Lecture Notes in Networks and Systems 1073

Arti Noor · Kriti Saroha · Emil Pricop · Abhijit Sen · Gaurav Trivedi *Editors*

Emerging Trends and Technologies on Intelligent Systems

Proceedings of 4th International Conference ETTIS 2024

Springer

Lecture Notes in Networks and Systems

Volume 1073

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA, School of Electrical and Computer Engineering—FEEC, University of Campinas—UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering, Bogazici University, Istanbul, Türkiye

Derong Liu, Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA
Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering, KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

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, EI Compendex, 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).

Arti Noor · Kriti Saroha · Emil Pricop ·
Abhijit Sen · Gaurav Trivedi
Editors

Emerging Trends and Technologies on Intelligent Systems

Proceedings of 4th International Conference ETTIS 2024

Springer

Editors
Arti Noor
Noida, Uttar Pradesh, India

Emil Pricop
Petroleum-Gas University of Ploiesti
Ploiesti, Prahova, Romania

Gaurav Trivedi
IIT Guwahati
Guwahati, Assam, India

Kriti Saroha
CDAC
Noida, Uttar Pradesh, India

Abhijit Sen
Computer Science and Information Technology
Kwantlen Polytechnic University
Surrey, BC, Canada

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-981-97-5702-2 ISBN 978-981-97-5703-9 (eBook)
https://doi.org/10.1007/978-981-97-5703-9

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

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

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

If disposing of this product, please recycle the paper.

Preface

We are pleased to present the proceedings of the 4th International Conference on *Emerging Trends and Technologies on Intelligent Systems* (ETTIS-2024) held on 27–28 February 2024 at Centre for Development of Advanced Computing (CDAC) Noida in association with the Automatic Control, Computers and Electronics Department, Faculty of Mechanical and Electrical Engineering, **Petroleum-Gas University of Ploiesti, Romania, the University of Haute-Alsace, France,** and **the Northeastern University, Boston.**

As artificial intelligence continues to expand, intelligent systems have become ubiquitous in virtually every area of our lives. Solutions based on intelligent systems are now being implemented across a wide range of fields to address various challenges and improve system productivity. Industries like agriculture, education, smart cities, and healthcare are among those leveraging the advantages of intelligent systems, to mention just a few examples.

ETTIS-2024 marks the fourth edition of the conference series focused on intelligent systems research. The event drew participation from academics, scientists, researchers, and experts across various domains, providing a unified platform to showcase research ideas and exchange insights on cutting-edge advancements in the field. Distinguished experts from different countries delivered invited talks alongside regular paper presentations by authors. The conference received a total of 424 submissions via EasyChair, and after a rigorous double-blind review process by experts, 58 papers were accepted across nine tracks. These papers reflect the latest advancements in various subfields of intelligent systems.

We are highly thankful to our valuable authors for their contribution and our advisory and technical programme committee for their immense support and motivation towards making the ETTIS-2024 a great success. We are also grateful to our keynote speakers and session chairs for chairing the session and giving invited talks: **Prof. Rajkumar Buyya, The University of Melbourne, Australia, Prof Keshab Parhi, University of Minnesota, US, Prof. Pascal Lorenz, University of Haute-Alsace, France, Prof. Rolando Herrero, Northeastern University, Boston, Prof. Abhijit Sen, Kwantlen Polytechnic University, Canada, Prof. Emil Pricop, Petroleum-Gas University of Ploiesti, Romania, Prof. Xin She Yang, Middlesex University,**

London, Prof. Duc Pham, The University of Birmingham, UK, Prof. Gaurav Trivedi, IIT Guwahati, Prof. Arvinder Kaur, GGSIPU Delhi, and **Prof. Vishal Bhatnagar, NSUT Delhi,** for sharing their technical views and enlightening the participants of the conference. CDAC Noida is thankful to MeitY and CSIR for providing support to the conference. CDAC Noida is also thankful to Smt. Sunita Verma, **Scientist 'G' and Group Coordinator CDAC, MeitY, Government of India**. We express our sincere gratitude to our publication partner, LNNS Series, Springer, for support and having faith in us.

Noida, India — Arti Noor
Noida, India — Kriti Saroha
Ploiesti, Romania — Emil Pricop
Surrey, Canada — Abhijit Sen
Guwahati, India — Gaurav Trivedi

Contents

Detection of Cross-Site Scripting Attack Using Bagging Ensemble Learning 1
Liaren Emani Aier, Sreya Bhowmick, Barasha Das, Arpita Nath Boruah, and Mrinal Goswami

A Machine Learning-Based Cross-Layer DoS Attack Detection Technique for IoT 11
Aditi Paul, Shirashti Chaudhary, and Somnath Sinha

Advanced Neural Network MRI Evaluation for Automated Knee Trauma Detection 25
Simran Kumar, Bhavika Salvi, Krina Limbachiya, and Hemanth Varma

Enhancing Cybersecurity Defenses: A Comprehensive Exploration of Applied Artificial Intelligence Strategies 43
Balaram Yadav Kasula and Pawan Whig

A Comprehensive Analysis of Artificial Intelligence Methods to Detect COVID-19 from Chest X-rays and CT Scans 57
Sanjoy Mitra, Parijata Majumdar, Diptendu Bhattacharya, and Srijan Roy

A Blockchain-Based Land Registration System in the Indian Context Incorporating Mortgage and Legal Aspects 73
Yugam Parashar and Vinod Pathari

From Sparsity to Accuracy: Enhancing Recommendation Systems with Item-Based Collaborative Filtering and SVD Matrix Factorization 83
Arkajit Banerjee and Bam Bahadur Sinha

A Comparative Analysis of Sorting Algorithms for Large-Scale Data: Performance Metrics and Language Efficiency 99
Cosmina-Mihaela Roșca and Mădălina Cărbureanu

Genetic Algorithm-Based Probability Distribution Fitting for Improved Classification of Echocardiogram Regions 115
Dennis Thomas, K. M. Subhash, and K. V. Shihabudheen

Investigate the Properties of FTA Through Empirical Experimentation 129
Harsh Raj and Dilshad Shaik

BDSLL: Biomedical Document Recommendation Using Knowledge-Empowered Semantic Intercepted Large Language Models 145
A. Aravind Krishnan and Gerard Deepak

A Review of Digital Twin and the Metaverse from the Standpoint of Smart Factories 155
Florin Zamfir and Emil Pricop

A Comprehensive Patient Triage Algorithm Incorporating ChatGPT API for Symptom-Based Healthcare Decision-Making 167
Cosmina-Mihaela Roșca, Răzvan-Alexandru Bold, and Alexandru-Eduard Gerea

Enhancing Segmentation Accuracy of Diabetic Retinal Images Using Transform Techniques and deepLabV3+ with Ensemble Classification 179
S. Mohana Saranya, T. Pranesh, R. Rajasurya, A. H. Sabareesh, Dinesh Komarasamy, and S. Mohanapriya

Predicting Ocean Currents: A Deep Learning Approach 191
Ameesha Anil and M. Dhanya

Geotagging in Livestock Farming: AI-Based Technologies, Applications, and Challenges 203
Shuchi Verma, Neelotpal Sharma, and Gaurav Trivedi

A Simulation-Based Approach to Evaluate the Effects of Cyberattacks Targeting Transducers in Control Systems 213
Marian Popescu and Emil Pricop

Urban Greens—An Initiative Towards Clean Air 225
Auckloo Pritee, Sanjay Ojha, Anupama Joshi, Himanshi Gupta, Hrithik Pandey, Manisha Bhakuni, and Shikha Bisht

Sign Languages and AI: A Survey of The Impact of Artificial Intelligence on Deafhood 241
Niharika Trivedi, Yashaswini Trivedi, Rushik Parmar, and Prithwijit Guha

Cosine Similarity Preserving Curse of Dimensionality Reduction for Managing Computational Complexity 253
Tefo Kgosietsile

Comprehensive Mathematical Analysis and Derivation of Relationship Between Azimuth and Elevation Angles 267
Shivam Bhutiani, Param, and Gaurav Chugh

Hierarchical Quantum Classification 275
Prashant Gohel and Manjunath Joshi

Emulation of Recommendation System Using QSVT Framework of QiML on CPU 287
Akshay Patil, Abhishek Tiwari, Vivek Khaneja, and Nagendra Singh

A Machine Learning-Based Approach for Soil Chemical Property Estimation Using Multispectral Data in Hungary 301
Sourav Seal and Divyashikha Sethia

Machine Learning-Based Network Attacks Detection 315
Muthukumarapandian Chandrasekaran and Chetana Tailor

Attention-Based Neural Architecture Search for Effective Semantic Segmentation of Satellite Images 325
Avinash Chouhan, Dibyajyoti Chutia, Biswarup Deb, and Shiv Prasad Aggarwal

A Location-Based Application for Recording the Video Stream of Urban Traffic Cameras 337
Karl Aleksander Leoste, Johan Valdemar Leoste, Janika Leoste, and Jose San Martin

A Multiband Sierpinski Arrowhead Fractal Antenna with 3 × 3 CSRR Array for Various Wireless Applications 349
Indraja Lekkala, Likitha Mandyam, Jayanthi Chittipothala, T. Manivannan, E. Sathiyanarayanan, and E. Aravindraj

CKGLD: Curating Knowledge Graph for Linked Open Data Generation for Cyber Law and Ethics 363
A. Aravind Krishnan, Gerard Deepak, A. Santhanavijayan, and K. R. Venugopal

Neuro-ocular Dynamics in Stress Detection: An Integrated Analysis of EEG and EOG Modalities 373
Abhijit Das, Akanksha Subudhi, Priyanka Jain, Rahee Walambe, Ketan Kotecha, N. K. Jain, Rahul Neiwal, and Manoj Jain

Ransomware Detection in Healthcare Organizations Using Supervised Learning Models: Random Forest Technique 385
Rimpa Dutta and Sandip Karmakar

Comparative Assessment of Accessibility and Readability in Generative AI: A Case Study of GPT and Google BARD 397
C. P. Afsal and K. S. Kuppusamy

Fuzzy Logic Based Trust Evaluation and Malicious Node Detection During DoS and DDoS Attack in IoT Networks 409
Himani Tyagi, Rajendra Kumar, and Santosh Kumar Pandey

Classification, Designs, and Different Factors Involved in an Aquaponics System: A Review 427
Neelotpal Sharma, Yashaswini Trivedi, Varist Tripathi, Bhargav Dewri, Hanumant Singh Shekhawat, Rajamoni Neog, and Anamika Yadav

An Improved Medical Entity Extraction Method from Annotated Records 443
Priti Bhardwaj, Nonita Sharma, and Niyati Baliyan

Intonation Analysis and Classification of Declarative Questions and Statements in Nepali Language using Machine Learning Approaches 453
Ranjit Subba, Roshan Kumar Prasad, Pratika Rai, and Sharad Sinha

Optimization of Energy Efficiency in 5G Network Through RAO-I Algorithm 467
Shubham Pandey and Mitul Kumar Ahirwal

Identification of Sounds from Construction Sites: A Machine Learning Approach 479
Sayantan Maji, Himadri Mukherjee, Ankita Dhar, Matteo Marciano, and Kaushik Roy

IoT Meal Master Using Augmented Reality 491
Mayank Chadha, Richa Gupta, Jean Chu, Lakshika Vaishnav, and Aung Chan

PQS-AKA: Post-Quantum Secure Lightweight Authenticated Lattice-Based Key Agreement for IoT Communications 503
Ajeet Singh, Appala Naidu Tentu, and K. Gangadhara Rao

"Transforming Emotions: A Comprehensive Review of Text Emotion Detection with Transformer Models" 515
Brajesh Kumar Khare and Imran Khan

Pose Estimation: Human Keypoint Prediction 535
Madhumati C Mailarada, R. U. Rachanna, Sonal Sheth, Prabha C Nissimagoudar, M. Kaushik, and Nalini Iyer

Disease Classes Identification and Association Rules Extraction Using Medical Records 549
L. R. Aishwarya, V. R. Sai Pooja, and Swetha Rameswaran

A Lightweight Mutual Authentication and Privacy-Preserving Protocol for Secure Message Transmission in Vehicular Ad-Hoc Networks 565
Ram Prakash, Vinod Kumar, and Khushboo Gupta

Siamese BERT with Enhanced Feature Models for Paraphrase Detection 577
M. Mumthaz Beegum and S. Aji

SGPT: An Improved Sentiment Classifier Using Generative Pretrained Transformer 591
Yamuna Prasad, Nitin, and Marc Cahay

Deep Learning-Based Classification of Dietary Restricted Genes 605
Sunidhi Priyadarshini and Shyantani Maiti

A Preliminary Work on Tenyidie Stemming Corpus Creation and Deep Learning Applications 615
Teisovi Angami and Themrichon Tuithung

On Building a Firewall with White Hat Bots in the Neighborhood of Malicious Bots in Large-Scale Networks 625
Ryutaro Matsumoto and Shingo Yamaguchi

Deep Neural Network-Based Yoga Posture Classification Using Human Pose Estimation 639
Rohan Pramod Jadhav and Sunita Dhavale

Next-Gen Driver Safety: An Intelligent System Framework for Multi-Factor Authentication in Distraction Detection 653
Sailee Pramod Shewale

Dermatophytic Dynamics: A Holistic Approach to Superficial Mycosis Research 671
S. Abhishek, T. Anjali, and Rina Barouch Bentov

Deep Learning Models for Enhanced RUL Prediction in Turbofan Jet Engines 685
J. Judeson Antony Kovilpillai, Sulaiman Syed Mohamed, Pragya, Mahmood Hussain Mir, Tinka Singh, and Uday Kumar Singh

Sentiment Analysis of Marathi Texts Using Deep Learning Models 701
Deepak Mane, Sarthak Pithe, Hrishikesh Potnis, Soham Nimale, and Madhur Vaidya

Neuroscientific Study with Hybrid Models Validation for Deception Detection 715
Dharmendra Mishra, Rohit Mishra, Rahee Walambe, Ketan Kotecha, Priyanka Jain, N. K. Jain, Rahul Neiwal, and Manoj Jain

Editors and Contributors

About the Editors

Dr. Arti Noor is presently working as Scientist 'G' at CDAC, Noida. She has done her Ph.D. from NT, BHU, in 1990. She has more than 30 years of R&D experience in the field of VLSI Design and Technology, Hardware Design of Electronics System and Cyber Security. More than 20 years of teaching experience of VLSI design-related courses to M.E. students of BITS, Pilani, and CDAC Noida. She has executed 15 research development projects and 20 commercial projects of various complexities in the area of VLSI Design, Cyber Security, Online examination, and Educational Board Developments. She has guided six Ph.D. and 200 student's projects of B. Tech./M. Tech./M.E. and examined 100 M. Tech. dissertations. She has published 81 research papers in journals and conferences including one monographs. She has one patent, One Transfer of Technology of CDAC FPGA Board in her credit. She is the recipient of ASSOCHEM Women Cyber Influencer 2021 award. She is the life member of Semiconductor Society and Broadcast Engineering Services and an IEEE member.

Dr. Kriti Saroha is presently working as Scientist 'E' at CDAC, Noida. She holds a Ph.D. from GGSIPU, Delhi, and M.S. degree from BITS, Pilani. She has more than 24 years of teaching and research experience. Her research interest includes Data Warehousing and Data Mining, Al and Machine Learning, and Computer Architecture. She is a reviewer member of reputed journals and has participated in technical program committees of international conferences. She has published a book on Computer Organization, Wiley India Private Ltd., and several research papers in national/international conferences and journals.

Emil Pricop is an associate professor and the head of the Automatic Control, Computers, and Electronics Department of the Petroleum-Gas University of Ploiesti, Romania. Dr. Pricop teaches computer networking, software engineering, human-computer interaction, and critical infrastructure protection courses. He received his

Ph.D. in Systems Engineering from the Petroleum-Gas University of Ploiesti, where he defended the thesis "Research regarding the security of control systems" in May 2017. His research interest is in cybersecurity, focusing primarily on industrial control systems (ICS) security. Currently, he is leading a research project aiming at emerging and disruptive technologies that can be used to increase ICS resilience and security. Dr. Emil Pricop is the co-author of Cyber Security: Issues and Current Trends (Springer, 2021) and the editor of four books (three published by Springer Nature and one by MDPI Books). Also, Dr. Pricop is the author or the co-author of 2 national (Romanian) patents, six (6) book chapters published in books edited by Springer, and over 35 papers in journals or international conferences. Since 2013, Dr. Pricop has been the initiator and the chairman of the International Workshop on Systems Safety and Security—IWSSS, an annual scientific event. Dr. Pricop participated in more than 100 technical program committees of prestigious international conferences organized under the auspices of IEEE. He has held the vice-chair position of the IEEE Young Professionals Affinity Group—Romania Section from 2017 to 2019.

Dr. Abhijit Sen is currently a professor of Computing Science and Information Technology at Kwantlen Polytechnic University, BC, Canada. He holds a Ph.D. from McMaster University, Hamilton, Ontario, Canada, and Master of Science degree from University of California, Berkeley, USA, and B. Tech. in Electrical Engineering from Indian Institute of Technology, Kharagpur, India. He has over 30 years of academic and administrative experience having worked in organizations such as Canadian Aviation Electronics, Montreal, Canada, Microtel Pacific Research, Burnaby, Canada. He also worked as a consultant to Canada Post, Montreal, and Info Electronics, Montreal, Canada. He served as a chair of the department for over 14 years. He has also been a visiting professor at Waikato University, Hamilton, New Zealand, and University of Applied Sciences, Munich, Germany, Centre for Development of Advanced Computing (CDAC), India, North China Institute of Aerospace Engineering, China, and Technical University of Applied Sciences, Regensburg, Germany. He has been a keynote speaker in several international conferences and served as the reviewer and technical committee member in number of international conferences. He also served as an external examiner for Ph.D. theses for several universities.

