyze the performance of both the databases. For this analysis the author created 10 different datasets for every read/write operation. This analysis study is based on the results carried out by instantiate, read, and write operations on these databases and therefore resulting how level DB is more efficient than Badger DB during read and write operations.

Keywords LevelDB · BadgerDB · LSM (log-structured merge tree) · Performance analysis

1 Introduction

In today's world data is important for everyone and for organization it is essential to handle the large amount of data. Basic requirement of the organization is storing and accessing the massive amount of data rapidly. There are many kinds of databases available, such as.

P. Vaishnav (✉) · L. Raja
Department of Computer Applications, Manipal University Jaipur, Jaipur, Rajasthan, India
e-mail: pragya.vaishnav23@gmail.com

A. Bhange
Debit Circle SDN BHD, Kuala Lumpur, Malaysia

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_39

1.1 Centralized Database

Centralized database system uses to store data. Users can fetch the saved data from various locations by using different applications.

- Centralized database size is huge, so the response time for retrieving the data is increased.
- It is tough to modify such a large database system.
- In case the server get failed, we will lose all the data that can be a big loss.

1.2 Distributed Database

In the distributed systems, organization's data is distributed between various database systems. All these database systems are connected through communication channels. Through these channels end-users can retrieve the data efficiently.

- Security and cost are an issue in the distributed system.
- Designing of distributed database is more complex compared with another database.

1.3 Relational Database

The data is stores in the form of rows (tuple) and columns (attributes) in the relational database, and combinedly in forms a table (relation). It is based on relational model.

- It demands large amount of physical memory due to rows and columns.
- Relational database performance depends on the size of tables. If number of tables increase, then it will take more response time to run the queries.

1.4 Cloud Database

The data is stored virtually and executed on the cloud computing platform in the cloud database. To access the database, various cloud computing services are available such as: SaaS, PaaS, IaaS, etc.

1.5 Object-Oriented Databases

In the database system for storing the data this database uses the object-based data model approach. The data is stored and represented as objects.

1.6 Hierarchical Databases

Hierarchical databases store the data in the form of parent–children relationship nodes. Data is arranged in a tree structure format.

1.7 Network Databases

It implements the network data model. Data is represented as nodes which associate through the links.

1.8 NoSQL Database

In this database the data is stored in a wide range. Unlike relational database that stores the data in tabular form, it stores data in various forms. NoSQL database is divided into the four types:

- **Key-value storage**: Every single item is stored as a key (or attribute name) and holds its value in this database together.
- **Document-Oriented Database**: The data is stored in JSON-like document.
- **Graph Databases**: This database stores large size of data in a graph structure.
- **Wide-Column Stores**: It stores data in large columns not in rows.

NoSQL is referred as "Not Only SQL" in the database family. It falls under the category of unstructured databases which include key-value databases, column family databases, and document databases [1]. Due to relational databases' inability to handle the vast amounts of data being transmitted over the Internet and to keep up with emerging technologies like cloud computing, big data, etc., the need for NoSQL surged. Nowadays the primary requirement of database is handling storage and access of large amount of data efficiently and speedily [2]. The LSM tree is the heart of several key-value storage systems with high write throughput, such as DynamoDB/Cassandra, BudgerDB, and LevelDB. The main reason behind it is that LSM implements high write throughput, and every write request is performed only "**in-memory.**"

2 Key-Value Database

The term key-value database is the data storage system that keeps data as a collection of unique identifiers. This data pairing is referred as a "key-value pair." The unique identifier is the "key" points to its associate value, and a value can be a data being identified or pointer of that data. For read and write operation storing key-value can be very fast and very flexible [3]. As data is valuable asset in modern world therefore, we can achieve more data without using traditional structures. And, in key-value storage “null” is not necessary as a place holder for optional value, thus they require less storage and frequently grow virtually and linearly with the number of nodes [4].

2.1 Badger DB

BadgerDB is an embedded, persistent, and fast key-value (KV) database written in Go. It is the underlying database for Dgraph, a fast, distributed graph database [5]. It implements delta encoding to cut down the keys size, and the size of the LSMs also. It stores a fingerprint of the key to save the space instead of key itself store, during read operation [6]. Table 1 presents how unique key identifies the fingerprint of a key and saves the information, but it uses less space.

During accessing data, Badger DB acquires locks on directories, for that multiple processes cannot open the same database at the same time. In case transaction is read-write, the transaction identifies whether there is a conflict or return or error if there is one.

Key-Only Iteration

- Badger stores keys in a lexicographically sorted manner. It exports transaction API, which can set, get, and delete keys. We can iterate over the keys in both forward and reverse order, using iterators [7].
- Figure 1 represents BadgerDB can store a lot of keys into a single SSTable whatever the size of values can be, because only small values or value pointers are stored in LSM tree [8]. Therefore, LSM tree is much smaller than the total amount of data, so we can easily load the tables in RAM (via **`options.LoadToRAM`**, or **`options.MemoryMap`**).

This fast RAM access to the LSM tree allows Badger to provide an option to iterate over the keys very speedily (**`PrefetchValues = false`**), and receive a bunch of unique information about the values without fetching them, such as: value identification bits, size of the value, expiry time, and so on [9].

Table 1 Key-value storage

Key	Value
K1	AAA, 010, XYZ
K2	PQR, PPP
K3	999, &AB
K4	979, ABC, HHH
K5	$123, R5

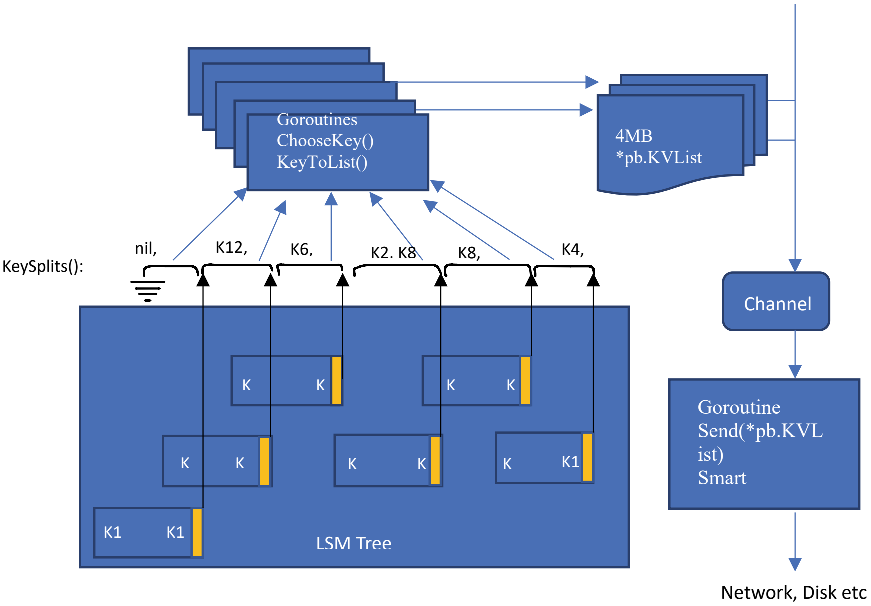


Fig. 1 Architecture of BadgerDB

In Dgraph, **`Item.EstimatedSize()`** is used to calculate the size of certain ranges of data without ever touching the values.