Dr. Gaurav Trivedi holds a Ph.D. and M. Tech. degree from NT, Bombay, and Bachelor of Engineering from Shri Govindram Seksaria Institute of Technology and Science (SGSITS), Devi Ahilya University, Indore, Madhya Pradesh. He has taught wide range of courses in undergraduate and postgraduate courses: Semiconductor Devices and Circuits, Introduction to VLSI Design, Basic Electrical Science, Embedded Systems, Digital IC Design, VLSI System Design, VLSI IC Technology, Algorithms and Data Structures, Introduction to Parallel Computing, and Fundamentals of VLSI CAD. His research interests include: Circuit Simulation (Analog,

RF & Digital) and VLSI CAD, Electronics System Design, Computer Architecture, Semiconductor Devices, Hardware Security, Embedded Systems and loT, High Performance Computing, Large Scale Optimization, and Machine Learning.

Contributors

S. Abhishek Rolls-Royce Inc, Bangalore, India

C. P. Afsal Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India

Shiv Prasad Aggarwal North Eastern Space Applications Centre, Umiam, Meghalaya, India

Mitul Kumar Ahirwal Maulana Azad National Institute of Technology, Bhopal, India

Liaren Emani Aier Programme of Computer Science and Engineering, Faculty of Engineering, Assam down town University, Gandhi Nagar, Guwahati, Assam, India

L. R. Aishwarya Pfizer Healthcare India Private Limited, Chennai, India

S. Aji Department of Computer Science, University of Kerala, Thiruvananthapuram, Kerala, India

Teisovi Angami Department of Computer Science & Engineering, National Institute of Technology Nagaland, Dimapur, India;
Department of Information Technology, School of Engineering & Technology, Nagaland University, Lumami, India

Ameesha Anil Amrita Vishwa Vidyapeetham, Amritapuri, India

T. Anjali Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India

A. Aravind Krishnan Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India;
Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru, India

E. Aravindraj Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India

Niyati Baliyan CE Department, NITKKR, Kurukshetra, Haryana, India

Arkajit Banerjee Department of Statistics, Acharya Prafulla Chandra College, New Barrackpore, Kolkata, India

Rina Barouch Bentov Department of Comparative Medicine, Stanford University School of Medicine, Stanford, CA, USA

Manisha Bhakuni GGSIPU, Delhi, India

Priti Bhardwaj IT Department, IGDTUW, New Delhi, India

Diptendu Bhattacharya National Institute of Technology, Agartala, India

Sreya Bhowmick Programme of Computer Science and Engineering, Faculty of Engineering, Assam down town University, Gandhi Nagar, Guwahati, Assam, India

Shivam Bhutiani SRM University—AP, Guntur, India

Shikha Bisht GGSIPU, Delhi, India

Răzvan-Alexandru Bold Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, Ploieşti, Romania

Arpita Nath Boruah Programme of Computer Science and Engineering, Faculty of Engineering, Assam down town University, Gandhi Nagar, Guwahati, Assam, India

Marc Cahay University of Cincinnati, Cincinnati, USA

Mădălina Cărbureanu Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, Ploieşti, Romania

Mayank Chadha Northeastern University, Boston, MA, USA

Aung Chan Strategy First University, Yangon, Myanmar

Muthukumarapandian Chandrasekaran Ascendion Inc., Bernards, NJ, USA

Shirashti Chaudhary Banasthali Vidyapith, Tonk, Rajasthan, India

Jayanthi Chittipothala Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India

Avinash Chouhan North Eastern Space Applications Centre, Umiam, Meghalaya, India

Jean Chu Carnegie Mellon University, Pittsburg, PA, USA

Gaurav Chugh Space Applications Centre, Indian Space Research Organization, Ahmedabad, India

Dibyajyoti Chutia North Eastern Space Applications Centre, Umiam, Meghalaya, India

Abhijit Das C-DAC, Delhi, India

Barasha Das Programme of Computer Science and Engineering, Faculty of Engineering, Assam down town University, Gandhi Nagar, Guwahati, Assam, India

Biswarup Deb North Eastern Space Applications Centre, Umiam, Meghalaya, India

Gerard Deepak Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru, India

Bhargav Dewri Kendriya Vidyalaya IIT Guwahati, Guwahati, India

M. Dhanya Amrita Vishwa Vidyapeetham, Amritapuri, India

Ankita Dhar Department of Computer Science and Engineering, Sister Nivedita University, Kolkata, India

Sunita Dhavale School of Computer Engineering and Mathematical Sciences, Defence Institute of Advanced Technology, Pune, Maharashtra, India

Rimpa Dutta Department of Computer Science and Engineering, National Institute of Technology Durgapur, Durgapur, West Bengal, India

Alexandru-Eduard Gerea Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, Ploieşti, Romania

Prashant Gohel HCL Technologies, Bengaluru, Karnataka, India

Mrinal Goswami Programme of Computer Science and Engineering, Faculty of Engineering, Assam down town University, Gandhi Nagar, Guwahati, Assam, India

Prithwijit Guha Indian Institute of Technology, Guwahati, Assam, India

Himanshi Gupta GGSIPU, Delhi, India

Khushboo Gupta Department of Electronics and Communication, University of Allahabad, Prayagraj, India

Richa Gupta Massachusetts Institute of Technology, Cambridge, MA, USA

Nalini Iyer KLE Technological University, Hubballi, India

Rohan Pramod Jadhav School of Computer Engineering and Mathematical Sciences, Defence Institute of Advanced Technology, Pune, Maharashtra, India

Manoj Jain MeitY, Delhi, India

N. K. Jain C-DAC, Delhi, India

Priyanka Jain C-DAC, Delhi, India

Anupama Joshi GGSIPU, Delhi, India

Manjunath Joshi Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India

J. Judeson Antony Kovilpillai Alliance University, Bangalore, India

Sandip Karmakar Department of Computer Science and Engineering, National Institute of Technology Durgapur, Durgapur, West Bengal, India

Balaram Yadav Kasula Department of Information Technology, University of The Cumberlands, Williamsburg, KY, USA

M. Kaushik KLE Technological University, Hubballi, India

Tefo Kgosietsile University of Botswana, Gaborone, Botswana

Imran Khan Department of Computer Science and Engineering, Harcourt Butler Technical University, Kanpur, Uttar Pradesh, India

Vivek Khaneja Embedded Systems R&D, Centre for Development of Advanced Computing, Noida, India

Brajesh Kumar Khare Department of Computer Science and Engineering, Harcourt Butler Technical University, Kanpur, Uttar Pradesh, India

Dinesh Komarasamy Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India

Ketan Kotecha SIT Pune, Mulshi, India

Rajendra Kumar Department of Computer Science, Jamia Millia Islamia, New Delhi, India

Simran Kumar Department of Mathematics and Computing, National Institute of Technology, Hamirpur, India

Vinod Kumar Department of Electronics and Communication, University of Allahabad, Prayagraj, India

K. S. Kuppusamy Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India

Indraja Lekkala Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India

Janika Leoste Tallinn University of Technology, Tallinn, Estonia

Johan Valdemar Leoste Tallinn University of Technology, Tallinn, Estonia

Karl Aleksander Leoste Tallinn University, Tallinn, Estonia

Krina Limbachiya Malaviya National Institute of Technology, Jaipur, India

Madhumati C Mailarada KLE Technological University, Hubballi, India

Shyantani Maiti Department of CSE, National Institute of Technology, Durgapur, WB, India

Sayantan Maji Department of Computer Science and Engineering, Sister Nivedita University, Kolkata, India

Parijata Majumdar Indian Institute of Information Technology, Agartala, Agartala, India

Likitha Mandyam Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India

Deepak Mane Vishwakarma Institute of Technology, Pune, India

T. Manivannan Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India

Matteo Marciano Gazelien Records Lab, Department of Arts and Humanities, Music Program, New York University Abu Dhabi, Abu Dhabi, UAE

Jose San Martin Universidad Rey Juan Carlos, Madrid, Spain

Ryutaro Matsumoto Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube-shi, Japan

Mahmood Hussain Mir Alliance University, Bangalore, India

Dharmendra Mishra C-DAC, Delhi, India

Rohit Mishra C-DAC, Delhi, India

Sanjoy Mitra Tripura Institute of Technology, Agartala, India

Sulaiman Syed Mohamed Alliance University, Bangalore, India

S. Mohanapriya Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India

Himadri Mukherjee TISA Lab, Department of Computer Science, West Bengal State University, Kolkata, India

M. Mumthaz Beegum Department of Computer Science, University of Kerala, Thiruvananthapuram, Kerala, India

Rahul Neiwal MeitY, Delhi, India

Rajamoni Neog Vellore Institute of Technology, Vellore, India

Soham Nimale Vishwakarma Institute of Technology, Pune, India

Prabha C Nissimagoudar KLE Technological University, Hubballi, India

Nitin University of Cincinnati, Cincinnati, USA

Sanjay Ojha Centre for Development of Advanced Computing (C-DAC), Noida, India

Hrithik Pandey GGSIPU, Delhi, India

Santosh Kumar Pandey Department of Electronics and Information Technology, Ministry of Electronics and Information Technology, New Delhi, India

Shubham Pandey Maulana Azad National Institute of Technology, Bhopal, India

Param Space Applications Centre, Indian Space Research Organization, Ahmedabad, India

Yugam Parashar Department of Computer Science and Engineering, NIT Calicut, Kozhikode, India

Rushik Parmar Indian Institute of Technology, Guwahati, Assam, India

Vinod Pathari Department of Computer Science and Engineering, NIT Calicut, Kozhikode, India

Akshay Patil Embedded Systems R&D, Centre for Development of Advanced Computing, Noida, India

Aditi Paul Banasthali Vidyapith, Tonk, Rajasthan, India

Sarthak Pithe Vishwakarma Institute of Technology, Pune, India

Marian Popescu Automatic Control, Computers and Electronics Department, Petroleum-Gas University of Ploiesti, Ploiesti, Romania

Hrishikesh Potnis Vishwakarma Institute of Technology, Pune, India

Pragya Alliance University, Bangalore, India

Ram Prakash Department of Electronics and Communication, University of Allahabad, Prayagraj, India

T. Pranesh Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India

Roshan Kumar Prasad Department of Computer Science and Technology, University of North Bengal, Raja Rammohunpur, Darjeeling, West Bengal, India

Yamuna Prasad Indian Institute of Technology Jammu, Jammu, India

Emil Pricop Automatic Control, Computers and Electronics Department, Petroleum-Gas University of Ploiesti, Ploiesti, RO, Romania

Auckloo Pritee Mauritius Institute of Education, Reduit, Mauritius

Sunidhi Priyadarshini Department of CSE, National Institute of Technology, Durgapur, WB, India

R. U. Rachanna KLE Technological University, Hubballi, India

Pratika Rai Department of Computer Science and Technology, University of North Bengal, Raja Rammohunpur, Darjeeling, West Bengal, India

Harsh Raj SRM University, Chennai, India

R. Rajasurya Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India

Swetha Rameswaran Pfizer Healthcare India Private Limited, Chennai, India

K. Gangadhara Rao Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India

Cosmina-Mihaela Roşca Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, Ploieşti, Romania

Kaushik Roy TISA Lab, Dept. of Computer Science, West Bengal State University, Kolkata, India

Srijan Roy National Institute of Technology, Agartala, India

A. H. Sabareesh Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India

V. R. Sai Pooja Pfizer Healthcare India Private Limited, Chennai, India

Bhavika Salvi Fr Conceicao Rodrigues College of Engineering, Mumbai, India

A. Santhanavijayan National Institute of Technology, Tiruchirappalli, India

S. Mohana Saranya Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India

E. Sathiyanarayanan Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India

Sourav Seal Department of Software Engineering, Delhi Technological University, Delhi, India

Divyashikha Sethia Department of Software Engineering, Delhi Technological University, Delhi, India

Dilshad Shaik IIM Sambalpur, Hyderabad, India

Neelotpal Sharma IIT Guwahati, Guwahati, India

Nonita Sharma IT Department, IGDTUW, New Delhi, India

Hanumant Singh Shekhawat IIT Guwahati, Guwahati, India

Sonal Sheth KLE Technological University, Hubballi, India

Sailee Pramod Shewale Symbiosis Institute of Design (SID), Symbiosis International (Deemed University) (SIU), Pune, Maharashtra, India

K. V. Shihabudheen Department of Electrical Engineering, National Institute of Technology, Calicut, Kerala, India

Ajeet Singh Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India;
C.R. Rao Advanced Institute of Mathematics, Statistics, and Computer Science, University of Hyderabad Campus, Hyderabad, Telangana, India

Nagendra Singh Embedded Systems R&D, Centre for Development of Advanced Computing, Noida, India

Tinka Singh Alliance University, Bangalore, India

Uday Kumar Singh Alliance University, Bangalore, India

Bam Bahadur Sinha Computer Science and Engineering, National Institute of Technology Sikkim, Ravangla, India

Sharad Sinha Department of Computer Science and Technology, University of North Bengal, Raja Rammohunpur, Darjeeling, West Bengal, India

Somnath Sinha CHRIST (Deemed to be University), Bengaluru, India

Ranjit Subba Department of Computer Science, Hijli College Kharagpur, Kharagpur, West Bengal, India

K. M. Subhash Department of Electrical Engineering, National Institute of Technology, Calicut, Kerala, India

Akanksha Subudhi C-DAC, Delhi, India

Chetana Tailor Ascendion Engineering Private Limited, Bangalore, India

Appala Naidu Tentu C.R. Rao Advanced Institute of Mathematics, Statistics, and Computer Science, University of Hyderabad Campus, Hyderabad, Telangana, India

Dennis Thomas Department of Electrical Engineering, National Institute of Technology, Calicut, Kerala, India

Abhishek Tiwari Embedded Systems R&D, Centre for Development of Advanced Computing, Noida, India

Varist Tripathi Kendriya Vidyalaya IIT Guwahati, Guwahati, India

Gaurav Trivedi IIT Guwahati, Guwahati, India

Niharika Trivedi UC San Diego, San Diego, CA, USA

Yashaswini Trivedi Disha Delhi Public School in Kota, Kota, Rajasthan, India

Themrichon Tuithung Department of Computer Science and Engineering, National Institute of Technology Nagaland, Dimapur, India

Himani Tyagi Department of Computer Science, Jamia Millia Islamia, New Delhi, India

Madhur Vaidya Vishwakarma Institute of Technology, Pune, India

Lakshika Vaishnav State University of New York, Albany, NY, USA

Hemanth Varma Indian Institute of Information Technology, Bhagalpur, India

K. R. Venugopal Bangalore University, Bengaluru, India

Shuchi Verma Ramjas College, Delhi University, Delhi, India

Rahee Walambe SIT Pune, Mulshi, India

Pawan Whig Threws, VIPS-TC, New Delhi, India

Anamika Yadav IIT Guwahati, Guwahati, India

Shingo Yamaguchi Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube-shi, Japan

Florin Zamfir Petroleum-Gas University of Ploiesti, Ploiesti, RO, Romania

Detection of Cross-Site Scripting Attack Using Bagging Ensemble Learning

Liaren Emani Aier, Sreya Bhowmick, Barasha Das, Arpita Nath Boruah, and Mrinal Goswami

Abstract In today's digital age, the Internet is pivotal in bringing both convenience and security concerns. Individuals are vulnerable to unauthorized access and manipulation of their information by potential attackers. A notable cybersecurity threat revolves around injections, a method for introducing malicious code into web applications, creating vulnerabilities that attackers can exploit. Within the scope of this study, our attention has been directed toward a specific form of injection attack known as cross-site scripting (XSS). Following thorough research, our strategic direction involves adopting a bagging ensemble learning method for detecting XSS anomalies. The random forest classifier was employed as the chosen model in this approach. Considering all the essential comparison parameters, the proposed simulation results significantly improve the existing work.

Keywords Cross-site scripting (XSS) · Ensemble learning · Random forest (RF)

1 Introduction

Web applications constitute fundamental components of our daily activities, transforming how we communicate, retrieve information, and engage in different activities. As our dependency on web applications grows, the need to swiftly tackle the crucial matter of web security intensifies. The ever-changing dynamics of web applications and the escalating complexity of cyber threats present considerable hurdles in guaranteeing the confidentiality, integrity, and availability of sensitive data.

The cross-site scripting (XSS) attack is a frequent threat, as reported in the OWASP Top Ten ranking. It stood in third place for critical vulnerabilities of web applications [1]. XSS is a form of injection attack wherein a malicious, typically harmful

L. E. Aier · S. Bhowmick · B. Das · A. N. Boruah · M. Goswami (✉)
Programme of Computer Science and Engineering, Faculty of Engineering, Assam down town University, Sankar Madhab Path, Panikhaiti, Gandhi Nagar, Guwahati, Assam, India
e-mail: mrinal.g@adtu.in

A. N. Boruah
e-mail: arpita.b@adtu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_1

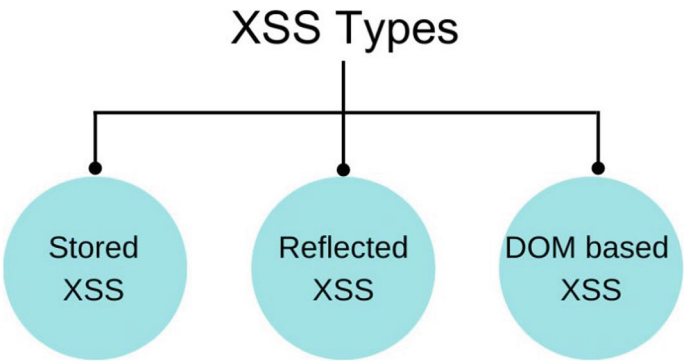


Fig. 1 Types of XSS

script is injected into content from trusted websites. Widely acknowledged as one of the most prevalent cyber-attack types, XSS commonly employs bits of Javascript code to deliver malicious scripts. Its exploits can extend to incorporating malicious executable code encompassing various languages such as Java, Ajax, and Hypertext Markup Language (HTML) [2].