2.2 *LevelDB*

LevelDB is an open-source key-value storage built by Google. It implements an ordered mapping from string keys to string values. The log-structured merge tree (LSM) is the core storage architecture of LevelDB that is a write-optimized B-tree variant. It supports for large sequential writes rather than to small random writes [10]. LevelDB stores keys and values in arbitrary byte arrays [11]. It implements batching writes, forward and backward iteration, and compression of the data.

LevelDB permits only one process to open at a time. To stop concurrent access the operation system implements the locking scheme. LevelDB can be accessed by multiple thread within one process [12].

Figure 2 presents LevelDB immutable data are stored on the disk which is shared by different cluster nodes. There are more than seven levels at most two in-memory tables.

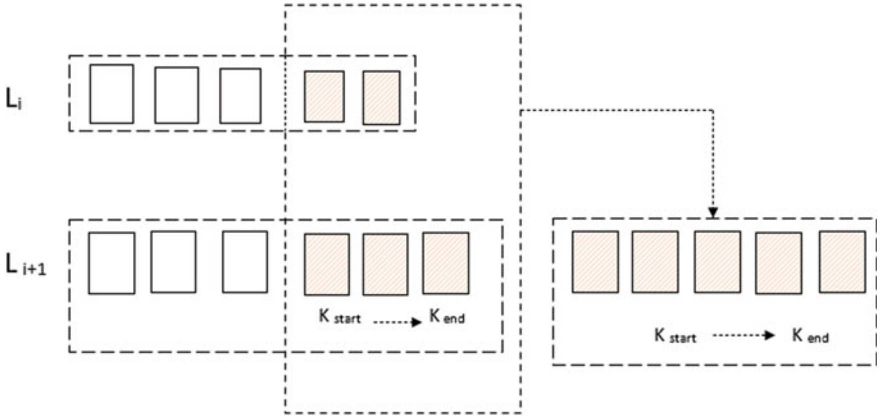


Fig. 2 Compaction procedure in LevelDB

Features

- Key-values are arbitrary byte arrays.
- Using key data is stored in sorted order.
- Callers are able to offer a unique comparison method to change the sort order.
- The main operations are `Put(key,value)`, `Get(key)`, `Delete(key)`.
- Multiple modifications can be done in one atomic batch.
- To get a consistent view of data, users can create a transient snapshot.
- Forward and backward iteration is supported over the data.
- Data is automatically compressed using the Snappy compression library, but Zstd compression is also supported.
- External activity (file system operations, etc.) is relayed via a virtual interface so users can customize the operating system interactions.

Figure 3 represents that there are a total of six levels with the higher level L0 having the latest data and the lower levels L6 storing the old data. Levels exponentially grow in size as we go from higher to lower and obsolete data is removed in a process called compaction.

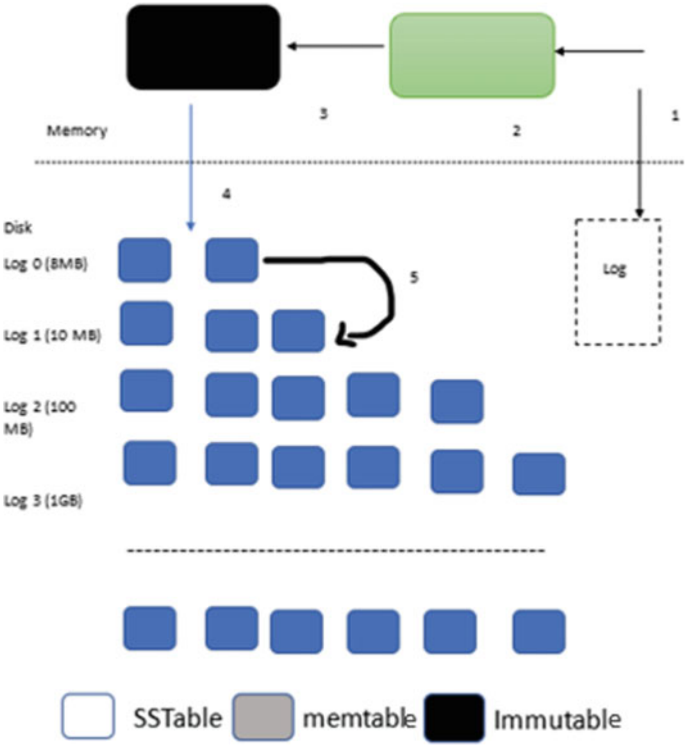


Fig. 3 LevelDB architecture, SSTable layout, and compaction procedure in LevelDB

3 Comparisons of BadgerDB and LevelDB

3.1 Comparison on the Basis of Features

3.1.1 Performance Benefits

BadgerDB provides better performance than LevelDB, both in terms of reads **and** writes. In writes each table in LSM tree can hold a lot more keys, so there are fewer compactions. In reads, the LSM tree in Badger is shallower, so fewer levels have to be queried [13]. Table 2 presents how both the DBs are similar and different to each other.

3.1.2 Design

Design of BadgerDB is inherently more complex than LevelDB, not only because Badger has to deal with an LSM tree, but it also has to maintain a value log [14].

Table 2 Comparison of BadgerDB and LevelDB based on features

Feature	BadgerDB	LevelDB
Design	LSM tree with value log	LSM tree with value log
High read throughput	Yes	Yes
High write throughput	Yes	Yes
Designed for SSDs	Yes (with latest research)	Yes
Embeddable	Yes	Yes
Sorted KV access	Yes	Yes
Pure Go (no Cgo)	Yes	No
Transactions	Yes, ACID, concurrent with SSI	Yes (but non-ACID)
Snapshots	Yes	Yes
TTL support	Yes	Yes
3D access (key-value version) [16]	Yes	No

There's the added complexity of maintaining a value log, doing live value log garbage collection, and moving valid key-value pointers around—all while keeping the LSM tree constraints valid (newer versions at the top, older below) [15].

3.2 Comparison Based on Read and Write Operation

Result of Read Operation

See Table 3

Table 3 Representing read operation timing of BadgerDB and LevelDB

Dataset size	LevelDB timing for read operation (ns)	BadgerDB timing for read operation (μs)
11	1.584	173.333
8	416	6
15	375	1.584
16	375	2.792
5	459	1.125
22	458	1.625
12	417	2.875
17	375	1.625
30	333	1.083
25	292	2.959
35	333	1

Table 4 Representing write operation timing of BadgerDB and LevelDB

Dataset size	LevelDB timing for write operation (μs)	BadgerDB timing for write operation (μs)
11	39.5	50.542
8	6.208	7.875
15	4.125	6.542
16	4.084	4.542
5	3.917	5.209
22	3.875	6.959
12	3.583	5.458
17	7.666	5.041
30	2.875	4.708
25	2.792	4.875
35	2.833	104.25

Result of Write Operation

See Table 4

4 Result Analysis

The author performed read and write operation on both the databases: BadgerDB and LevelDB to check the performance. These operations have been executed in the Golang on the Linux operating system. LevelDB is taking time to perform read operation in nanoseconds, so the author converts the nanoseconds in microseconds and then analysis is done. Tables 3 and 4 represent the timing of read and writing operation on both the databases in microseconds and nanoseconds. We can see timing of LevelDB is taking very less time to perform read and write operation. Therefore, the result is that LevelDB is far faster when compared with BadgerDB. Charts 1 and 2 is representing the read and write operation timings in microseconds of BadgerDB and LevelDB.

Read Operation

See Chart 1.

Write Operation

See Chart 2.

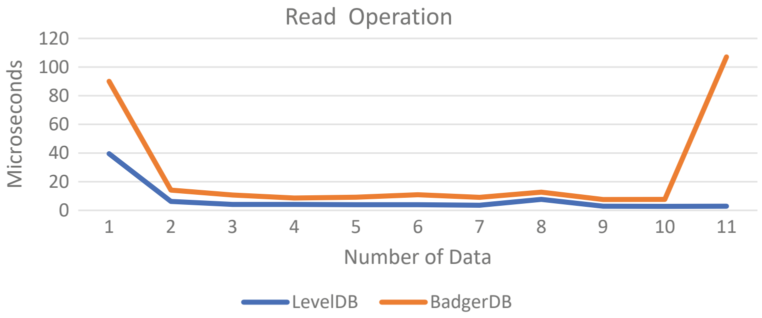


Chart 1 Representing read operation timing differences in microseconds

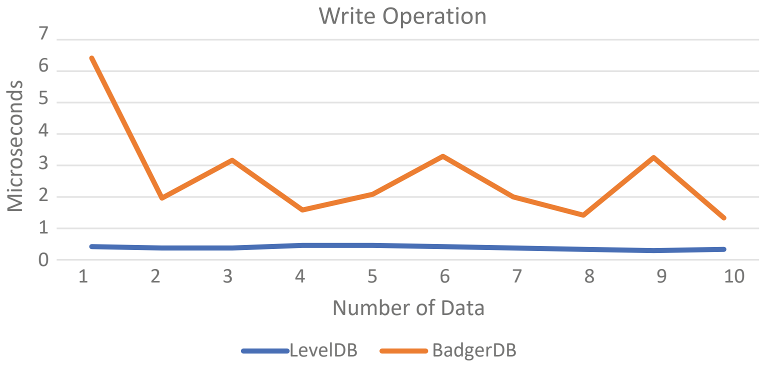


Chart 2 Representing write operation timing differences in microseconds

5 Conclusion

BadgerDB and LevelDB both are LSM tree-based database, which store the data in the key-value format. Therefore, both the databases are capable to perform the reading and writing operation so quickly for huge amount of data. The author compared both the databases: BadgerDB and LevelDB on the basis of features and reading and writing operations. After comparing the author received that LevelDB is much faster and more efficient to perform the read and write operations for large amount of data. The result analysis has been done in Golang on Linux operating system.

References

1. Pokorny J (2013) NoSQL databases: a step to database scalability in web environment. Int J Web Info Syst 9(1):69–82

2. Rai Jain M (2023) Introducing badger: a fast key-value store written purely in Go. Retrieved from https://dgraph.io/blog/post/badger/. Accessed on 09 Nov 2023
3. Gyorodi C, Gyorodi R, Pecherle G, Olah A (2015) A comparative study: MongoDB vs. MySQL. In: 2015 13th international conference on engineering of modern electric systems (EMES)
4. Dgraph, BadgerDB Retrieved from https://dbdb.io/db/badgerdb. Accessed on 09 Nov 2023
5. Herberts M (2023) Demystifying LevelDB. Retrieved from https://blog.senx.io/demystifying-leveldb/. Accessed on 09 Nov 2023
6. Allen M (2015) Relational databases are not designed for scale, relational-databases-scale
7. Wang G, Tang J (2012) The NoSQL principles and basic application of Cassandra model. In: 2012 international conference on computer science and service system
8. Parth T, Nathan S, Viswanathan B (2018) Performance benchmarking and optimizing hyperledger fabric blockchain platform. In: 2018 IEEE 26th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS). IEEE
9. https://github.com/google/leveldb. Accessed on 09 Nov 2023
10. Wood G (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151(2014):1–32
11. An oracle white paper november 2012 oracle NoSQL database
12. Elli A et al (2018) Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference
13. Mattias S (2017) Performance and scalability of blockchain networks and smart contracts
14. Ye G, Liang C (2016)Blockchain application and outlook in the banking industry. Financ Innov 2:1–12
15. Peilin Z et al (2018) A detailed and real-time performance monitoring framework for blockchain systems. In: Proceedings of the 40th international conference on software engineering: software engineering in practice
16. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4:2292–2303

An Improved Snow Ablation Optimizer for Stabilizing the Artificial Neural Network

Pedda Nagyalla Maddaiah and Pournami Pulinthanathu Narayanan

Abstract Artificial neural networks give more promising and accurate results than other methods for prediction, classification, and segmentation engineering problems. The accuracy of the artificial neural network is affected by the training algorithm used. Gradient-based optimization algorithms are traditional methods to train artificial neural networks. They find an accurate solution to the problem. However, they are sensitive to initial values. It makes them unstable for finding better accuracy results. Moreover, training time becomes higher. To overcome these problems, we proposed an improved snow ablation optimizer (ISAO) algorithm and used it to find the pre-trained weights and biases for initializing the artificial neural network's weights and biases. Its performance was tested on the MNIST data set and compared with SGDM-BP, SAO, and GOA algorithms. The improved ISAO algorithm achieved better results than compared algorithms regarding cross-entropy, testing, and training accuracy.