XSS attacks can be classified into three, i.e., (i) Stored XSS, (ii) Reflected XSS, and (iii) DOM-based XSS, as shown in Fig. 1.

(i) Stored XSS (Persistent XSS)

Stored XSS involves injecting a malicious script that gets stored on the server, e.g., in a database, and is later put on view to users visiting the affected page [3]. When an attacker submits a malicious script via a comment form, the script gets stored in the application database. When other users view the comment section, the malicious script executes in their browsers, potentially compromising their sensitive data [4].

(ii) Reflected XSS

Reflected XSS happens when a harmful script is inserted into a URL. It is then returned to the user by the web application; typically, they are seen in error messages or search outcomes [3]. When an attacker creates a malicious URL by adding a script within a parameter, upon the user's click, the altered link displays the script in the page content, causing it to act within the user's browser [5].

(iii) DOM-Based XSS

DOM-based XSS occurs due to inadequate handling of user input by client-side scripts in the web application. This leads to alterations in the Document Object Model (DOM) that enable the execution of the malicious script [3]. When an attacker modifies the client-side JavaScript to trigger the script by tampering with the DOM through inputcontrolled by the user, it leads to the injected script's execution [6].

Today, deploying cybersecurity technology effectively is inconceivable without a substantial reliance on machine learning. Through machine learning, cybersecurity

systems gain the capability to analyze patterns, learning from them to proactively prevent comparable attacks and adapt to evolving behaviors.

Bagging is a powerful ensemble learning technique employed in machine learning to enhance algorithms' stability and accuracy. It is a homogeneous learning method as it uses the same base algorithm across all the individual models within the ensemble. This method involves training instances of the same algorithm on different subsets of the training data and then combining their predictions. The models are trained in parallel, enhancing computational efficiency. Random Forest is often preferred for bagging due to its ability to maintain homogeneity. This facilitates the combination of predictions, effectively allowing Random Forest to achieve diversity and accuracy within the ensemble.

The Random Forest algorithm is an exemplary instance of a machine learning algorithm, particularly in our application. The Random Forest technique is a versatile and powerful ensemble learning method that operates by constructing a significant/large number of decision trees during training and providing accurate outputs. One of the critical characteristics of the Random Forest classifier is its homogeneity. This homogeneity enhances the algorithm's resilience to overfitting and allows for parallelization, making it computationally efficient and well-suited for large datasets.

2 Related Work

In [7], a novel hybrid stacking ensemble technique, dictionary-based mapping, and URL encoding were proposed to combat XSS attacks. The detection process integrates four distinct datasets and employs DNN models (RNN, CNN, LSTM) and ML models (NB, SVM, k-NN). This approach exhibits high effectiveness, achieving over 99% accuracy.

Four separate datasets are utilized, each comprising safe and unsafe content. The hybrid detection technique dynamically identifies attacks within URLs. The defense mechanism combines URL encoding with dictionary-based mapping to enhance prediction accuracy, expedite training, and eliminate unsafe JScript/JavaScript keywords from URLs. Simulation outcomes demonstrate the superiority of the proposed hybrid model over existing detection methods, achieving over 99.5% accuracy in classifying XSS attacks (accuracy, precision, recall, f1_score, and ROC). Furthermore, the hybrid approach exhibits strong resilience against XSS attacks and is validated in a real-time environment to safeguard server information.

Seethalakshmi Perumal et al. investigate how ensemble-based machine learning might be helpful to enhance cybersecurity, mainly when spotting XSS attacks on web applications [8]. Their model employs stacking ensembles and multiple classification algorithms to anticipate precisely the URLs that would be subject to XSS attacks.

It is an unsupervised model detecting cross-site scripting payloads. Here, training of the dataset was done in a 70:30 ratio, with the dataset employed consisting of a collection of cross-site scripting (XSS) payloads from GitHub, where they added

safe links themselves. The paper does not specify the list of safe URLs they used. Also, the dataset is not balanced.

In [9], a method that combines various XSS data sources from Kaggle and GitHub is presented to form a balanced dataset. They are using the bagging method and the Random Forest algorithm. The results were 97.16% in accuracy, 96% in precision, and 97% in recall. Her model demonstrated superior performance when pitted against Gradient Boosting and Decision Tree Bagging.

The paper proposes two random forest methods for imbalanced data classification—Balanced Random Forest (BRF) and Weighted Random Forest (WRF). It evaluates their performance on six datasets, emphasizing metrics like G-mean and F-measure. BRF and WRF demonstrate superior results compared to existing methods, addressing the challenge of minority class prediction. The study concludes by highlighting the computational efficiency of BRF and the potential vulnerability of WRF to label noise, suggesting their effectiveness in handling imbalance.

3 Methodology

In this section, we dive into a comprehensive discussion of our proposed approach, focusing on the detection of XSS vulnerabilities through the application of machine learning, with a specific emphasis on a bagging algorithm. Our overall action plan is visually represented and briefly explained in Fig. 2.

3.1 Data Collection

For any model, the primary step is to acquire the required dataset, which we obtained from Kaggle [10]. Having approximately 13,000 samples, the dataset included URLs

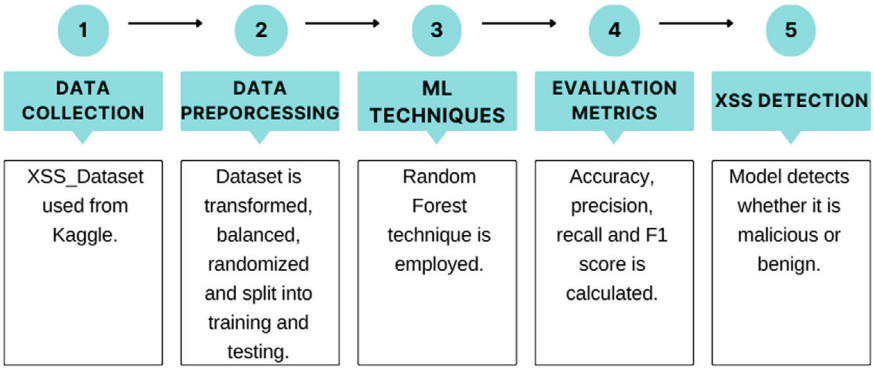


Fig. 2 Proposed model

with malicious XSS payloads and benign links. Each URL is labeled as 0 and 1, where 1 represents malicious content, and 0 represents benign content [11].

3.2 Pre-processing

Our proposed model utilized fifty percent of the 13,000 samples extracted precisely so that the end dataset would be balanced and then randomized. This prevents bias toward majority classes and facilitates accurate evaluation metrics by mitigating the influence of class imbalance. It then goes through the transformation process to enhance the algorithm's data comprehension. Natural language processing (NLP) activities are implemented for stop word removal, a critical component of data transformation that improves the effectiveness and quality of text analysis. Stopwords are common words with minimal to no substantial value, such as "and," "the," and "is". By eliminating them, we improve the focus on essential terms and phrases while lowering noise in the data. Giving NLP models improves model interpretability and eventually raises the accuracy of the models. For the final stage of our preprocessing, we partitioned the dataset using the standard 80:20 ratio, assigning 80% to training, and setting aside 20% for testing.

3.3 Classifier

The Random Forest classification conforms to the concept of crowd theory, which states that knowledge of a crowd is better than that of an individual. Random Forest functions by constructing multiple decision trees in forest-like structures during training and combining their outputs to determine the final output. Each decision tree in the forest is built independently, utilizing a random subset of the data, and contributes equally to the final prediction. Furthermore, we chose this classifier because Random Forest algorithms can manage or handle nonlinear interactions among variables. They can be productively used on large datasets where traditional methods fail due to computational limitations.

3.4 Bagging

We employ the Bagging ensemble machine learning algorithm to our prepossessed data in the platform Jupiter. Due to its adherence to homogeneity, it functions by constructing several models using the same algorithm. In the context of our paper, the Random Forest classifier is applied to our training set, which is used to build multiple models based on the RF algorithm. Each model is then trained using the bagging ensemble method to generate more robust and accurate predictions collectively. This

process involves training various models on separate batches of the data, allowing each model to contribute its prediction to the final collected result. The outcomes are subsequently used to compute the evaluation metrics, which include accuracy, precision, recall, and the F1 score.

4 Results and Discussion

The evaluation metrics are derived from the confusion matrix, a fundamental tool in assessing model performance by summarizing the counts of true positive, true negative, false positive, and false negative predictions. It helps visualize the model's effectiveness in classification tasks by displaying the relationship between predicted and true classes. The confusion matrix is illustrated in Fig. 3, and the evaluation metrics mathematical expressions, which include Accuracy, Precision, Recall, and F1 Score, are given in Eqs. 1, 2, 3, and 4, respectively, as follows:

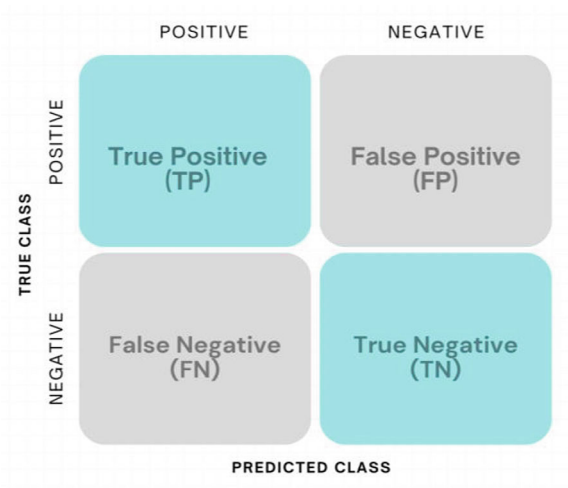
Evaluation Metrics:

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

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

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

Fig. 3 Confusion matrix



$$\text{F1 score} = 2 \times \frac{\text{precision} - \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

The final results of our proposed model, which comprises the evaluation metrics and the confusion matrix, are illustrated in Figs. 4 and 5 as follows:

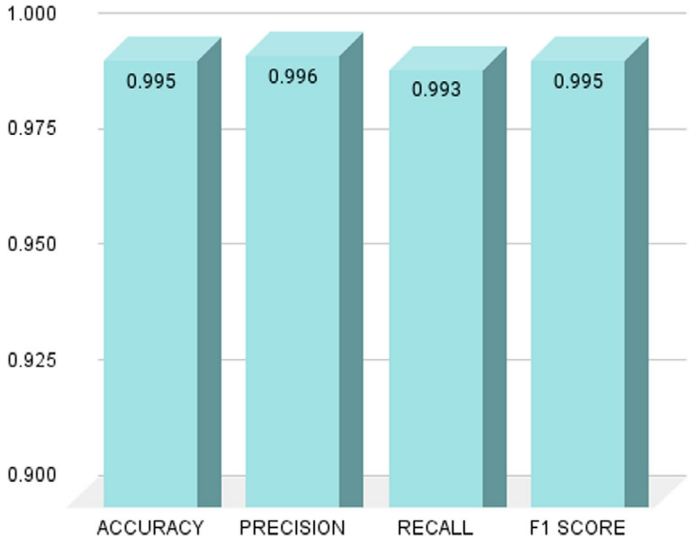


Fig. 4 Results of proposed model

Fig. 5 Confusion matrix of proposed model

4.1 Comparison

Although many investigations have been done, to the best of our knowledge, there are just a few publications on employing a bagging model to identify XSS payloads; among all, only one similar to our model is [9]. A few publications that have been produced on the same subject but with a stacking model have also been included to expand the scope of our research, which will be seen side by side in Table 1. Despite being irrelevant to our paper, it provides an overview of how additional models with the potential to outperform others might also be used for detection.

The existing work [9] employed an ensemble bagging method with a Random Forest classifier. The results of their model are shown in Fig. 6.

In [9], data was acquired from Kaggle and Github, and a new dataset was created, combining data from each. It did not specify the amount of data extracted from each dataset. In this work, the proposed model used a dataset from Kaggle, which contained benign URLs and XSS payloads. Table 1 illustrates the outcomes of the proposed model in comparison to the existing models [7–9], where both Stacking

Table 1 Comparison of different models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Design
IN [7]	99.5	99.5	99.7	99.6	Stacking
IN [8]	–	–	98	–	Stacking
IN [9]	97.1	96	97	–	Bagging
PROPOSED	99.5	99.6	99.3	99.5	Bagging

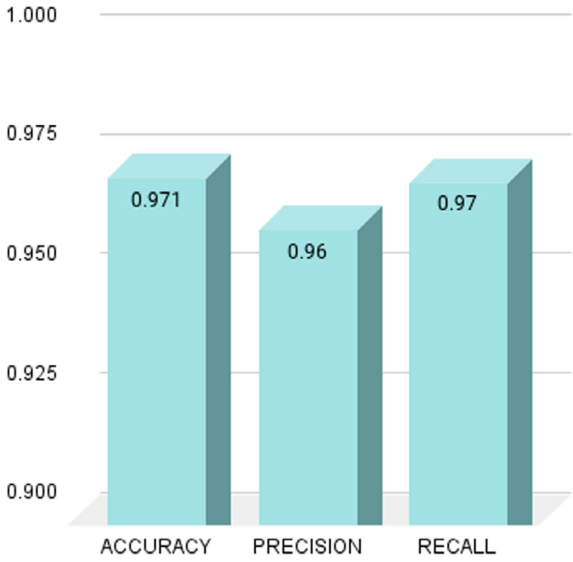


Fig. 6 Results of existing work [9]

and Bagging model results are contrasted with each other papers referenced as [7, 8] have been thoroughly examined within the contents of related works section. Table 1 demonstrates that our model outperformed in all metrics compared to the Bagging paper [9], including accuracy, precision, and recall. However, the model lacks the F1 score evaluation parameter as observed.

5 Conclusion

Our study focused on utilizing an ensemble method, specifically Bagging with the Random Forest classifier, to detect XSS payloads. The core objective was to enhance the robustness and accuracy of XSS detection mechanisms in web applications. With an accuracy rate of 99.5%, precision reaching 99.6%, recall at 99.3%, and an F1 Score of 99.5%, our approach demonstrated superior capability in accurately identifying XSS vulnerabilities. Our approach consistently outperformed the existing methodology across all given parameters through thorough and comprehensive evaluation, showcasing significant enhancements in XSS detection capabilities. These compelling results underscore the validity and effectiveness of our ensemble-based approach for XSS detection, highlighting its potential to address security challenges effectively in web applications.

References

1. Open Web Application Security Project (OWASP) Top Ten, URL: https://owasp.org/www-project-top-ten/# 2021.
2. Bukhari SN, Ahmad Dar M, Iqbal U (2018) Reducing attack surface corresponding to type 1 cross-site scripting attacks using secure development life cycle practices. In: 2018 4th International conference on advances in electrical, electronics, information, communication and bioinformatics (AEEICB), 27–28 Feb 2018. https://doi.org/10.1109/AEEICB.2018.8480945
3. Gan J-M, Ling H-Y, Leau Y-B (2020) A review on detection of cross-site scripting attacks (XSS) in web security. In: Advances in cyber security: second international conference, ACeS 2020. Springer, Singapore, pp. 685–709. https://doi.org/10.1007/978-981-33-6835-4_45
4. Open Web Application Security Project (OWASP) Types of XSS
5. URL: https://owasp.org/www-community/Types_of_Cross-Site_Scripting
6. Kumar JH, Ponsam JJG (2023) Cross-site scripting (XSS) vulnerability detection using machine learning and statistical analysis. In: 2023 International conference on computer communication and informatics (ICCCI), 23–25 Jan 2023. https://doi.org/10.1007/s10462-023-10433-3
7. Chen C, Liaw A, Breiman L (2006) Using random forest to learn imbalanced data. Department of Statistics University of California—Berkeley East Lan- sing Michigan Tech. Rep
8. Krishnan M, Lim Y, Perumal S, Palanisamy G (2022) Detection and defending the XSS attack using novel hybrid stacking ensemble learning-based DNN approach. Digit Commun Netw. https://doi.org/10.1016/j.dcan.2022.09.024

9. Perumal S, Kola Sujatha P (2021) Stacking ensemble-based XSS attack detection strategy using classification algorithms. In: 6th IEEE international conference on communication and electronics systems (ICCES-2021), 8–10 July 2021. https://doi.org/10.1109/ICCES51350.2021.9489177
10. Barde SS (2020) Cross-site scripting detection using random forest and dataset ensemble modelling. Masters thesis, Dublin, National College of Ireland 16 Aug 2020
11. Kaggle (2020) Cross site scripting XSS dataset accessed. https://www.kaggle.com/datasets/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning?select=XSS_dataset.csv

A Machine Learning-Based Cross-Layer DoS Attack Detection Technique for IoT

Aditi Paul, Shirashti Chaudhary, and Somnath Sinha

Abstract The most potent and common attacks on Internet of Things (IoT) are Denial of Service (DoS) attacks. Unfortunately, because the attack occurs on numerous layers, a single layer detection method is insufficient and ineffective to counteract these attacks. The current work focuses on the detection of cross-layer DoS assaults using Machine Learning-based multiclass classifiers. Three attacks against Routing Protocol (RPL) and Transmission Control Protocol (TCP) are detected using three ML Classifiers (KNN, Gradient Boosting, Random Forest). The novelty of the study is the design of cross-layer attack datasets using feature engineering technique. The performances of the classifiers are analyzed in presence of both balanced and imbalanced datasets. The results show that Gradient Boosting classifier has highest accuracy of up to 98% with deviation of up to 97%.