Keywords Numerical optimization · Metaheuristic · Artificial neural network · Snow ablation optimizer

1 Introduction

An artificial neural network (ANN) is more powerful than traditional methods to solve the problems of prediction, segmentation, and classification problems due to its promising results. The ANN easily adapts to the problems of disciplines. One of the major problems of the ANN is the training of it. The ANN training takes massive time due to sensitivity to initial weights and biases and the training algorithm trapping into local minima. To train the ANN, the classical or gradient-based optimization

P. N. Maddaiah (✉) · P. P. Narayanan
Department of Computer Science and Engineering, National Institute of Technology Calicut,
Calicut, Kerala 673601, India
e-mail: pedda_p180075cs@nitc.ac.in

P. P. Narayanan
e-mail: pournamipn@nitc.ac.in

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9_40

algorithms with back propagation (BP) are popularly used [25]. The gradient-based optimization algorithms suffer the following sensitivity to initial values, and at local minima, the gradient vanishes. As a result, the training of the ANN is sensitive to initial weights and biases, and accuracy becomes low, training time becomes high, and makes ANN model unstable at accuracy. There are two ways in the literature to solve above mentioned problems as follows.

- Initialize the ANN's weights and biases by pre-trained weights and biases found using a metaheuristic algorithm.
- Train the ANN using a metaheuristic algorithm instead of a gradient-based algorithm such as SGD, SGDM, or ADAM with back propagation (BP).

Different metaheuristic algorithms have been suggested to initialize the weights and biases of ANN to avoid the sensitivity to the initial weights and biases. Chen et al. [11] to reduce the sensitivity on initial weights, falling into local minima, and slow training of backpropagation (BP), the cuckoo search (CS) algorithm proposed to initializing weights and bias of neural network. Ghanem and Jantan [14] proposed a hybrid algorithm by combining the monarch butterfly optimization (MBO) and artificial bee colony algorithm (ABC) for ANN's weights and biases initialization. The trained ANN detects the intrusion of the network. Wang et al. [24] to improve the pressure measurement accuracy, an improved cuckoo search (CS) algorithm was used to initialize the BP-neural network's weights and biases. Phatai et al. [22] proposed a cultural algorithm for initializing weights and biases of ANN that predicts the Thailand stock exchange (SET) movement. Tanhaeean et al. [23] proposed the boxing match algorithm for initializing the BPNN weights and biases to increase the model's accuracy. Its performance was tested using two function approximations and a forecasting engineering problem.

Diverse metaheuristic algorithms have been suggested to optimize the train or ANN's weights and biases to avoid the sensitivity to the initial weights and biases and trapping into local minima. Bairathi and Gopalani [8] proposed a salp swarm algorithm (SSA) metaheuristic algorithm for updating the feed-forward neural network (FNN) weights and biases. Jalali et al. [16] proposed a butterfly optimization algorithm (BOA) for optimizing the ANN's weights and biases to avoid the local minima. Milosevic et al. [20] proposed a hybrid bat algorithm to optimize the ANN's weights and biases. Agarwal et al. [3] proposed a hybrid Harris hawk whale optimization algorithm to train the ANN. Ang et al. [7] proposed a variant of the teaching-learning-based optimization (TLBO) algorithm for training the ANN to solve the various benchmark problems. Bansal et al.[10] proposed a greedy genetic algorithm for optimizing the weights and biases of multilayer perceptron (MLP). Bangyal et al. [9] proposed an improved PSO algorithm to update the weights and biases of the feed-forward neural network (FNN) for data classification. Abu Doush et al. [2] proposed a variant of coronavirus herd immunity optimizer for training the MLP to classify the data. Khan et al. [18] proposed an improved reptile search algorithm (IRSA) by incorporating the levy flight and sine operator from the sine cosine algorithm to overcome the local minima problem of the reptile search algorithm.

Levy flight with a small step size increases the ability of local search. The sine operator increases the search process diversity to overcome the problem of local minima. The IRSA algorithm optimized the multilayer perceptron and RBF neural network parameters for regression and classification problems. Liu et al. [19] proposed an improved black widow optimization algorithm for RBF neural network parameters training for classification and regression problems. The black widow optimization algorithm uses the nonlinear time-varying factor to enhance the diversification and intensification. Kaya et al. [17] proposed a hybrid metaheuristic algorithm to optimize the weights of DNN to solve gradient-based algorithms' problem of local minima. The PSO algorithm and the human mental search algorithm were used for the hybrid algorithm. The trained DNN was used for predicting sepsis. Ozsoydan et al. [15] proposed an improved arithmetic optimization algorithm by incorporating the local escaping and highly disruptive polynomial mutation operators to train ANN under dynamic environment conditions. The proposed techniques help to overcome the local minima problem of AOA. Abu-Doush et al. [1] proposed archive-based Harris hawks optimizer for optimizing the MLP neural network's weights and biases. The proposed technique saves the current best solutions to utilize in the next iteration. It increases the exploitation of the search process. Ajith and Jolly [5] proposed an African Vulture Updated Honey Badger Optimization (AVUHBO) algorithm for optimizing the weights and biases of proposed hybrid neural networks. It increases the accuracy of the proposed model to detect an object in an image captured by a drone. Alweshah et al. [6] proposed the MBA-SA method to optimize weights and biases of BPNN for software fault prediction. The MBA-SA algorithm comprises the mine blast algorithm and simulated annealing to explore and exploit the search space.

ANN's weights and biases initialization with pre-trained weights and biases of the metaheuristic algorithm helps to avoid the initial weights and biases sensitivity. It increases the accuracy and stability of the ANN model. In this paper, we propose an improved snow ablation optimizer (ISAO) for initializing the ANN's weights and biases. Snow ablation optimizer is a recent metaheuristic algorithm. It is motivated by the natural snow evaporation process. It has the excellent ability to converge global minima for real engineering optimization problems. The snow ablation optimizer has improved by considering a difference calculated between the best solution and Elite information while exploring the search space. It resists the individuals that come near to Elite information. It helps to avoid the local minima and increases the convergence speed.

The remainder of the paper is as follows. The snow ablation optimizer (SAO) is provided in Sect. 2, and Sect. 3 presents an improved snow ablation optimizer (ISAO). Section 4 discusses the method of training ANN with the ISAO algorithm. Results and discussion are provided in Sect. 5. Conclusions and future work are provided in Sect. 6.

2 Snow Ablation Optimizer (SAO)

The snow ablation optimizer (SAO) algorithm introduced by Lingyun Deng and Sanyang Liu [13] inspired by the ablation process of snow in nature. The snow ablation process involves two transform stages snow to liquid water and liquid water to stream by evaporation. At the same time, snow also can become steam. In physics, snow-to-liquid-water transformation is called the melting process, and liquid water to stream transform called sublimation. The equivalent behavior mathematical models of melting and sublimation process developed as exploitation and exploration of the algorithm. Moreover, the dual-population technique was also introduced. In the dual population, the current population has to be randomly split into two subgroups for N_b iterations. The exploitation and exploration of the algorithm are given in the following Eq. (1).

$$Z_i(t+1) = \begin{cases} \text{Elite}(t) + BM_i(t) \otimes (\theta_1 * (G(t) - Z_i(t)) \\ +(1-\theta_1) * (\bar{Z}(t) - Z_i(t))), i \in \text{index}_a, \\ \\ M * G(t) + BM_i(t) \otimes (\theta_2 * (G(t) - Z_i(t)) \\ +(1-\theta_2) * (\bar{Z}(t) - Z_i(t))), i \in \text{index}_b \end{cases} \tag{1}$$

where $\otimes$ is elementwise multiplication operation. t is the iteration, and Elite(t) is the first, second, third, and mean information of the first $N/2$ sorted ordered population. $Elite$ information randomly selected from the first, second, third, and mean of $N/2$ sorted population. $BM_i(t)$ is the random number from the Gaussian distribution, and Brownian motion indicates liquid water to steam process. Equation (2) finds the $BM_i(t)$. $G(t)$ is the best solution so far and $Z_i(t)$ is the current individual solution. $\bar{Z}(t)$ is the Mean of the N population according to the Eq. (3). N is population size. M is the number that denotes the rating of melting of snow according to the Eq. (4). index$_a$ and $index_b$ are the first and second group population indexes, respectively. The equation that belongs to $index_a$ is the exploration equation, and the equation that belongs to index$_b$ is the exploitation equation. An Algorithm 1 shows the dual-population mechanism of the algorithm. An Algorithm 2 shows the algorithm of SAO.

$$f_{BM}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} * \exp(-\frac{x^2}{2}) \tag{2}$$

$$\bar{Z}(t) = \frac{1}{N} * \sum_{i=1}^{N} Z_i(t) \tag{3}$$

$$M = (0.35 + 0.25 * \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1}) * T(t), T(t) = e^{\frac{-t}{t_{max}}} \tag{4}$$

Algorithm 1: Dual-population mechanism

```
Initialize $t = 0, N, t_{\max}, N_a = N/2, N_b = N/2$
while $t \leq t_{max}$ do
    if $N_a < N$ then
        $N_a = N_a + 1$
        $N_b = N_b - 1$
    end
    $t = t + 1$
end
```

Algorithm 2: Snow Ablation Optimizer (SAO)

```
Begin
Initialize $t = 0, N, t_{max}, N_a = N/2, N_b = N/2$
Initialize population $Z_i, i = 1\ to\ N$
Find fitness value of $Z_i, i = 1\ to\ N$
Find the best solution $G(t)$
Find the $Elite(t)$
while $t \leq t_{max}$ do
    Find the melting parameter $M$ from the equation 4
    Split the population into two groups by randomly selecting the individual as $p_a$ and $P_b$
    for Each individual do
        Update population individual position by equation 1.
    end
    Find fitness value of $Z_i, i = 1\ to\ N$
    Find the best solution $G(t)$
    Update the $Elite(t)$
    $t = t + 1$
end
End
Result: Return Best Solution
```

3 Improved Snow Ablation Optimizer (ISAO)

The snow ablation optimizer (SAO) algorithm has improved by considering the difference between *Elite* information and the best solution. The found difference is added to the exploration equation that updates the position of the individuals. This difference helps to avoid the individuals that come very close to the *Elite* position. It avoids trapping into local minima and increases the convergence speed. The Eq. (5) describes the new exploration equation for the update rule of individuals.

$$Z_i(t+1) = \begin{cases} \text{Elite}(t) + rand * (G(t) - \text{Elite}(t)) \\ +BM_i(t) \otimes (\theta_1 * (G(t) - Z_i(t)) \\ +(1-\theta_1) * (\bar{Z}(t) - Z_i(t))), i \in index_a \end{cases} \tag{5}$$

rand $\in$ [0 1] is a random number. In the existing SAO algorithm, the *Elite* information contains the first solution (i.e., best solution), second solution, third solution, and average solution of the first $N/2$ ranked solutions. The $Elite$ information has to be chosen randomly from four solutions at each iteration. It helps to avoid local minima. However, as long as it reaches to max iterations, there is no need to consider the fourth solution average of the first $N/2$ ranked solutions. It makes the search region wider at the end of the iterations. The search region has to narrow down as it reaches to max iterations. It helps the algorithm to converge the best solution. The proposed Eq. (5) can be used according to the Eq. (6).

$$\begin{aligned}
&\text{Elite}(t) = \{X_{\text{First}}, X_{\text{Second}}, X_{\text{Third}}, X_{\text{Average}}\}\\
&\text{Elite}(K) = \{X_{\text{First}}, X_{\text{Second}}, X_{\text{Third}}\}\\
&Z_i(t+1) = \begin{cases} \text{Elite}(t) + rand * (G(t) - \text{Elite}(t)) \\ \quad +BM_i(t) \otimes (\theta_1 * (G(t) - Z_i(t)) \\ +(1-\theta_1) * (\bar{Z}(t) - Z_i(t))), i \in index_a,\ t < t_{\max} * 0.85 \\ \text{Elite}(K) + \text{rand} * (G(t) - \text{Elite}(K)) \\ \quad +BM_i(t) \otimes (\theta_1 * (G(t) - Z_i(t)) \\ +(1-\theta_1) * (\bar{Z}(t) - Z_i(t))), i \in \text{index}_a,\ t > t_{\max} * 0.85 \end{cases}
\end{aligned} \tag{6}$$

where $\{X_{First}, X_{Second}, X_{Third}, X_{Average}\}$ are first, second, third, and average solutions of the first $N/2$ ranked population. t is iteration number, t_{max} is total number of iterations. $Elite(K)$ has to be chosen randomly from the first three solutions $\{X_{First}, X_{Second}, X_{Third}\}$. Algorithm3 shows the algorithm of the proposed ISAO algorithm.

4 ISAO for ANN

Traditional ANN training algorithms are sensitive to initial values of weights and biases. It leads the model to unstable accuracy and high training time. The proposed ISAO algorithm stabilizes the accuracy of the ANN model by initializing the proper values of weights and biases. In order to train ANN with ISAO, we need to define the loss function, problem, and encoding.

4.1 Problem Formulation

ANN's training is an optimization problem. It must find optimum weights and biases to minimize the loss or error between actual and predicted values. It is mathematically defined as follows Eq. (7).

$$\underset{W, b \in \mathbb{R}^D}{\text{Argmin}} \quad f(W, b, X) \tag{7}$$

where f, X, and D are objective function, input feature vector, and dimension of the search space, respectively. W, b are weights and biases, respectively.