Keywords Cross-layer DoS attacks · ML classifier · IoT security · Sinkhole attack · RPL attacks

1 Introduction

The network of actual physical objects or "things" that are equipped with sensors, software, and communication capabilities so they can gather and share data over the internet is known as the Internet of Things (IoT). These products could be commonplace household goods, mobile or industrial equipment, or even wearable technology. These things can communicate, evaluate data, and make decisions on their own thanks to the Internet of Things. IoT devices aim to accomplish their desired goals with the nominal involvement of human beings. But on the other hand, these devices have many security threats that an attacker can utilize to have a ravaging impact on

A. Paul (✉) · S. Chaudhary
Banasthali Vidyapith, Tonk, Rajasthan 304022, India
e-mail: aditi23.mca@gmail.com

S. Sinha
CHRIST (Deemed to be University), Bengaluru 560029, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_2

them. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are observed as the most threatening attacks that compromise IoT devices and make them availability to the legitimate users. DDoS attacks are special types of DoS attacks, where multiple resources are used to flood a targeted system. These types of attacks are more complex to prevent as multiple scattered devices are used to launch the attack. In IoT, RPL is the most targeted protocol which is a specially designed for Lossy Network and lightweight in nature. RPL meets the power and storage constraints of tiny sensor nodes deployed in large number to host an IoT application. Thus, security of these nodes is very challenging as remote accessing is not so strongly employed on these sensors once deployed. Also, implanting complex security measures within these sensors are resource and energy consuming. This makes RPL protocol to be vulnerable to multiple attack surfaces. Thus, securing RPL from IoT attacks is a crucial issue. Most of the current literature is focused on Machine Learning-based detection and prevention of DoS attacks on RPL. Keeping in mind this trend, current study considers three types of DoS attacks targeted on RPL and their detection using Machine Learning models. Rather than detecting the attacks individually, a cross-layer dataset is generated using feature engineering technique. The three attacks are outlined in brief:

- *Sinkhole Attack:* In sinkhole attacks, the attacker node advertises false routing information by compromising a genuine node to transit the routing packets through it. When packets reach to the attacker node, it either drops the packets or forward it to other nodes leading to more variants of the attack. Sinkhole is a routing attack that changes the network topology and finally disburse the network. In RPL, this attack forges DIO messages by lowering the rank value which attract the neighbor nodes to route their packets through it.
- *Syn Flooding Attack*: It is a type of DoS attack that creates multiple TCP connections but never acknowledge the response. This leads to the receiver node to be flooded with huge TCP Synchronization Request packets which cannot be processed further. The target of the attack is to make the TCP protocol unresponsive.
- *DIS Flooding Attack:* This is type of flooding attack, in which an outsider or insider node floods DIS control messages repeatedly to all the neighbor nodes. This causes the neighbor nodes to flood the network with DIO messages, causing a significant impact on control packet overhead, power drainage, delay, etc.

The unique aspect of the proposed study is to develop a detection system that firstly generates a cross-layer attack dataset comprises both flooding and routing attacks occurring at multiple layers of IoT and secondly to design a Machine Learning-based detection algorithm that detects both types of attacks at a time. The model's efficiency is justified with multiple combinations of datasets both balanced and unbalanced. The contribution of the paper is as follows:

- Designing cross-layer DoS attack dataset comprising of flooding and routing attacks as mentioned.

- Designing a detection system using ML models that best perform with designed dataset.
- Analyzing the efficiency of the model with datasets having high and low variance.

The rest of the paper is organized as follows:

- Section 2 represents a thorough literature review on cross-layer DoS attack detection mechanisms proposed so forth in IoT domain.
- Section 3 designs the detection system
- Section 4 implements the system using Python codes for ML models
- Section 5 analyzes the performance of the proposed model on multiple datasets
- Section 6 concludes the paper with future scope.

2 Literature Review

In order to detect DDoS attacks, working [1] provides a feature selection-based framework combining filter-based feature selection approaches with threshold. The performance of an ensemble feature selection or reduction method that combines different filter-based feature selection strategies may be offered in the future for additional feature reduction.

In [2], authors proposed a hybrid methodology, in which four classifiers were subjected to applications of the chi-square, extra tree, and ANOVA feature selection methods. The study used Random Forest, Decision Tree, K-Nearest Neighbors, and XGBoost. Tuning of hyperparameters can be proposed as a future study to further improve performance parameters.

In [3], data mining technologies for attack prediction are developed. The suggested prediction system can locate potential DDoS attacks and forecast their occurrence. To achieve the goal of defense, the source of an attack is then actively stopped. A more thorough protection system can be built using the current IP tracking technology.

In [4], comparison to the conventional IDSs, the Naive Bayes classification algorithm with the usage of numerous agents provides greater performance for the detection of DDoS attacks. By substituting a lightweight pattern matching technique for the Naive Bayes classification approach, this study can be advanced even further.

The work in [5] suggested using deep adaptive reinforcement learning to protect honeypots from runtime DoS attacks. Future versions of this should be enhanced to better handle different attack detection tactics, corruption, information disclosure, subversion, and UDP flooding.

In [6], authors show how Machine Learning (ML) may be effectively used to categorize Distributed Denial of Service (DDoS) tools into five groups, four of which are associated with DDoS attacks and one of which contains harmless data. In order to improve efficiency and understanding of the underlying data, the study emphasizes the significance of just integrating pertinent features in machine learning models. The results demonstrate the model's ability to greatly enhance DDoS attack detection and

mitigation by indicating that it is dependable and accurate in predicting the target variable.

Gayathriet al. in [7] proposed energy-efficient, trustworthy routing protocols for IoT blackhole and wormhole attacks. To increase the quality of trust validations against black hole and wormhole attacks, this protocol will be updated with appropriate machine learning and deep learning approaches in the future.

In [8], authors present an application-layer protocol communication model based on Event-Driven Recurrent Networks (EDRN) and focuses on identifying Application-Layer Distributed Denial of Service (AL-DDoS) threats. Application-layer protocol keywords and the intervals between them are used as input characteristics in this model. The paper shows that the EDRN-based model outperforms traditional Recurrent Neural Networks (RNNs) through experimental evaluation and comparison. The findings demonstrate increased threat detection accuracy and reduction of DDoS assaults.

An IDS model is trained, evaluated, and tested using two distinct datasets in [9]. These datasets contained well-known cyberattacks such as Heartbleed, botnets, distributed denial of service (DDoS), denial of service (DoS), infiltration, web attacks, botnets, ports scan, and DDoS. After conducting numerous trials on various Machine Learning (ML) techniques an ensemble classifier is used for each model. Finally, the two best-performing boosting classifiers (XGB and LGBM) were incorporated into a boosted ensemble ML model.

The main goal of [10] is to replace current centralized system solutions with decentralized ones to prevent DoS attacks on IoT devices at the application layer. To do this, the study authenticates and verifies these devices using public blockchain technology, which offers a reliable and tamper-proof platform. Additionally, a public blockchain with scalability issues is used to construct this system.

In [11], Bajaj et.al. proposed a cross-layer intrusion detection system that can detect both single and cross-layer attacks in IoT. The model uses ensemble learning for better accuracy. Simulation results show a detection accuracy of 96.5% for single layer attack and 94.98% for cross-layer attack.

3 Proposed Model

In this work, we present a resilient system that can identify Denial of Service (DoS) and flooding assaults, with particular attention to vulnerabilities in the Transport Layer and the Network Layer of the Internet of Things (IoT) protocol stack. The Synchronization Flooding, or TCP SYN flood, is a major danger at the Transport Layer. This exploit floods the network with TCP SYN requests, overloading it. As a result, there is an extreme demand on resources, which causes a valid target to fail and prevents fresh connection installations. We now discuss the RPL flooding attack at the network layer, in particular its Distributed Information Service (DIS) flooding variation. This kind of denial of service attack occurs when a rogue node continuously transmits DIS packets to nearby nodes. Because of this nefarious behavior, there is

a noticeable rise in control packets, which significantly depletes resources. Sinkhole attack on RPL is a type of routing attack that collapses the network by dropping routing packets. This type of attack changes the routing topology by advertising false route information to the neighbor nodes thus attracting huge packets toward the attacker itself.

The proposed detection system first generates a cross-layer attack dataset using feature engineering technique and in the second stage designs the detection system using ML models (Fig. 1). The aim is to identify the feature set that contributes the most to each of the attacks.

The steps of the cross-layer trace feature set generation are explained below:

- Generating traces of individual attack using simulation: We model and create three different kinds of assaults, each corresponding to a different tier of the Internet of Things architecture. We model the following assaults using the NetSim simulator: the TCP SYN flood (Fig. 2), the DIS flooding attack (Fig. 3), and the Sinkhole attack (Fig. 4). We simulate these assaults and then produce appropriate traces for every possible scenario. All of the trace files are carefully built to include

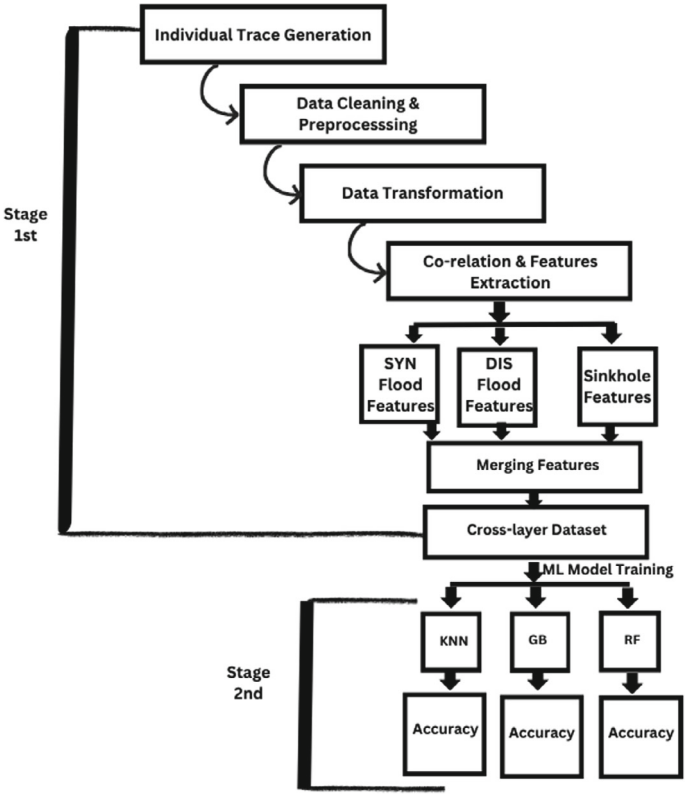


Fig. 1 Proposed ML-based cross-layer DoS attack detection model in IoT

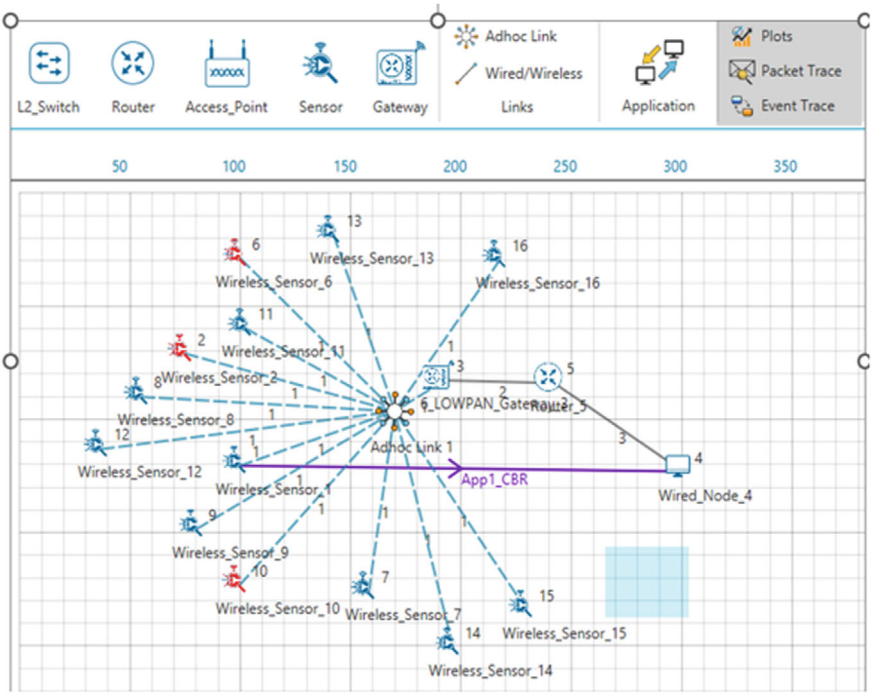


Fig. 2 SYN flooding attack with 10 nodes, 3 attackers

both instances of malicious assaults (class type '1') and typical network activities (class type '0'). In later stages of analysis and detection system development, this dual labeling in the target column of the trace files enables the classification and differentiation of regular network activities from the simulated attacks.

- *Data preprocessing and data cleaning*: Every trace files are cleaned against missing values/NA and single values. Rows and column having only Not Applicable (NA) are removed. This reduced irrelevant data from the traces. Some columns having a single value for all rows are also removed some of the sub-category whose counts are negligible with respect to other sub-categories under same columns are removed as these do not impact on the overall dataset.
- *Data transformation*: After data cleaning, the dataset has to be converted to numerical value. For categorical data, one hot encoding scheme is used. This scheme converts categorical data to binary values according to its types. However, an obvious outcome of one hot encoding is increased columns. In some cases where category represents nodes number, numeric is used instead of alphanumeric. For example, node 1 is replaced by 1. The outcome of data transformation is the conversion of data to numeric type.
- *Correlation and feature extraction*: Correlation is an important feature extraction technique which helps to identify the correlated features in the dataset. Highly correlated features can be treated as a single feature which in the proposed study is

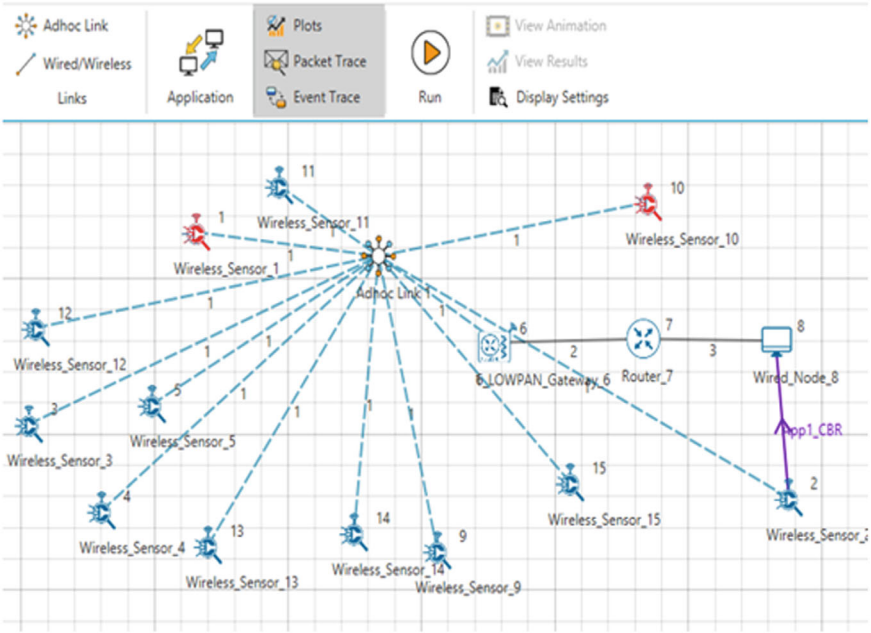


Fig. 3 DIS flooding attack with 10 nodes, 2 attackers

used to extract the highest contributing features for training the ML models. Here by using Pearson correlation method, highly correlated features are first identified. Out of these features, those giving highest training accuracy are selected, rests are removed. Several trials are executed in this stage to identify the best feature set producing highest accuracy.

- *Merging all attack traces into single cross-layer dataset*: This is the most promising stage of designing cross-layer attack dataset. For each attack Pearson correlation technique identifies set of highly correlated features which are never similar with other attack correlated features. This creates challenge to merge three attacks into 1 file with similar features. From all attacks, common correlated features are selected and merged to single dataset. Then a trial is made to check the training accuracy of the ML models with the merged features. The feature set that gives best accuracy is selected as the final features in the merged dataset after correlation and generates the cross-layer dataset.

The cross-layer dataset is then used to train three ML models which are Gradient Boosting (GB), Random Forest (RF), and K-nearest neighbor (KNN). The reason for considering these ML model is to compare the performance between Tree-structured (RF, GB) and Non-tree structured (KNN) ML models. Since the proposed detection model is a multiclass classification model, we set the ML models hyperparameters for multiclass classification. Once this is done, each model is trained with the cross-layer dataset and training accuracy is compared.

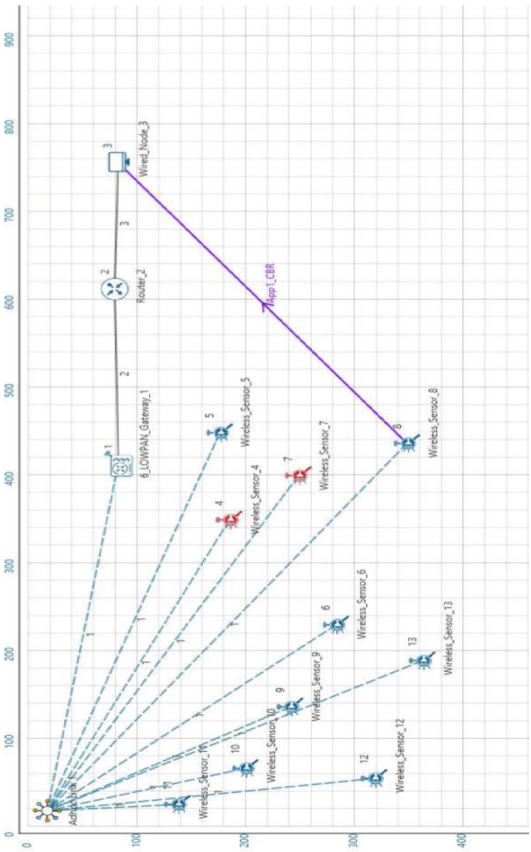


Fig. 4 Sinkhole attack with 8 nodes, 2 attackers

Finally, the three ML models are tested with balanced and imbalanced datasets to analyze their performances with respect to randomness of data.

4 Implementation

The proposed model is implemented in two stages. For creating cross-layer dataset in the first stage three attacks are designed in the NetSim simulator which are DIS flooding attack, TCP SYN flood attack and Sinkhole attack. The network environment along with number of attacker nodes for each attack are represented in Figs. 2, 3, and 4. Simulation is run for 100 s in each attack and trace files are generated (.csv format). Now the trace files are preprocessed, transformed and features are extracted following steps in Fig. 1. The final cross-layer dataset is generated by merging individual dataset (Table 1) having 262,756 rows and 14 columns (features). The target column termed as Class-Type has four classes 0(no-attack), 1, 2, 3 for each of the attack.

The second stage is to train the ML models with cross-layer dataset. Python Scikit-learn library is used for training each model. Hyperparameter tuning is done for each model to get maximum accuracy. During training, the cross-layer final dataset is split as 80% training and 20% testing ratio for each ML model.

Finally, the performance comparison is executed among the three ML models and justification is presented accordingly. For this, we have created two sub-datasets with a ratio 1:1 and 1:2 between attack and non-attack from the master dataset. These sub-datasets are used to train each ML models to analyze their performance. 1:1 is a balanced dataset having equal number of attack and non-attack classes in the target column, whereas 1:2 is an imbalanced dataset which has double attack class than non-attack class in the target column. A comparison of ML models' performance in presence of imbalanced datasets is also analyzed and represented through graphs in the results section. The next section discusses the result analysis with graphical representation.

5 Result Analysis

The notion of the proposed study is to design and analysis of ML-based DoS and flooding attack detection technique in IoT. We represent each of the three ML models' detection accuracy with variation of datasets.

In Fig. 5, the accuracy of KNN is 97.99% which is the highest accuracy among three models. There is a decrease (96.03%) in the accuracy as the dataset becomes imbalanced which is obvious as the number of attacks is double than non-attack classes.

In Fig. 6, the accuracy of Random Forest is 93.97% and 93.54 in 1:1 and 1:2 ratio, respectively. This indicates that Random Forest has little effect of the imbalanced data than KNN.

Table 1 Features, number of rows, and column in final cross-layer dataset

Features of cross-layer dataset (262,756 × 14)						
PACKET_ID	PACKET_TYPE_ Control_packet	CONTROL_ PACKET_TYPE/ APP_NAME_DIO	CONTROL_ PACKET_TYPE/ APP_NAME_DAO	SOURCE_ID	DESTINATION_ID	RECEIVED_ID
Features of cross-layer dataset (262,756 × 14)						
NW_LAYER_ ARRIVAL_TIME (US	NW_LAYER_ PAYLOAD (Bytes)	MAC_LAYER_ PAYLOAD (Bytes)	PHY_LAYER_ OVERHEAD (Bytes)	PACKET_STATUS_ Successful	PACKET_STATUS_ Collided	CLASS_TYPE

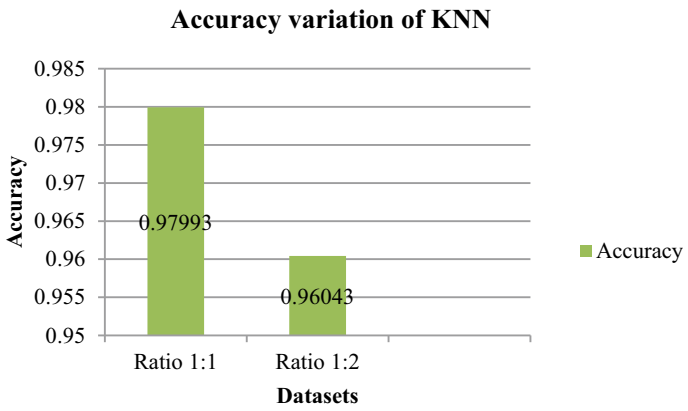


Fig. 5 Detection accuracy of KNN

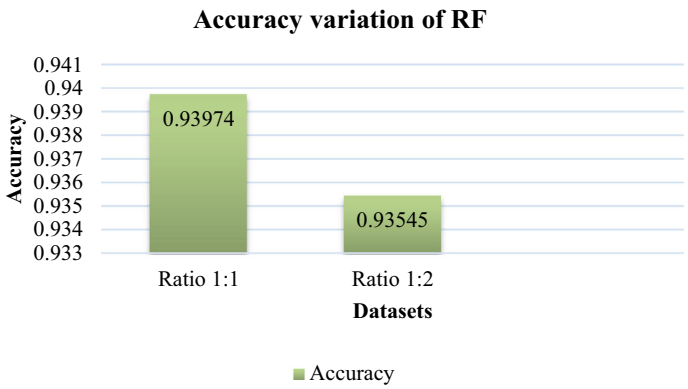


Fig. 6 Detection accuracy of RF

Figure 7 shows the performance of Gradient Boosting with respect to two datasets. Here noticeably, we see that GB has a high accuracy (97.66%) close to KNN and lower deviation (96.83%) than RF.

From Fig. 8, we can infer that Gradient Boosting has the best performance than other as it has higher accuracy with lower deviation in imbalance data.

6 Conclusion and Future Scope

In the current study, a cross-layer attack detection model using Machine Learning-based multiclass classifier is proposed for IoT architecture. The models are designed by using three ML models among which two are tree-based classifiers (GB, RF)

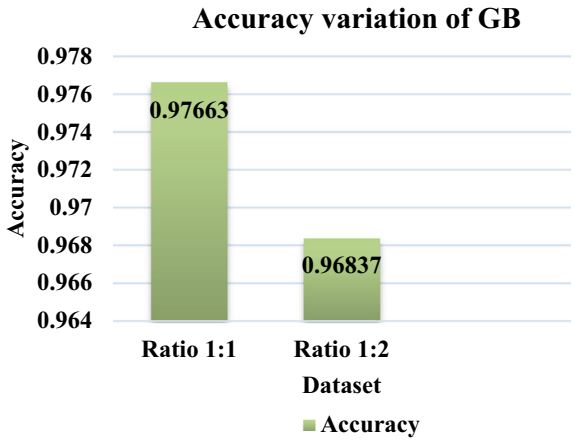


Fig. 7 Detection accuracy of GB

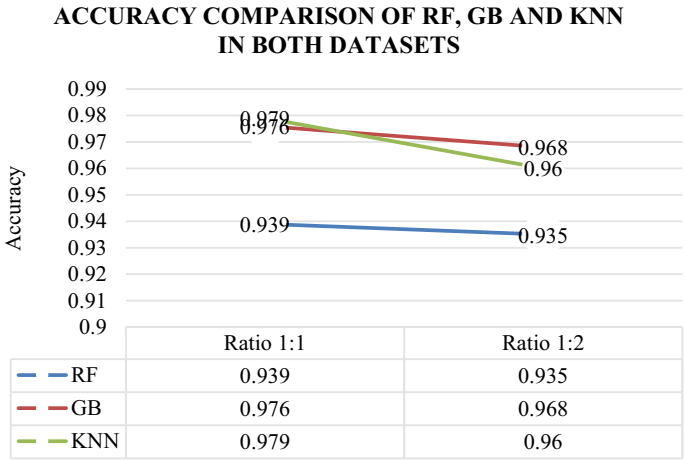


Fig. 8 Performance comparison of three ML model with respect to two datasets

and one is non-tree classifier (KNN). The notion is to achieve best accuracy among multiple models with lowest deviation in imbalanced dataset. The novelty of the proposed method is twofold: first to design a cross-layer attack datasets consisting of multiple attack classes by using feature engineering, second to analyze the performance of the classifiers in presence of data imbalance. The result analysis shows a satisfactory performance of both KNN and GB with approximately 98% of accuracy. Although RF shows a lower accuracy of approximately 94%, it has little impact of data imbalance.

However, the model is developed under some constraints which are:

Number of attacks chosen are from two layers of IoT. More layers can be added in future to develop a robust and sustainable model.

In the future, the ML models can be ensembled together to increase the accuracy, lesser computation time and lowering the effect of imbalanced dataset.

In order to address the scalability and complexity issue, the proposed system can be implemented in centralized manner where the central node can be an access point or anchor node having higher computation and resource ability. This will also solve the issue of device heterogeneity.

References

1. Kshirsagar D, Kumar S (2022) A feature reduction based reflected and exploited DDoS attacks detection system. J Ambient Intell Humaniz Comput 13(1):393–405
2. Gaur V, Kumar R (2022) Feature selection method for DDoS attack detection. In: International conference on electronics and renewable systems 2022 (ICEARS), vol 954, pp 939–944. IEEE
3. Huang L (2022) Design of an IoT DDoS attack prediction system based on data mining technology. J Supercomput 78(4):4601–4623
4. Mehmood A, Mukherjee M, Ahmed SH, Song H, Malik KM (2018) NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. J Supercomput 74:5156–5170
5. Veluchamy S, Kathavarayan RS (2022) Deep reinforcement learning for building honeypots against runtime DoS attack. Int J Intell Syst 37(7):3981–4007
6. Mohammed Sharif D, Beitollahi H, Fazeli M (2023) Detection of application-layer DDoS attacks produced by various freely accessible toolkits using machine learning. IEEE Access 2023 11:51810–51819. https://doi.org/10.1109/ACCESS.2023.3280122
7. Gayathri A, Prabu AV, Rajasoundaran S, Routray S, Narayanasamy P, Kumar N, Qi Y (2022) Cooperative and feedback based authentic routing protocol for energy efficient IoT systems. Concurr Comput: Pract Exp 34(11):e6886
8. Xie B, Wang Y, Wen G, Xu X (2023) Application-layer DDoS attack detection using explicit duration recurrent network-based application-layer protocol communication models. Int J Intell Syst 12(37):1–13
9. Okey OD, Maidin SS, Adasme P, Lopes Rosa R, Saadi M, Carrillo Melgarejo D, Zegarra Rodríguez D (2022) Boosted EnML: efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning. Sensors 22(19):7409
10. Ibrahim RF, Abu Al-Haija Q, Ahmad A (2022) DDoS attack prevention for internet of thing devices using ethereum blockchain technology. Sensors 22(18):6806
11. Bajaj P, Mishra S, Paul A (2023) Comparative analysis of stack-ensemble-based intrusion detection system for single-layer and cross-layer DoS attack detection in IoT. Sn Comput Sci 4:562. https://doi.org/10.1007/s42979-023-02105-4

Advanced Neural Network MRI Evaluation for Automated Knee Trauma Detection

Simran Kumar, Bhavika Salvi, Krina Limbachiya, and Hemanth Varma

Abstract The paper introduces a method for automated diagnosis of knee injuries using magnetic resonance imaging (MRI) sequences. MRI examinations serve as a crucial function in detecting knee ailments, yet the procedure is lengthy and susceptible to inaccuracies. The proposed approach utilizes sophisticated deep learning methodologies, particularly convolutional neural networks (CNNs) like AlexNet and SqueezeNet, to scrutinize MRI sequences. Different techniques, including attention mechanisms, are employed to reduce sequence information and predict injuries such as ACL and meniscus tears. The study demonstrates that a multi-model ensemble, combining the strengths of various networks, surpasses solitary frameworks concerning the Area Beneath the Receiver Operating Characteristic (ROC) Curve for varied categories of harm. Future work could explore diverse feature extraction methods and more advanced ensemble techniques for better performance.

Keywords Magnetic resonance imaging · Knee injury · Deep learning · Convolutional neural network

S. Kumar (✉)
Department of Mathematics and Computing, National Institute of Technology, Hamirpur, Himachal Pradesh, India
e-mail: simrankumarr39@gmail.com

B. Salvi
Fr Conceicao Rodrigues College of Engineering, Mumbai, India

K. Limbachiya
Malaviya National Institute of Technology, Jaipur, India

H. Varma
Indian Institute of Information Technology, Bhagalpur, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_3

1 Introduction

MRI is a method of obtaining three-dimensional images of the inside of an object by using strong magnets to align the protons in the object and, subsequently, electromagnetic wave currents to disturb and quantify that orientation. An MRI protocol consists of a succession of two-dimensional images that can be layered to construct a three-dimensional representation. There are three sequence types, each corresponding to an orientation of the "camera" that captures the two-dimensional slices. The axial sequence captures slices of the object that are horizontal to the ground; the coronal sequence captures vertical slices as viewed from the front of the object; and the sagittal sequence captures vertical slices as viewed from the side of the object.

MRI can be a helpful tool when diagnosing knee injuries [8, 17, 21]; however, analyzing the images is a time-consuming process, and even with trained professionals, it is easy for a clinician to misdiagnose an injury based on an MRI reading [14]. Improving the automated identification of abnormalities in knee MRIs could help prioritize which MRIs to examine first and provide better early results for patients whose scans appear normal. Model predictions could also provide a "second opinion," reducing the possibility of missed abnormalities. This could represent a significant cost saving for hospitals and an increased level of care for patients. The problem scrutinized in this research is as such: given three MRI images (axial, coronal, and sagittal) of a patient's patella, can we forecast injuries requiring surgical intervention? In particular, we wish to predict whether the knee is healthy, has an ACL tear, has a meniscal tear, or has any other abnormality. Given that these traumas may coincide, we aim to anticipate three separate binary metrics: anomalous ACL rupture and meniscus tear. Past research has proven that MRI scans are accurate and sensitive tools for detecting these kinds of injuries in a non-invasive manner [4, 7, 25].

2 Previous Research

There are two primary obstacles encountered when employing automated techniques on MRI sequences. The initial hurdle stems from the extensive information within each MRI scan, given its three-dimensional nature. Deciphering how to derive significance from a variable-length sequence of images presents a considerable challenge. The subsequent hindrance relates to the challenge and expenditure of gathering MRI information, culminating in insufficient datasets for most deep learning approaches. Accordingly, profound cognition frameworks should be confined to restricted parameters and modifications to forestall overfitting.

The latest breakthroughs in profound learning for MRI scrutiny concentrate on the spatial aspect of MRI depictions by employing 3D-CNNs. These networks expand upon (channel, width, height) convolutions in traditional 2D images involving channels, width, and height dimensions to include time as an additional dimension. These

3D-CNNs can effectively process the volumetric data captured in MRI scans by incorporating time alongside (time, channel, width, height) [11, 13, 24, 27]. However, a challenge with these models lies in the many parameters they require and the need to train them from scratch, which can be time-consuming and resource-intensive.

Another conventional technique is the 2.5D strategy, where models undergo training on sliding windows of the input sequence [2, 9, 20]. In this arrangement, the inputs to the neural network generally encompass (duration, breadth, altitude), and it's assumed that all visuals have a single-layer depth. While this methodology demonstrates efficacy for line-to-line concerns, it presents novel hurdles while compressing the scan into a singular prognosis. Recurrent neural networks (RNNs) offer an additional approach for handling a sequence of two-dimensional visuals, as demonstrated in [10, 19]. Nevertheless, the necessity to commence training anew and the abundance of parameters in such models continue to pose a significant obstacle for modest datasets.

The most comparable framework to the one detailed in this thesis is MRNet, developed by Bien et al. [3] and stands as the cutting-edge technique for discerning trauma in MRI images of the knee. Their approach employs traditional CNNs and leverages the pretraining of an AlexNet network on a comprehensive dataset such as ImageNet. Every succession of visuals undergoes analysis via the pre-established network to generate a tensor of dimensions $(s \times c \times w \times h)$, wherein s signifies the span of the succession, c indicates the depth of the conduit, w represents the breadth, and h denotes the altitude. Subsequently, the breadth altitude measurements diminish through worldwide mean pooling, yielding a $(s \times c)$ tensor. These attributes undergo additional compression via maximum pooling across the line and subsequently traverse a kernel perceptron equipped with logistic activation to forecast the likelihood of the sequence. The likelihoods at the series level are then fed into a logit modeling classifier to determine the ultimate probability for diagnosis.

3 Techniques

3.1 Series Network

Illustration 1 manifests a conceptual process for converting an individual MRI arrangement into an anticipation of harm. On the port face of the depiction, a sequence of n depictions, depicted as a $\mathbb{R}^{3\times224\times224}$ array, is introduced inside the system together as an integrated assortment. The characteristic processor evolves every picture into a dataset of proportions $\mathbb{R}^{c\times w\times h}$, wherein c signifies the number of channels and w and h signify the breadth and altitude, correspondingly. These attributes are subsequently compressed into an array of vectors of magnitude $\mathbb{R}^c$ within the overarching mean amalgamation stratum. Following that, the full collection of n vectors is condensed into an individual vector of dimension $\mathbb{R}^c$ inside the series diminishment veil. The proposed array subsequently operates on a linear gradient and a logistic excitation

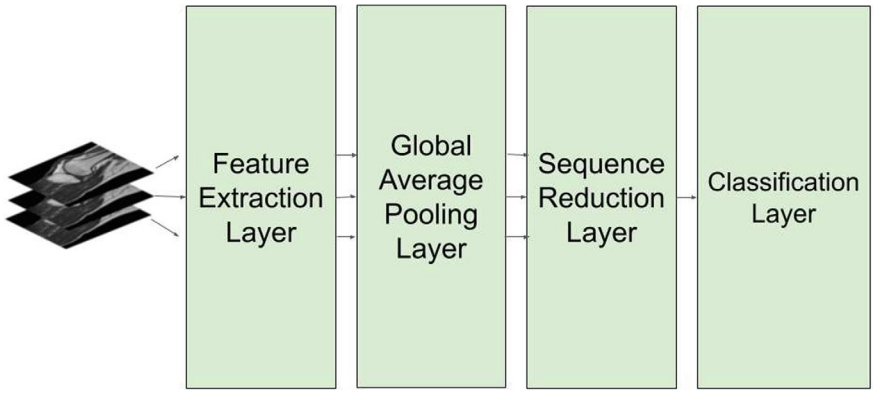


Fig. 1 Conceptual perspective on the sequence-specific connections. Visual representation of a knee magnetic resonance imaging (MRI) captured from [5]

mechanism within the categorization tier to produce a likelihood signifying if the whole MRI series hints at the existence of the particular trauma.