4.2 Loss Function

The loss or objective function is essential to build a better ANN model as it directs the weight and biases update to become optimum. This paper uses a categorical cross-entropy loss function to measure the error between actual and predicted values. The loss function is mathematically defined as follows Eq. (8).

$$\begin{aligned}&\text{Categorical cross-entropy:}\\ &f(W, b, X) = -\tfrac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log \hat{y}_{ij} + \eta\, ||W||\end{aligned} \tag{8}$$

where $\hat{y}_{ij}$ is ANN predicted value and y_{ij} is true value. K and$N$ are the number of classes and samples, respectively. The Eq. (9) finds the $\hat{y}_{ij}$. $||W||$ is 2^{nd} norm of W and η is regularization coefficient.

Algorithm 3: Improved Snow Ablation Optimizer (ISAO)

```
Begin
Initialize $t = 0, N, t_{max}, N_a = N/2, N_b = N/2$
Initialize population $Z_i, i = 1\ to\ N$
Find fitness value of $Z_i, i = 1\ to\ N$
Find the best solution $G(t)$
Find the $Elite(t)$
while $t \leq t_{max}$ do
    Find the melting parameter $M$ from the equation 4
    Split the population into two groups by randomly selecting the individual as $p_a$ and $P_b$
    Find indexes $index_a$ of $P_a$ and $index_b$ of $P_b$
    for Each individual of $index_a$ do
        Update population individual position by equation 6
    end
    for Each individual of $index_b$ do
        Update population individual position by equation 1
    end
    Find fitness value of $Z_i, i = 1\ to\ N$
    Find the best solution $G(t)$
    Update the $Elite(t)$
    Update the $Elite(K)$
    $t = t + 1$
end
End
Result: Return Best Solution
```

$$\hat{y}_j^{(l)} = \frac{1}{1+\exp(-(\sum_{i=1}^{n} W_{ij}^{(l)} X_i^{(l-1)} + b_j^{(l)}))} \tag{9}$$

where $\hat{y_j}$ is predicted output at l^{th} layer j^{th} neuron, W_{ij} is weight connection between i^{th} neuron of $(l-1)^{th}$ layer and j^{th} neuron of l^{th} layer. b_j is bias at j^{th} neuron of l^{th} layer. X_i is output feature at i^{th} neuron of $(l-1)^{th}$ layer.

4.3 *Encoding Strategy*

To train an ANN, an individual of the population must be encoded such that the decoding of the individual is easy. There are three ways to encode the individual vector, matrix, and binary representations according to [21, 26]. In this paper, we represented the population individual as a vector. We fixed the number of layers and neurons, so it is easy to decode as weights and biases of layers from the vector.

5 Results and Discussion

To examine the proposed ISAO algorithm's performance in training ANN, we selected the MNIST data set[12] and Stochastic Gradient Descent with Momentum based Back Propagation algorithm (SGDM-BP) [25], Gazelle Optimization Algorithm(GOA) [4], and snow ablation optimizer (SAO) algorithms [13] are selected to compare the ISAO algorithm's performance.

The MNIST data set is a ten-class classification problem. From the MNIST data set, we selected 70 samples randomly for each class to train ANN. Another 30 samples were selected for each class to test a trained ANN model. The structure of ANN is 784-150-10. It means 784 neurons at the input layer. One hundred fifty neurons at the first hidden layer. Ten neurons at the output layer. The experimental setup is Ubuntu 16.04 LTS, 64-bit operating system, 3.8GiB RAM, 3rd generation, 3.20 GHZ, Intel core i5 processor. In the Matlab R2021b version, all algorithms were implemented and tested.

Firstly, we have trained ANN with metaheuristic algorithms ISAO, SAO, and GOA with 20 population sizes and 80 iterations for obtaining the optimum weights and biases. The initial population was selected randomly between -1 and 1. Secondly, the training of the traditional ANN has resumed after the initialization of weights and biases by the obtained optimum weights and biases from the metaheuristic algorithms. Four hundred fifty epochs were used to resume the training of ANN with the SGDM-BP training algorithm. This experiment was repeated 15 times on each metaheuristic algorithm, and the traditional training algorithm SGDM-BP also ran 15 times to train ANN without weights and biases initialization by pre-trained. The cross-entropy and accuracy were used to measure the algorithm's performance. Convergence curves are used to show the algorithm's convergence behavior. Table 1 describes the performance of the compared algorithms. Figure 1 shows the convergence behaviors of the algorithms while training the ANN to get the initial optimum weights and biases.

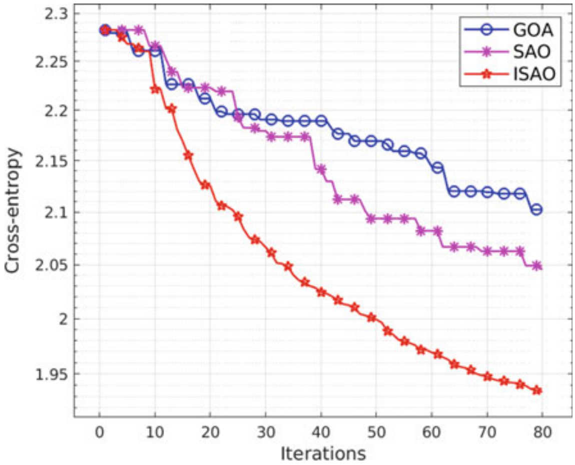
Table 1 Algorithms' ANN training performance comparison on MNIST data set

Methods	Cross-entropy				Accuracy$_{training}$		
	Min.	Mean	Max.	Std.	Min.	Mean	Max.
$SGDM - BP$	1.6006	1.7209	1.9812	0.1079	22.1429	65.9048	86.2857
$GOA - ANN$	1.5405	1.5803	1.6672	0.0324	34.5714	85.6476	93
$SAO - ANN$	1.5447	1.5539	1.5736	0.0082	90	91.9143	93.2857
$ISAO - ANN$	**1.5244**	**1.5396**	**1.5526**	0.0083	**91.5714**	**93.4952**	**95.1429**

Methods	Accuracy$_{testing}$		
	Min.	Mean	Max.
$SGDM - BP$	18	62.3111	79
$GOA - ANN$	32	79.1778	86.3333
$SAO - ANN$	**82.6667**	84.0889	85.6667
$ISAO - ANN$	80.6667	**84.3556**	**86.6667**

From Fig. 1, the convergence curves clearly show that the proposed ISAO algorithm minimizes ANN's prediction error or training error better than other metaheuristic algorithms. The training error has gradually decreased with the increase of iterations by the ISAO training algorithm. The compared algorithms SAO and GOA are trapped in local minima regions, got poor local minima values, and ended with poor local minima values. In contrast, after the 10th iteration, the improved algorithm ISAO started to explore other regions to find a global minimum, got better minimum values toward the iterations, and ended with better local minima. Because the newly

Fig. 1 Metaheuristic algorithms' convergence curves while training ANN on the MNIST dataset



added difference helps to explore new regions around Elite information, it avoids the individuals that are trapped in local minima. At the end of the max iteration, SAO and GOA algorithms training errors were flattened and decreased little. At the same time, the proposed ISAO algorithm's training error gradually decreased with iteration. Because at the end iterations, it is concentrated more on the promising region only by not considering the 4th Elite information.