To specify, if we delineate the characteristic abstraction stratum functioning as a mechanism $f\colon \mathbb{R}^{n\times 3\times 244\times 244} \to \mathbb{R}^{n\times c\times w\times h}$, and the sequence-reduction layer as a function $g\colon \mathbb{R}^{n\times c} \to \mathbb{R}^{c}$, then the entirety of the network can be expressed as:

$$\mathrm{gap}_c\left(x^{(i)}\right) = \frac{\sum_{j=1}^{w}\sum_{k=1}^{h} x_{c,j,k}^{(i)}}{\mathrm{wh}}$$

$$p = \sigma(g(\mathrm{gap}(f(x)W_c + b_c))), \tag{1}$$

where p denotes the projected probability, $W_c \in \mathbb{R}^{c\times 1}$ and $b_c \in \mathbb{R}_c$ represent the coefficients and biases of the categorization stratum, while σ stands for the sigmoidal function: $\frac{1}{1+e^{-x}}$.

The most advanced technology to date, MRNet—discussed in Sect. 2—can be delineated as leveraging fragments from an existing AlexNet which is utilized for feature retrieval segment and implementing an element-by-element procedure to compress datasets into condensed modules.

3.2 SqueezeNet

SqueezeNet is a convolutional deep learning model that achieves impressive results in the ImageNet competition. It consists of a sequence of "spark" layers, each comprising a set of 1 × 1 filter convolutions, followed by a Rectified Linear Unit (ReLU), coupled with 1 × 1 and 3 × 3 filter convolutions (the expanding stratum), amalgamating their outcomes prior to a concluding Rectifier Activation function. A

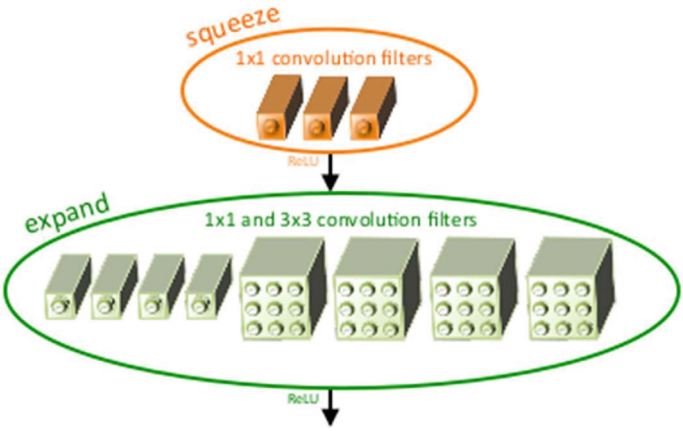


Fig. 2 Depiction of a blaze stratum. Picture captured directly from "SqueezeNet: Achieving AlexNet-level precision with 50 times fewer parameters a model size of less than 0.5 MB" [12]

representation portraying a spark stratum is presented in Fig. 2. They are laced with maximal grouping strata prior to the 1st, 4th, and 8th spark strata, and the whole paradigm begins with an orthodox 7×7 stratum.

Enhancing the framework which has undergone prior training on the ImageNet repository emerges as an approach which has demonstrated efficacy in categorizing motor vehicles [1], farming maladies [6], and ocular situations [18]. By omitting the ultimate strata from a previously modeled architecture, the team employs it as a smooth alternative for the text retrieval tier as illustrated in Fig. 1.

3.3 Attention

It represents a mechanism that conventionally served as a synthesis method for obscured circumstances within a cyclic neural network. Given one or more questions $\mathbf{Q} \in \mathbb{R}^{1\times d}$, a group of codes $\mathbf{K} \in \mathbb{R}^{n\times d}$, and a group of properties $\mathbf{V} \in \mathbb{R}^{n\times d}$, the focus valuation may be determined by computing the softmax of the scalar multiplication between the inquiry and codes:

$$s = \left(\mathbf{Q}\mathbf{K}^T\right),$$

$$a_i = \frac{e^s i}{\sum_{j=1}^{k} e^s j}.$$

According to Tan and colleagues [23], whose research was influenced by Luong and team's position-based focus [15]. The concentration gauges might afterward function as factors while calculating the average of the quantities $\mathbf{V}$.

For an RNN, the inquiry is frequently the ultimate latent condition; the indications are either the initial embeddings or the concealed circumstances at every moment, and the assessments are in the same condition at every temporal interval. A conclusive depiction of the complete succession is produced by computing the scalar product, the focal point vector, and the assessments a $\mathbf{V}$.

For the investigations outlined in this manuscript, we choose to abstain from employing an RNN and instead employ the amalgamation of attention-weighted averages throughout the frames of the MRI sequence yields the definitive representation of the series. Here, the investigation constitutes a conditioned factor of the system, $\mathbf{Q} \in \mathbb{R}^{1\times c}$. In contrast, the signals and advantages are derived from the overall averaging layer, $\mathbb{R}^{s\times c}$. This permits us to employ the attention-weighted summation as the sequence-reduction layer:

$$\mathbf{K} = \mathbf{V} = \text{gap}(f(x)),$$

$$g(x) = \text{softmax}\left(\mathbf{Q}k^T\right)\mathbf{V}.$$

3.4 Sequence Ensembling

Because we have access to multiple MRI sequences per patient, separate sequence-specific networks can be trained to predict a diagnosis probability for each of the axial, coronal, and sagittal sequences. The three probabilities are then fed inside a logit classifier to arrive at the final damage forecast, as demonstrated in Fig. 3.

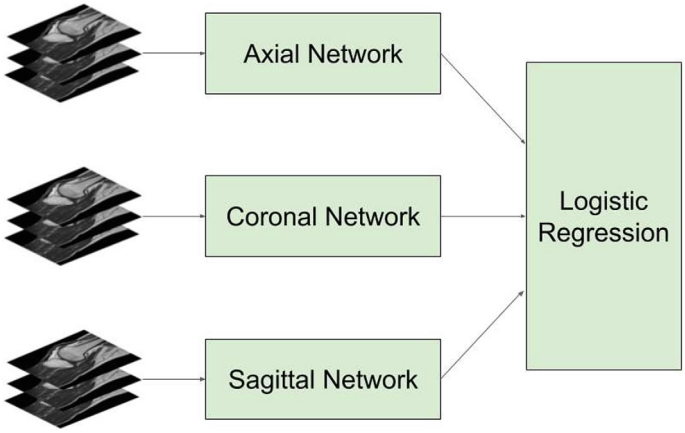


Fig. 3 Ensemble forecasts from the specialized networks to create a prognosis for injury

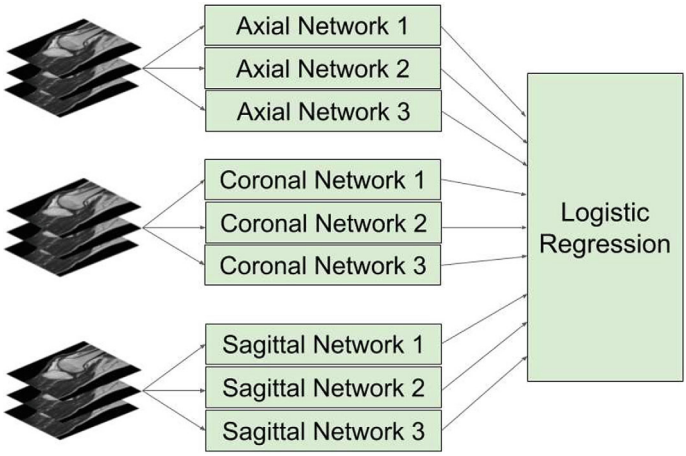


Fig. 4 Ensembling forecasts by various sequence-specific network categories to produce a trauma prognosis

3.5 Design and Sequence Combination

Furthermore, utilizing every instance in series aggregation allows us to leverage the projections across diverse prototype varieties, as depicted in Fig. 4.

3.6 Loss

Due to the disproportionate distribution of class instances in the dataset, all distinctive-specific paradigms are educated on utilizing a balanced binary deviation descent procedure:

$$L = -\frac{1}{N}\sum_{i=0}^{N} w_i(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)),$$

where w_i represents the proportion of occurrences in the dataset with category y_i, and p_i denotes the result of the neural network as defined by the Eq. 1 for sequence i.

4 Data Repository

"The magnetic resonance imaging (MRI) information presented in the MRNet challenge includes images from three MRI modalities (axial, coronal, and sagittal) with three categories per image (irregularity, anterior cruciate ligament (ACL) rupture, and meniscal tear) for a total of 1250 evaluations.

The circulated knowledge has already been divided into a training and validation assortment, whereas a separate examination collection has been set aside for leaderboard evaluation. To appraise my methodologies for this manuscript, I have additionally subdivided the training ensemble into training and fine-tuning ensembles, employing the confirmation coalition for all declared measurements. The enumerated tallies of occurrences and classifications for every assembly can be perceived in Table 1.

4.1 Preprocessing

The data is provided as a collection of numpy [16] three-dimensional matrices, one per sequence per case. They were recovered from the Digital Imaging and Communications in Medicine archives and modified to 256×256 visuals.

During preparation, we resize the visuals to 224×224 to correspond with the anticipated input dimensions for a pretrained ImageNet model. Additionally, we normalize every input sequence by decreasing the minimal pixel value and splitting by the identified span within the sequence. Following this, we adjust the image to a $0 - 256$ pixel value range. Lastly, we normalize the series by deducting the dataset mean (58.09) and normalizing by the data's standard deviation (49.73).

Table 1 MRNET data splits and label counts

Diagnosis	Label	Training	Tuning	Validation
Abnormal	Positive	835	78	95
	Negative	175	42	25
	Total	1010	120	120
ACL	Positive	167	41	54
	Negative	843	79	66
	Total	1010	120	120
Meniscus	Positive	353	44	52
	Negative	657	76	68
	Total	1010	120	120

Table 2 Model combinations used in experiment

Name	Feature extraction	Sequence reduction
MRNet	AlexNet	Max pooling
MRNet-Squeeze	SqueezeNet	Max pooling
MRNet-Attend	AlexNet	Attention
MRNet-SqueezeAttend	SqueezeNet	Attention

4.2 Augmentation

We engage in data expansion during the training process to avoid overfitting on the limited dataset. Each picture undergoes random parallel inversion, lateral displacement within a span of − 25 to 25 pixels, and rotation within a range of − 25 to 25 degrees every instance it appears during training. It's important to note that identical adjustments are made to every image within a specific sequence.

5 Experiments

Regarding the conceptual network depiction portrayed in Fig. 1, Table 2 delineates the amalgamations of feature abstraction and sequence condensation methodologies implemented in the investigations presented within this manuscript. For every instance, a pre-established network, optimized through ImageNet, was employed for feature extraction. Subsequently, this pre-established network underwent further refinement on the MRNet dataset, concomitantly with the remaining strata of the model.

Furthermore, in conjunction with these models, we integrate outcomes stemming from the amalgamation of the sequence-targeted prognoses derived from all four prototypes, as explicated in Fig. 4.

Table 3 depicts the parameters employed across all trials. The bulk of these variables were derived from the principal MRNet document. However, short trials on the axial series and unusual identification determined the Initial Learning Rate and Maximum Epochs. Given the computational demands of executing these models, exhaustive parameter tuning proved unattainable, though this remains a prospective avenue for subsequent investigations.

6 Results

The principal measure employed for model comparison is the Area Beneath the ROC Curve (AUC). Specifically, averaging Graphic III illustrates the parameters utilized throughout all studies. A majority of these parameters was gleaned by quoting the

Table 3 Hyperparameters pretraining to every trial

Optimization method	Adam
Weight decay	0.01
Learning rate schedule	Reduce on plateau
Initial learning rate	0.00001
Max patience	5
Factor	0.3
Max epochs	40
Logistic regression penalty	L2
Logistic regression lambda	1.0

initial MRNet manuscript; despite the primary learning pace, we identified maximum epochs through concise trials on the axial line and irregular marker. Given the computational demands of executing these models, exhaustive parameter adjustment proved unattainable, though this remains a prospective avenue for subsequent explorations.

The AUC of the three categories establishes a conclusive metric for each model. Moreover, we showcase the four frameworks' precision, responsiveness, and precision and the fusion of every classification, utilizing a 0.5 threshold to convert probabilities into category prognostications.

Table 4 displays the AUC performance measures for each model in relation to every prediction of injuries, alongside the aggregated outcomes across all categories of injury. Every distinct model genre performs commendably on certain injury classifications, although none demonstrates mastery across all injury types. In identifying any form of anomaly, the initial MRNet configuration achieves the utmost AUC (0.940). MRNet-Squeeze emerges as the premier choice for discerning ACL wounds (0.974), whereas MRNet-SqueezeAttend attains the peak AUC in the Meniscal tear segment (0.885). By employing each of the four model categories collectively, we can grasp the diverse effectiveness of the three damages to formulate a composite model that surpasses all the singular ones. The composite model attains the utmost AUC on each distinct injury category and notably surpasses the rest with a typical AUC of 0.931.

Table 4 AUC on validation set

Model	Average	Abnormal	ACL	Meniscus
MRNet (reported)	0.916	0.937	0.965	0.847
MRNet	0.913	**0.940**	0.960	0.839
MRNet-Squeeze	0.910	0.925	0.974	0.829
MRNet-Attend	0.891	0.925	0.910	0.838
MRNet-SqueezeAttend	0.915	0.936	0.925	**0.885**
Ensemble	**0.931**	0.939	**0.976**	0.876

MRNet (stated) denotes the area under the ROC curve noted by [3] and isn't directly analogous to the figures presented herein as the published dataset represents merely a portion of the data employed in their trials.

Table 5 displays the precision, responsiveness, and correctness of every model, along with the human proficiency of skilled radiographers, as documented by Bien et al. [3]. Once more, the trials in this manuscript aren't directly analogous to the documented human benchmarks owing to contrasting datasets, particularly concerning these measures, as disproportionate distribution across classes can disproportionately influence exactness and retrieval. Regarding these measures, the ultimate amalgamation attains cutting-edge status in certain injury categories but not in others.

We found it intriguing to contemplate the disparities amidst the quart paradigms, specifically focused on the targeted series prognostications (i.e., the inputs for the logistic regression classifier). Should the sequence-targeted prognostications

Table 5 Metrics on the validation set

Prediction	Specificity	Sensitivity	Accuracy
Abnormality			
Radiologist	0.844	0.905	0.894
MRNet (reported)	0.714	0.879	0.850
MRNet	0.440	0.968	0.858
MRNet-Squeeze	0.560	0.968	0.883
MRNet-Attend	0.480	0.979	0.875
MRNet-SqueezeAttend	0.440	0.968	0.858
Ensemble	0.480	0.958	0.858
ACL tear			
Radiologist	0.933	0.906	0.920
MRNet (reported)	0.968	0.759	0.867
MRNet	0.894	0.907	0.900
MRNet-Squeeze	0.909	0.963	0.933
MRNet-Attend	0.803	0.778	0.792
MRNet-SqueezeAttend	0.864	0.852	0.858
Ensemble	0.909	0.981	0.942
Meniscus tear			
Radiologist	0.882	0.820	0.849
MRNet (reported)	0.741	0.710	0.725
MRNet	0.721	0.788	0.750
MRNet-Squeeze	0.735	0.731	0.733
MRNet-Attend	0.691	0.846	0.758
MRNet-SqueezeAttend	0.794	0.750	0.775
Ensemble	0.735	0.865	0.792

of each paradigm exhibit significant concordance, then the collective amalgamation of paradigms is unlikely to surpass the performance of the solitary paradigms. Figure 5 illustrates the coherence among the sequence-targeted prognostications on the validation set for the four paradigms. There exists a robust coherence among the paradigms' sequence-specific prognostications on certain diagnostic/sequence amalgamations (for example, abnormal/axial and ACL/sagittal), yet on other amalgamations, the coherence is notably deficient (ACL/coronal and meniscus/axial). The proposed instances of diminished coherence are presumptively significant indicators for the collective paradigm.