In Table 1, the bold values show the training algorithm's better performance. It is clear that the proposed ISAO-ANN model got better cross-entropy, training accuracy, and testing accuracy values than SGDM-BP, GOA-ANN, and SAO-ANN models in min. mean, and max. cases. At the same time, SAO-ANN got higher testing accuracy than all compared models in the min. case. However, the ISAO-ANN model got higher average testing accuracy than all compared models. It shows the improvement of the ISAO algorithm over SAO, GAO, and SGDM algorithms. The traditional SGDM-based BP algorithm trained ANN with 22.1429%, 18% minimum training and testing accuracies, and 86.2857%, 79% maximum training and testing accuracies. It shows that the ANN model is unstable with the traditional SGDM-BP training algorithm and trapped in local minima. On the other hand, the ANN model with the improved ISAO and BP training algorithms got promising training accuracy and testing accuracy compared to other training algorithms. The ISAO-ANN model has, in the min. case, 91.5714%, 80.6667%, and in the max. case, 95.1429%, 86.6667% training, and testing accuracies. The ISAO-ANN model is more stable with the improved ISAO training algorithm. The ISAO-ANN is more accurate and stable than other ANN models trained by other algorithms. These results ensure that the proposed ISAO algorithm stabilizes the ANN by initializing the pre-trained weights and biases.

6 Conclusion

We proposed an improved ISAO algorithm to initialize the ANN's weights and biases to overcome the sensitivity towards the initial ANN's weights and biases. The SAO algorithm has improved by adding the difference between the best solution and Elite information while updating the individual positions. An improved ISAO algorithm performance was tested on the MNIST data set. Moreover, the ISAO algorithm's performance was compared with SGDM-BP, SAO, and GOA algorithms. It has achieved better results than SGDM-BP, SAO, and GOA algorithms regarding cross-entropy, training, and testing accuracy. The proposed ISAO-ANN model got higher stabilization than other models.

In the future, it can be hybridized with another metaheuristic algorithm and applied to solve real engineering optimization problems.

References

1. Abu-Doush I, Ahmed B, Awadallah MA, Al-Betar MA, Rababaah AR (2023) Enhancing multilayer perceptron neural network using archive-based harris hawks optimizer to predict gold prices. J King Saud Univ Comput Inf Sci 35(5):101—557 (2023). https://doi.org/10.1016/j.jksuci.2023.101557
2. Abu Doush I, Awadallah MA, Al-Betar MA, Alomari OA, Makhadmeh SN, Abasi AK, Alyasseri ZAA (2023) Archive-based coronavirus herd immunity algorithm for optimizing weights in neural networks. Neural Comput Applic 35(21):15,923–15,941. https://doi.org/10.1007/s00521-023-08577-y
3. Agarwal P, Farooqi N, Gupta A, Mehta S, Khandelwal S (2021) A new harris hawk whale optimization algorithm for enhancing neural networks. In: ACM international conference proceeding series, pp 179–186. DOI 10.1145/3474124.3474149
4. Agushaka JO, Ezugwu AE, Abualigah L (2023) Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. Neural Comput Appl 35(5):4099–4131
5. Ajith VS, Jolly K (2023) G: Hybrid deep learning for object detection in drone imagery: A new metaheuristic based model. Multimed Tools Appl. https://doi.org/10.1007/s11042-023-15785-0
6. Alweshah M, Kassaymeh S, Alkhalaileh S, Almseidin M, Altarawni I (2023) An Efficient Hybrid Mine Blast Algorithm for Tackling Software Fault Prediction Problem. Neural Process Lett. https://doi.org/10.1007/s11063-023-11357-3
7. Ang KM, Lim WH, Tiang SS, Ang CK, Natarajan E, Ahamed Khan MKA (2022) Optimal training of feedforward neural networks using teaching-learning-based optimization with modified learning phases. In: Isa K, Zain ZM, Mohd-Mokhtar R, Mat Noh M, Ismail ZH, Yusof AA, Mohamad Ayob AF, Azhar Ali SS, Abdul Kadir H (eds) Proceedings of the 12th national technical seminar on unmanned system technology 2020, Lecture notes in electrical engineering. Springer, Singapore, pp 867–887 (2022). 10.1007/978-981-16-2406-3_65
8. Bairathi D, Gopalani D (2019) Salp swarm algorithm (SSA) for-training feed-forward neural networks. Adv Intell Syst Comput 816:521–534. https://doi.org/10.1007/978-981-13-1592-3_41
9. Bangyal WH, Nisar K, Soomro TR, Ag Ibrahim AA, Mallah GA, Hassan NU, Rehman NU (2023) An improved particle swarm optimization algorithm for data classification. Appl Sci 13(1):283. https://doi.org/10.3390/app13010283
10. Bansal P, Lamba R, Jain V, Jain T, Shokeen S, Kumar S, Singh PK, Khan, B (2022) GGA-MLP: a greedy genetic algorithm to optimize Weights and biases in multilayer perceptron. contrast media & molecular imaging, (2022) e4036,035 (2022). DOI. https://doi.org/10.1155/2022/4036035
11. Chen YT (2014) Novel back propagation optimization by Cuckoo search algorithm. Sci World J (2014)e878,262. https://doi.org/10.1155/2014/878262
12. Deng L (2012) The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Proc Mag 29(6):141–142
13. Deng L, Liu S (2023) Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. Expert Syst Appl 225:120,069
14. Ghanem WAHM, Jantan A (2020) Training a neural network for cyberattack classification applications using hybridization of an artificial Bee Colony and Monarch Butterfly optimization. Neural Process Lett 51(1):905–946. https://doi.org/10.1007/s11063-019-10120-x
15. Gölcük İ, Ozsoydan FB, Durmaz ED (2023) An improved arithmetic optimization algorithm for training feedforward neural networks under dynamic environments. Knowl-Based Syst 263:110–274 (2023). https://doi.org/10.1016/j.knosys.2023.110274
16. Jalali SMJ, Ahmadian S, Kebria PM, Khosravi A, Lim CP, Nahavandi S (2019) Evolving artificial neural networks using butterfly optimization algorithm for data classification. In: Gedeon T, Wong KW, Lee M (eds) Neural information processing, lecture notes in computer science. Springer International Publishing, Cham, pp 596–607. DOI 10.1007/978-3-030-36708-4_49