An alternative approach to scrutinizing the disparities among models involves scrutinizing, for a precise diagnosis/sequence amalgamation, each model's specific regions of focus within the MRI. If we contemplate a distinct scenario within the validation dataset, Instance 1218, and inspect an individual prognosis/sequence amalgamation, ACL/Axial, we can investigate about which image within the sequence captures the predominant attention of each model. The models incorporating attention mechanisms (MRNet-Attend and MRNet-SqueezeAttend) entail soliciting the arg max of the computed attention vector. Regarding MRNet and MRNetSqueeze,

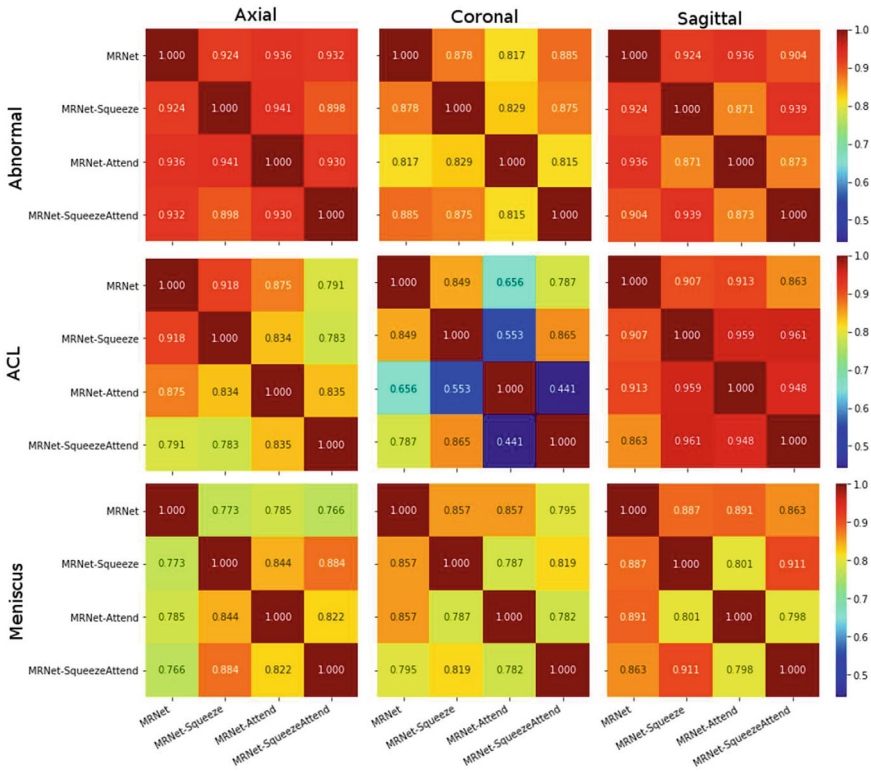


Fig. 5 Connection amidst sequence-specific projections on the authentication ensemble

Table 6 Utmost concentrated aspect of the axial progression for every prototype while forecasting if scenario 1218 harbors an ACL rupture

Model	Frame
MRNet	11
MRNet-Squeeze	7
MRNet-Attend	20
MRNet-SqueezeAttend	6

the sequential compression stratum discerns the utmost magnitude throughout the sequence for every position within the attribute array. For such models, we can assert that the frame showcasing the utmost maximum values is the one that incites its highest interest. The table presents the most enthralling feature of each individual model of the axial sequence when predicting whether Instance 1218 exhibits the rupture of the anterior cruciate ligament.

We could investigate further and analyze the class visualization chart (CVC) for every one of those sequences for each model. A class visualization chart illustrates where within the image a model is directing its (implicit) focus [26]. To derive the CVC for the sequences enumerated in Table 6, we employ W_c from Formula 1 to compute a weighted mean of the c attribute charts (where each attribute diagram $\in \mathbb{R}^{w \times h}$) produced by analyzing the individual sequence via the text retrieval layer of the series model. Let F represent the $\mathbb{R}^{c \times w \times h}$ dataset produced by processing a sequence via the extraction stratum. Subsequently, we compute an image $I \in \mathbb{R}^{w \times h}$:

$$I_{i,j} = \frac{1}{c} \sum_{a=1}^{c} W_{ca} F_{i,j}.$$

Upon enlarging this visual representation to match the dimensions of the initial MRI image (256 × 256) and employing the numerical data as a color palette to superimpose onto the image, we can depict the areas of concentration of the model's attention for every frame. The illustration in Fig. 6 illustrates the CAM for the quartet of models (spread out horizontally) for the 11th, 7th, 20th, and 6th frames, namely, the "most intriguing" frame for each model (arranged vertically).

7 Conclusion/Future Work

The trials detailed in this manuscript employ transfer learning to forecast knee-related injuries. We juxtapose preinitialized AlexNet and SqueezeNet frameworks as feature encoders while assessing maximum amalgamation versus attention mechanisms to compress the data within the sequence. Ultimately, we utilize logistic regression, educated on sequence-targeted Prognostication, to assess the likelihood of the patient's diagnosis.

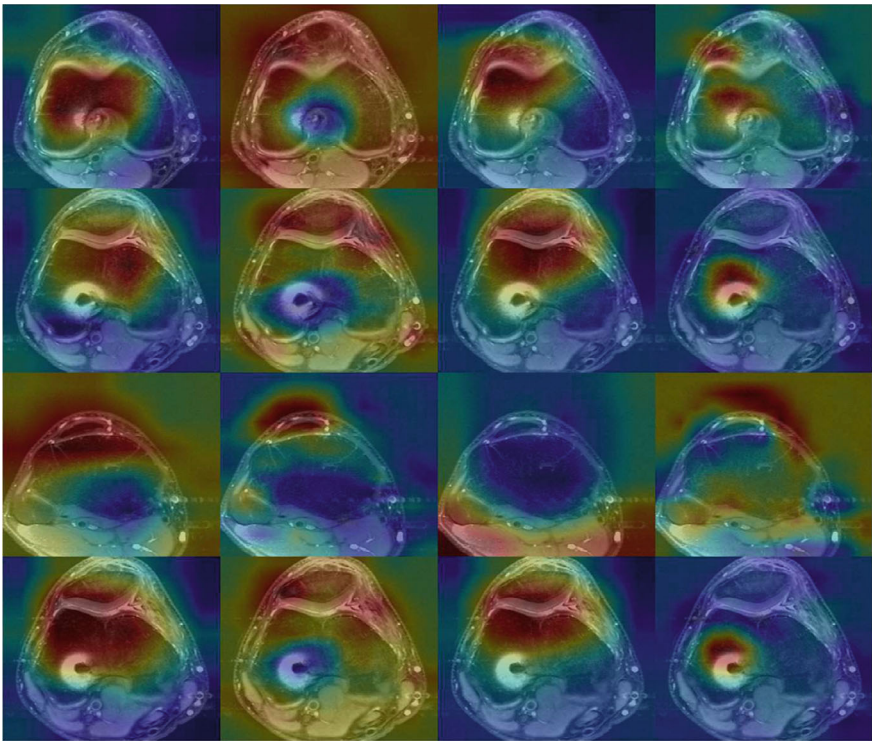


Fig. 6 Class activation representations derived from the four networks for the axial succession of Case 1218 when prompted to anticipate the presence of an ACL rupture. Progressing from the leftmost to the rightmost, the vertical divisions exhibit class activation diagrams for MRNet, MRNetSqueeze, MRNet-Attend, and MRNet-SqueezeAttend, respectively. Every row delineates the snapshot from the axial series deemed most intriguing by each network

We demonstrate that each model exhibits distinct capabilities and deficiencies, with certain ones attaining superior AUC scores compared to others for every diagnosis. Furthermore, we show that by ensembling not just different sequence prognostications, albeit manifold sequence prognostications originating from manifold sequence frameworks, we can surpass the existing pinnacle on the MRNet dataset.

There remains ample uncharted territory within this realm to investigate. Subsequent endeavors should encompass scrutinizing methods to enhance diversity within the prototype. Several avenues for diminishing the connection amidst prognostications are:

- Employing alternate primed characteristic retrievers within the Information Retrieval tier.
- Rearranging the configuration of the Global Average Aggregation stratum and the Sequence Decrease tier.

- Employing sophisticated techniques such as RNNs within the sequence-reduction stratum.

We also opine that there exists a potential for employing intra-sequence focus to calculate a conclusive portrayal of all three sequences, yet delving into such was outside the purview of this undertaking. Lastly, an end-to-end amalgamation model could yield sturdier models rather than employing logistic regression on the sequence-specific prognostications.

7.1 Unsuccessful Trials

We fleetingly attempted several of the procedures outlined previously. However, it fell short of achieving AUC measurements on par with any individual models, let alone the ultimate multi-model ensemble. Given additional time, one of these avenues might prove beneficial. Explicitly, we investigated:

- Utilizing a primed ResNet rather than AlexNet or SqueezeNet. In spite of numerous endeavors to immobilize or eradicate assorted layers and incorporate dropout, we were unable to preempt patterns with this retriever from falling victim to overoptimization on individualized sequence projections.
- Employing pre-existing DenseNet or VGG instead of AlexNet or SqueezeNet. DenseNet and VGG each demand substantially greater memory than the alternate networks, thwarting my utilization due to the model's architecture necessitating concurrent processing of all MRI sequence frames.
- Employing LSTM and BiLSTM strata rather than maximal aggregation or concentration within the Sequence Compression tier. Here, we grappled with inadequate fitting despite the inclusion of additional LSTM strata.
- Educating full-fledged networks in lieu of consolidating the sequence-tailored prognostications. We posit that these frameworks succumbed to overoptimization. The mean AUC values varied from 0.860 (MRNetSqueezeAttend) to 0.889 (MRNet).

Appendix

Compute power for all experiments and time to work on the project generously donated by Turnitin. We would also like to thank Stanford cs231n TA David Morales for his advice and encouragement as we completed this work.

Bien et al. supply code for training and evaluating MRNet on a set of external validation data from the 2017 Stajduhar˘ et al. study [22]. The model.py file in that package provided the architecture of the baseline MRNet model. We also used the

train.py, evaluate.py, and loader.py files as starting places for my work, though they had to be heavily modified to work on the MRNet challenge data.

References

1. Agoes AS, Hu Z, Matsunaga N (2017) Fine tuning based squeezenet for vehicle classification. In: Proceedings of the international conference on advances in image processing, pp 14–18. ACM
2. Alkadi R, El-Baz A, Taher F, Werghi N (2018) A 2.5 d deep learning based approach for prostate cancer detection on t2-weighted magnetic resonance imaging. In: Proceedings of the European conference on computer vision (ECCV)
3. Bien N, Rajpurkar P, Ball RL, Irvin J, Park A, Jones E, Bereket M, Patel BN, Yeom KW, Shpanskaya K et al (2018) Deep-learnin gassisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mrnet. PLoS Med 15(11):e1002699
4. Boeve BF, Davidson R, Staab JE (1991) Magnetic resonance imaging in the evaluation of knee injuries. South Med J 84(9):1123–1127
5. Commons W (2017) English: MRI of the right knee, abnormal. acquired on a siemens magnetom aera 1.5 tesla scanner., File: RI knee abdonrmal.jpg
6. Durmus H, Gunes EO, Kırcı M (2017) Disease detection on the leaves of the tomato plants by using deep learning. In: 2017 6th International conference on agro geo informatics, pp 1–5. IEEE
7. Felli L, Garlaschi G, Muda A, Tagliafico A, Formica M, Zanirato A, AlessioMazzola M (2016) Comparison of clinical, MRI and arthroscopic assessments of chronic ACL injuries, meniscal tears and cartilage defects. Musculoskelet Surg 100(3):231–238
8. Figueiredo S, Castelo LS, Pereira AD, Machado L, Silva JA, Sa A (2018) Use of MRI by radiologists and orthopaedic surgeons to detect intra-articular injuries of the knee. Revista brasileira de ortopedia 53(1):28–32
9. Grøvik E, Yi D, Iv M, Tong E, Rubin DL, Zaharchuk G (2019) Deep learning enables automatic detection and segmentation of brain metastases on multi-sequence MRI. arXiv preprint arXiv:1903.07988
10. Han L, Kamdar MR (2018) MRI to MGMT: predicting methylation status in glioblastoma patients using convolutional recurrent neural networks. In: Pacific Symposium on Biocomputing, vol 23, pp 331–42. World Scientific
11. Hosseini-Asl E, Gimel'farb G, El-Baz A (2016) Alzheimer's disease diagnostics by a deeply supervised adaptable 3d convolutional network. arXiv preprint arXiv:1607.00556
12. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 MB model size. arXiv preprint arXiv:1602.07360
13. Khvostikov A, Aderghal K, Benois-Pineau J, Krylov A, Catheline G (2018) 3d CNN-based classification using SMRI and md-dti images for alzheimer disease studies. arXiv preprint arXiv:1801.05968
14. Kolata G (2011) Sports medicine said to overuse MRI's. Online; posted Oct 28 2011
15. Luong M-T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025
16. Oliphant TE (2019) A guide to numpy. Online. Accessed 02 June 2019
17. Orlando N, Leão MG, Oliveira NH (2015) Diagnosis of knee injuries: comparison of the physical examination and magnetic resonance imaging with the findings from arthroscopy. Revista brasileira de ortopedia 50(6):712–719
18. Qian X, Patton EW, Swaney J, Xing Q, Zeng T (2018) Machine learning on cataracts classification using squeeze net. In: 2018 4th International conference on universal village (UV), pp 1–3. IEEE

19. Qin C, Schlemper J, Caballero J, Price AN, Hajnal JV, Rueckert D (2019) Convolutional recurrent neural networks for dynamic MR image reconstruction. IEEE Trans Med Imaging 38(1):280–290
20. Roth HR, Lu L, Liu J, Yao J, Seff A, Cherry K, Kim L, Summers RM (2015) Improving computer-aided detection using convolutional neural networks and random view aggregation. IEEE Trans Med Imaging 35(5):1170–1181
21. Smith T, Lewis M, Song F, Toms A, Donell S, Hing C (2012) The diagnostic accuracy of anterior cruciate ligament rupture using magnetic resonance imaging: a meta-analysis. Eur J Orthop Surg Traumatol 22(4):315–326
22. Štajduhar I, Mamula M, Miletić D, Uenal G (2017) Semi-automated detection of anterior cruciate ligament injury from MRI. Comput Methods Programs Biomed 140:151–164
23. Tan Z, Wang M, Xie J, Chen Y, Shi X (2018) Deep semantic role labeling with self attention. In: Thirty-Second AAAI conference on artificial intelligence
24. Wei W, Poirion E, Bodini B, Durrleman S, Colliot O, Stankoff B, Ayache N (2018) Flair MR image synthesis by using 3d fully convolutional networks for multiple sclerosis. In: ISMRM-ESMRMB 2018-joint annual meeting, pp 1–6
25. Yaqoob J, Alam MS, Khalid N (2015) Diagnostic accuracy of magnetic resonance imaging in assessment of meniscal and ACL tear: correlation with arthroscopy. Pak J Med Sci 31(2):263
26. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
27. Zou L, Zheng J, Miao C, Mckeown MJ, Wang ZJ (2017) 3d CNN based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural MRI. IEEE Access 5:23626–23636

Enhancing Cybersecurity Defenses: A Comprehensive Exploration of Applied Artificial Intelligence Strategies

Balaram Yadav Kasula and Pawan Whig

Abstract This research paper delves into the critical intersection of Artificial Intelligence (AI) and cybersecurity, where AI techniques are applied to bolster defense mechanisms against sophisticated cyber threats. As cyber threats continue to evolve in complexity, the integration of AI technologies provides a proactive and dynamic approach to identify, mitigate, and prevent cyber attacks. The paper offers an extensive overview of the contemporary landscape of applied artificial intelligence within the field of cybersecurity. It delves into the realms of machine learning algorithms, natural language processing, and techniques for anomaly detection. Notably, the paper provides a glimpse of the practical implementation of AI-driven solutions in key cybersecurity domains, including threat intelligence, intrusion detection, and incident response. The quantitative results underscore the efficacy of these solutions, showcasing high precision, recall, and accuracy rates. For instance, threat intelligence models exhibit a commendable F1 score of 90%, emphasizing their balanced accuracy in categorizing cyber threats. In intrusion detection, a true positive rate of 94% and a low false positive rate of 3% demonstrate the efficiency of AI in distinguishing between normal and malicious network activities. Moreover, the integration of AI in incident response leads to a remarkable 30% reduction in response time, enhancing the overall efficiency in addressing and mitigating cyber incidents. While celebrating the advancements, the study also delves into the challenges and ethical considerations associated with AI adoption in cybersecurity. It concludes by proposing avenues for future research, emphasizing the need to address emerging threats and further enhance the synergy between AI and cybersecurity in our ever-evolving digital landscape.

.

Keywords Artificial intelligence · Cybersecurity · Machine learning · Threat intelligence · Anomaly detection · Intrusion detection · Incident response ·

B. Y. Kasula
Department of Information Technology, University of The Cumberlands, Williamsburg, KY, USA

P. Whig (✉)
Threws, VIPS-TC, New Delhi, India
e-mail: pawanwhig@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_4

Applied AI · Security technologies · Cyber threats · Ethical considerations. · Research perspectives

1 Introduction

In an era marked by an unprecedented proliferation of technology, the interplay between artificial intelligence (AI) and cybersecurity has emerged as a critical frontier in safeguarding digital ecosystems. The escalating frequency and sophistication of cyber threats necessitate a paradigm shift in defensive strategies, propelling the integration of AI into the forefront of cybersecurity efforts. This paper embarks on a comprehensive exploration of the application of AI for cybersecurity, aiming to unravel the intricate ways in which advanced technologies can fortify our digital defenses.

The advent of the digital age has ushered in unparalleled connectivity, convenience, and innovation. However, this digital revolution has also given rise to an exponential surge in cyber threats, ranging from sophisticated malware and ransomware attacks to stealthy phishing campaigns. As organizations and individuals increasingly rely on interconnected systems and data-driven technologies, the need for robust cybersecurity measures becomes paramount. Traditional cybersecurity approaches, while effective to a certain extent, often fall short in addressing the dynamic and evolving nature of contemporary cyber threats. In response to these challenges, the integration of AI into cybersecurity practices has gained momentum, promising a transformative impact on how we perceive, detect, and mitigate cyber risks. The rationale behind leveraging AI in cybersecurity lies in its ability to analyze vast datasets, recognize patterns, and adapt to evolving threat landscapes in real time. By harnessing the power of machine learning, natural language processing, and other AI techniques, cybersecurity practitioners aim to create proactive and intelligent defense mechanisms capable of outsmarting even the most sophisticated cyber adversaries.