17. Kaya U, Yılmaz A, Aşar S (2023) Sepsis prediction by using a hybrid metaheuristic algorithm: a novel approach for optimizing deep neural networks. Diagnostics 13(12):2023. https://doi.org/10.3390/diagnostics13122023
18. Khan MK, Zafar MH, Rashid S, Mansoor M, Moosavi SKR, Sanfilippo F (2023) Improved reptile search optimization algorithm: application on regression and classification problems. Appl Sci 13(2):945. https://doi.org/10.3390/app13020945
19. Liu H, Zhou G, Zhou Y, Huang H, Wei X (2023) An RBF neural network based on improved black widow optimization algorithm for classification and regression problems. Front Neuroinformatics 16
20. Milosevic S, Bezdan T, Zivkovic M, Bacanin N, Strumberger I, Tuba M (2021) Feed-Forward neural network training by hybrid bat algorithm. Commun Comput Inf Sci 1341:52–66. https://doi.org/10.1007/978-3-030-68527-0_4
21. Mirjalili S, Mohd Hashim SZ, Moradian Sardroudi H (2012) Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Appl Math Comput 218(22):11125–11137. https://doi.org/10.1016/j.amc.2012.04.069
22. Phatai G, Chiewchanwattana S, Sunat K (2022) Initialization of smooth adaptive neural network weights with a cultural algorithm for SET index prediction. JIntell Fuzzy Syst 43(4):4987–5000. https://doi.org/10.3233/JIFS-213233
23. Tanhaeean M, Ghaderi SF, Sheikhalishahi M (2023) Optimization of backpropagation neural network models for reliability forecasting using the boxing match algorithm: Electromechanical case. J Comput Des Eng 10(2):918–933. https://doi.org/10.1093/jcde/qwad032
24. Wang H, Zeng Q, Zhang Z, Wang H (2022) Research on temperature compensation of multichannel pressure scanner based on an improved Cuckoo search optimizing a BP neural network. Micromachines 13(8):1351. https://doi.org/10.3390/mi13081351
25. Yu X, Efe MO, Kaynak O (2002) A general backpropagation algorithm for feedforward neural networks learning. IEEE Trans Neural Netw 13(1):251–254
26. Zhang JR, Zhang J, Lok TM, Lyu MR (2007) A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Appl Math Comput 185(2):1026–1037. https://doi.org/10.1016/j.amc.2006.07.025

Author Index

A
Abir, Fazle Rabbi, 435
Agarwal, Aaryan, 461
Ajmera, Pawan K., 461
Alsaffar, Qusay S., 81
Anusuya, S., 45
Ashwin, S., 279
Ayed, Leila Ben, 81

B
Bakliwal, Aryan, 157
Banubakode, A. N., 449
Baruah, Nomi, 485
Bathla, Gaurav, 61
Bawa, Rajesh K., 143
Bhange, Aniket, 513
Biswal, Sumitra, 251
Boora, Ritu, 199
Buttigieg, Sandra, 227

C
Calleja, Neville, 227
Chakkarapani, V., 129
Cicirelli, Franco, 211

D
Dasgupta, Mou, 17
Dash, Sonali, 61
Dcunha, Royce, 173
Deeksha, 373
Dewangan, Harsh, 309
Divya, Yeluri, 71

G
Gaikwad, Snehal, 185
Galdies, Charles, 227
Garg, Lalit, 227
Garg, Rachit, 31
Gowda, Mayur S., 279
Gowtham, A. R. P. S., 423
Gupta, Anmol, 309
Gupta, Anshul, 31

I
Islam, Fariya, 435

J
Jadhav, A. J., 449
Jain, Amita, 295
Jain, Minni, 295
Jangra, Manisha, 199
Jidney, Tasmia Tahmida, 435
Justina, Michael, 359

K
Kabra, Sanchit M., 461
Kalpoma, Kazi A., 435
Kambli, Mansi, 1
Kapdi, Tanvi, 107
Kapse, Prasanna, 387
Katarya, Rahul, 265
Kaur, Kanwarpreet, 61
Kopinski, Thomas, 473
Koul, Apeksha, 143
Kumar, Chava Pavan, 423

S. Das et al. (eds.), *Advances in Data-Driven Computing and Intelligent Systems*,
Lecture Notes in Networks and Systems 892,
https://doi.org/10.1007/978-981-99-9521-9

Kumar, Pravin, 329
Kumar, Vinay, 345
Kumar, Yogesh, 143

L
Lakshmi Prasudha, Mynapati, 71
Lasushe, Kekhelo, 499
Lawrence, Linju, 411
Lhila, Ankur, 461
Lode, Vijaya, 499

M
Maddaiah, Pedda Nagyalla, 525
Maharaju, Rajkumar, 423
Malathi, P., 185
Mall, Anish, 461
Mamatha, H. R., 399
Mangotra, Ishan, 295
Manish, S., 279
Mate, G. S., 449
Meenpal, Toshanlal, 373
Meshram, Tanuj, 17

N
Nagure, Pragnya, 329
Narayanan, Pournami Pulinthanathu, 525
Neog, Mandira, 485
Neubürger, Felix, 473
Nigro, Libero, 211
Nihal, Mohammed, 329
Nikam, Disha Sunil, 399
Nileema, Tajruba Tahsin, 435
Nisha Murthy, D., 399

P
Palkar, Bhakti, 1
Pant, Bhaskar, 119
Panwar, Deepak, 157
Patil, D. H., 449
Pawar, Bhushan, 227
Pinapati, Anil, 499
Poornapushpakala, S., 129
Pragada, Sreeramya Dharani, 399
Prakash, Vijay, 227
Pupo, Francesco, 211

R
Raghuvanshi, Ajay Singh, 309
Raja, Linesh, 513
Ranjan, Nihar, 449
Reddy, Challa Koti, 423
Rodrigues, Aaron, 173
Rodrigues, Cassandra, 173
Rout, Dillip, 387
Roy, Bholanath, 387

S
Saeid, Yasser, 473
Saini, G. L., 157
Saralaya, Sridevi, 329
Sathish, Sriman, 279
Shah, Apurva, 107
Sharma, Geeta, 345
Shehzad, Mohammed, 329
Sheoran, Anshul, 199
Shreelekshmi, R., 411
Shrivastava, Namit, 239
Shukapuri, Nishanth S., 279
Singhal, Nisha, 95
Singh, Ankit Kumar, 309
Singh, Jyoti, 295
Singh, Saurabh, 265
Sonawane, Kavita, 173
Soundarya, M., 45
Srivastava, Manoj K., 239

T
Talasila, Viswanath, 279
Thenmozhi, M., 359
Tripathi, Vikas, 119

V
Vaishnav, Pragya, 513
Valupadasu, Rama, 423
Verma, Ajay, 95
Verma, Poonam, 119
Vidyullatha, Sukhavasi, 71