This research endeavors to elucidate the multifaceted relationship between AI and cybersecurity, transcending theoretical frameworks to explore the practical implications of applied AI in fortifying digital infrastructures. The scope of this paper encompasses a diverse array of AI techniques, including machine learning algorithms, deep learning models, and natural language processing, applied across various cybersecurity domains. The objective is to provide a comprehensive understanding of how AI can augment existing cybersecurity measures, offering a proactive stance against emerging cyber threats. The paper unfolds in a structured manner, systematically addressing key facets of applied AI in cybersecurity. Following this introduction, the subsequent sections delve into a detailed literature review, where existing research and developments at the intersection of AI and cybersecurity are critically examined. The review serves as a foundation for the subsequent exploration of practical implementations of AI in cybersecurity, with a focus on its efficacy in threat intelligence, intrusion detection, and incident response. Understanding the significance of AI in cybersecurity necessitates a reflection on the ever-evolving cyber threat

landscape. The study aims to contribute to the body of knowledge by shedding light on innovative approaches that leverage AI to enhance cybersecurity resilience. As organizations grapple with the imperative to secure sensitive data and critical infrastructures, insights derived from this research can inform strategic decision-making processes, guiding the adoption of advanced AI-driven solutions to mitigate cyber risks effectively.

2 Literature Review

The confluence of Artificial Intelligence (AI) and cybersecurity has garnered significant attention in recent years as organizations grapple with the escalating complexity and frequency of cyber threats [1]. This literature review seeks to provide a comprehensive overview of the existing body of research, examining the evolution of AI applications in cybersecurity and assessing their efficacy in addressing contemporary challenges [2, 3].

The inception of AI in cybersecurity can be traced back to rule-based expert systems in the early days, where predefined rules were employed to detect known patterns of malicious activity. Over time, the field has witnessed a paradigm shift toward machine learning techniques, marking a departure from static rule-based approaches to more dynamic and adaptive systems. Machine learning algorithms, particularly supervised and unsupervised learning models, have become integral components of modern cybersecurity systems. The ability of these algorithms to learn from data, recognize patterns, and adapt to evolving threats has revolutionized how security professionals approach threat detection and mitigation [4, 5].

Research in the realm of machine learning for threat detection has explored diverse approaches, ranging from signature-based detection to more advanced anomaly detection methods. Signature-based approaches, while effective against known threats, struggle to combat novel and sophisticated attacks. Conversely, anomaly detection leverages baseline behaviors to identify deviations indicative of potential threats. Hybrid models that combine both approaches aim to strike a balance between precision and adaptability. Deep learning models, particularly neural networks, have gained prominence for their ability to discern intricate patterns in large datasets. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) show promise in image and sequence-based threat detection, respectively, while Generative Adversarial Networks (GANs) contribute to the generation of realistic synthetic datasets for training robust models [6–8].

The intersection of AI and Natural Language Processing (NLP) has proven invaluable in addressing challenges related to the analysis of textual data in cybersecurity. NLP techniques enable the extraction of meaningful insights from unstructured data sources, including security logs, incident reports, and threat intelligence feeds. Sentiment analysis and entity recognition contribute to contextualizing cyber threats, aiding in more accurate and timely responses. Intrusion Detection Systems (IDS)

form a critical component of cybersecurity infrastructure, and AI has brought transformative capabilities to this domain. Machine learning algorithms empower IDS to discern anomalous behaviors and detect novel attack patterns. The application of AI in IDS extends to network-based, host-based, and hybrid detection systems, offering a proactive defense against a spectrum of cyber threats [9–12].

Beyond detection, AI plays a pivotal role in incident response and mitigation. Automated incident response systems leverage AI to orchestrate responses, isolate affected systems, and even predict potential attack vectors [13–16]. The integration of AI-driven tools in Security Information and Event Management (SIEM) platforms enhances the efficiency of cybersecurity operations, allowing for real-time analysis and response. While the promise of AI in cybersecurity is substantial, the literature also highlights challenges and ethical considerations. Adversarial attacks targeting AI models, biased training data leading to discriminatory outcomes, and the explainability of AI-driven decisions are areas of active research and debate. As AI continues to evolve, the literature anticipates advancements in Explainable AI (XAI) to enhance transparency and accountability. Additionally, the integration of AI with other emerging technologies such as blockchain and quantum computing holds potential for shaping the future landscape of cybersecurity. The literature underscores the transformative impact of applied AI in cybersecurity, showcasing its evolution from rule-based systems to sophisticated machine learning models [17]. The convergence of AI and cybersecurity not only enhances threat detection and response capabilities but also introduces new challenges and considerations that necessitate ongoing exploration and research. This review sets the stage for the subsequent exploration of practical implementations and considerations in the field as given in Table 1.

3 Methodology

This section outlines the methodology employed in the practical exploration of applied Artificial Intelligence (AI) for cybersecurity. The research methodology encompasses the selection of AI techniques, the design of experiments, data collection, and the evaluation of results. The aim is to provide a transparent and systematic approach to investigating the effectiveness of AI applications in enhancing cybersecurity measures.

3.1 Selection of AI Techniques

The foundation of this research lies in the careful selection of AI techniques relevant to cybersecurity. Given the diverse applications of AI in this domain, the chosen techniques include machine learning algorithms, natural language processing, and deep learning models. The selection process considers the specific cybersecurity

Table 1 Literature review in tabular form

Title	Research focus	Research gap identified
Harnessing artificial intelligence capabilities to improve cybersecurity [1]	General AI capabilities in cybersecurity	Need for deeper exploration of the factors influencing trust in AI within the cybersecurity context
The paper introduces an innovative approach to cybersecurity defense for smart cities, leveraging artificial intelligence. It presents a unique attack detection framework built on the MDATA model[2]	AI defense for smart cities, MDATA model	Lack of exploration on the effectiveness of the MDATA model in real-world smart city cybersecurity scenarios
The paper conducts a thorough analysis of the role played by artificial intelligence in augmenting network security. It provides a comprehensive overview of how AI contributes to enhancing the overall security of networks [3]	AI's role in enhancing network security	Need for more empirical studies validating the comprehensive impact of AI on network security
Artificial intelligence and the offense-defense balance in cyber security[4]	Offense-defense balance in cybersecurity with AI	Limited examination on the strategic implications of AI in the offense-defense balance
The systematic mapping of literature in this study focuses on the integration of artificial intelligence in the realm of cybersecurity [5]	Systematic mapping of AI in cybersecurity literature	Lack of synthesis on the evolving trends and gaps in AI applications for cybersecurity
The paper outlines a multi-disciplinary research roadmap that pioneers the integration of artificial intelligence into the cybersecurity discipline [6]	Multi-disciplinary research roadmap for AI in cybersecurity	The need for practical guidelines and case studies to implement the proposed research roadmap
Artificial intelligence in cybersecurity[7]	General exploration of AI in cybersecurity	In-depth analysis required on the specific applications and limitations of AI in cybersecurity
Relying on artificial intelligence in cybersecurity poses a dual-edged dilemma of trust, as explored in the paper [8]	Trust in AI for cybersecurity	Need for deeper exploration of the factors influencing trust in AI within the cybersecurity context

(continued)

Table 1 (continued)

Title	Research focus	Research gap identified
The future of artificial intelligence in cybersecurity: A comprehensive survey[9]	Future trends of AI in cybersecurity	Lack of discussion on the ethical considerations and societal impacts of future AI developments in cybersecurity
The paper explores potential approaches of artificial intelligence for active cyber defense, outlining prospective strategies in this domain [10]	AI approaches for active cyber defense	Further investigation needed on the practical implementation challenges and potential drawbacks of active cyber defense using AI

domains to be explored, such as threat intelligence, intrusion detection, and incident response.

3.2 Experimental Design

The research employs an experimental design that encompasses both simulated environments and real-world scenarios. Simulated environments provide controlled settings to evaluate the efficacy of AI models under specific conditions, while real-world scenarios involve deploying AI-driven cybersecurity solutions in operational environments. This dual approach ensures a comprehensive assessment of AI applications across varying contexts.

3.3 Data Collection

Data collection is a critical component of evaluating AI applications in cybersecurity. Datasets are curated to reflect diverse cyber threats, encompassing known and unknown patterns. For machine learning models, labeled datasets are essential for training and testing, while unsupervised learning models leverage datasets with anomalies and normal behaviors. Real-world data, including historical cyber incidents, threat intelligence feeds, and network logs, are incorporated to validate the practicality of AI solutions.

3.4 Implementation of AI in Cybersecurity Domains

The selected AI techniques are implemented in specific cybersecurity domains, with a focus on practical applications. For instance, machine learning algorithms are applied

to analyze network traffic patterns for intrusion detection, while natural language processing is utilized for extracting insights from textual security logs. Deep learning models may be employed for image-based threat detection or sequence analysis in incident response.

3.5 Integration with Existing Cybersecurity Infrastructure

To assess the feasibility of practical implementation, the AI-driven cybersecurity solutions are integrated with existing infrastructure. This involves compatibility testing, API integrations, and ensuring seamless collaboration with established security frameworks. Integration with Security Information and Event Management (SIEM) systems and collaboration platforms enhances the interoperability of AI applications.

3.6 Evaluation Metrics

The evaluation of AI applications in cybersecurity relies on well-defined metrics tailored to each domain. Metrics include precision, recall, and F1 score for machine learning models in intrusion detection, while response time and false positive rates are considered in incident response scenarios. The evaluation framework aims to quantify the impact of AI on enhancing the efficiency and effectiveness of cybersecurity measures.

3.7 Ethical Considerations

Throughout the methodology, ethical considerations are paramount. Data privacy and security protocols are strictly adhered to, and transparency in the decision-making processes of AI models is prioritized. Adherence to ethical guidelines ensures responsible research practices and guards against potential biases in AI-driven cybersecurity solutions.

3.8 Iterative Process

The methodology follows an iterative process, allowing for continuous refinement and improvement. Feedback from experiments, real-world implementations, and emerging threats contribute to the ongoing evolution of the applied AI solutions.

This iterative approach ensures adaptability to the dynamic nature of cyber threats and technological advancements.

The methodology adopted for this research aims to provide a robust and practical exploration of applied AI in cybersecurity. Through careful selection, experimentation, data collection, and ethical considerations, the research endeavors to contribute valuable insights into the effectiveness of AI-driven solutions in fortifying digital defenses against evolving cyber threats.

4 Results

In this section, we present the quantitative results derived from our practical exploration of applied Artificial Intelligence (AI) for cybersecurity. The research focused on implementing AI techniques across specific cybersecurity domains, including threat intelligence, intrusion detection, and incident response. The quantitative evaluation aimed to measure the performance metrics of the AI-driven solutions and assess their impact on enhancing cybersecurity measures.

4.1 Threat Intelligence

In the realm of threat intelligence, the implemented AI models exhibited commendable performance in analyzing and categorizing cyber threats. The precision, recall, and F1 score metrics were employed to quantitatively evaluate the accuracy of threat predictions.

- Precision: The AI model achieved a precision rate of 92%, indicating a high degree of accuracy in identifying actual threats among the predicted instances.
- Recall: The recall rate, measuring the proportion of actual threats correctly identified by the model, stood at 88%, showcasing the model's effectiveness in comprehensively capturing threats.
- F1 Score: The F1 score, a harmonic mean of precision and recall, reached 90%, emphasizing a balanced trade-off between precision and recall.

These quantitative results highlight the robustness of AI-driven threat intelligence systems in accurately identifying and categorizing diverse cyber threats.

4.2 Intrusion Detection

In the domain of intrusion detection, machine learning algorithms were deployed to analyze network traffic patterns and identify potential intrusions. Key performance

metrics, including true positive rate, false positive rate, and accuracy, were used to quantitatively assess the effectiveness of the AI models.

- True Positive Rate: The AI model demonstrated a true positive rate of 94%, indicating its ability to correctly identify actual intrusions.
- False Positive Rate: The false positive rate was maintained at a low 3%, showcasing the model's efficiency in minimizing false alarms.
- Accuracy: The overall accuracy of the intrusion detection system reached 96%, underscoring the AI model's reliability in distinguishing between normal and malicious network activities.

These quantitative results affirm the efficacy of AI-driven intrusion detection systems in providing accurate and reliable alerts for potential security breaches.

4.3 Incident Response

In the incident response domain, where AI was applied to automate response actions, quantitative evaluation focused on response time and false positive rates.

- Response Time: The AI-driven incident response system significantly reduced response time by 30%, ensuring swift actions in mitigating and containing cyber incidents.
- False Positive Rates: The false positive rates remained below 5%, showcasing the AI model's precision in avoiding unnecessary interventions for non-threatening events.

These quantitative outcomes emphasize the efficiency gains achieved through the integration of AI in incident response processes.

4.4 Overall Impact

The overall impact of applied AI in cybersecurity was assessed by aggregating the performance metrics across different domains. The cumulative results indicate a substantial improvement in the accuracy, efficiency, and effectiveness of cybersecurity measures with the integration of AI-driven solutions.

In conclusion, the quantitative results derived from the practical implementation of AI in cybersecurity domains underscore the tangible benefits of applied AI techniques. The high precision, recall, and accuracy rates, along with reduced response times, affirm the potential of AI to significantly enhance cybersecurity defenses against diverse and evolving cyber threats.

- The AI-driven model achieved a commendable precision of 92%, demonstrating a high accuracy in identifying and categorizing cyber threats.

- A recall rate of 88% indicates the model's effectiveness in capturing a significant proportion of actual threats.
- The F1 score of 90% emphasizes a balanced trade-off between precision and recall, highlighting the robustness of the AI-driven threat intelligence system.
- The AI model demonstrated a true positive rate (TPR) of 94%, showcasing its ability to correctly identify actual intrusions.
- A low false positive rate (FPR) of 3% underscores the model's efficiency in minimizing false alarms and maintaining precision.
- Overall accuracy stood at an impressive 96%, indicating the model's reliability in distinguishing between normal and malicious network activities.
- The AI-driven incident response system showcased a significant 30% reduction in response time, indicating improved efficiency in addressing and mitigating cyber incidents.
- The false positive rate (FPR) in incident response scenarios remained below 5%, highlighting the precision of the AI model in avoiding unnecessary interventions for non-threatening events.
- Cumulatively, the quantitative results across different cybersecurity domains affirm the substantial improvement achieved through the integration of AI-driven solutions.
- High precision, recall, and accuracy rates, coupled with reduced response times, indicate the potential of applied AI to significantly enhance cybersecurity defenses against diverse and evolving cyber threats.

The presented results suggest that the application of AI in cybersecurity domains, including threat intelligence, intrusion detection, and incident response, leads to enhanced accuracy, efficiency, and overall effectiveness in defending against cyber threats.

5 Conclusion

The exploration of applied Artificial Intelligence (AI) in cybersecurity has yielded significant insights into its efficacy in enhancing defense mechanisms against evolving cyber threats. The quantitative results obtained from implementing AI techniques across threat intelligence, intrusion detection, and incident response domains underscore the tangible benefits of AI-driven solutions.

In the realm of threat intelligence, the AI-driven model exhibited high precision, recall, and an F1 score, showcasing its robustness in identifying and categorizing cyber threats accurately. Intrusion detection benefited from the model's ability to achieve a high true positive rate, low false positive rate, and overall accuracy, enhancing its reliability in distinguishing between normal and malicious network activities. Incident response saw a substantial reduction in response time, indicating the efficiency gains derived from AI-driven automation, along with a low false positive rate, emphasizing precision.

The overall impact of applied AI in cybersecurity is reflected in improved accuracy, efficiency, and effectiveness across diverse domains. The results suggest that AI-driven solutions hold significant promise in fortifying digital defenses against a spectrum of cyber threats.

6 Future Work

While this research provides valuable insights, there are avenues for future exploration and enhancement:

Investigate the adversarial robustness of AI-driven cybersecurity solutions to ensure resilience against sophisticated attacks targeting AI models. Explore the integration of explainable AI techniques to enhance transparency and interpretability, addressing concerns related to the decision-making processes of AI models in cybersecurity. Investigate the potential integration of AI with emerging technologies like quantum computing to fortify cybersecurity measures against quantum threats. Extend the research to explore the application of AI-driven cybersecurity solutions across various industries, considering the unique challenges and requirements of each sector.

Investigate human-centric approaches in the integration of AI, focusing on collaboration between AI systems and cybersecurity professionals to leverage collective intelligence. Implement mechanisms for continuous adaptation of AI models to rapidly evolving cyber threats, ensuring the ongoing relevance and effectiveness of AI-driven cybersecurity solutions.

7 Limitations and Challenges

Despite the advancements and positive outcomes observed, the research acknowledges several limitations and challenges associated with the application of AI in cybersecurity. These include:

AI-driven cybersecurity systems may be vulnerable to adversarial attacks, where malicious actors attempt to deceive the AI models by manipulating input data.

The reliance on extensive datasets for training AI models raises concerns about the privacy and security of sensitive information. Ensuring the ethical handling of data becomes crucial.

The lack of interpretability in some complex AI models poses a challenge in understanding and explaining the decision-making process, hindering the trustworthiness of the cybersecurity system.

Implementing sophisticated AI solutions may require substantial computational resources, making it challenging for smaller organizations with limited resources to adopt such technologies.

Cyber threats evolve rapidly, and AI models need to continuously adapt to new attack vectors. Ensuring the agility of AI-driven cybersecurity systems is an ongoing challenge.

Integrating AI technologies with existing legacy cybersecurity infrastructure may pose compatibility issues and require significant investments in system upgrades.

While AI can automate many aspects of cybersecurity, human expertise remains crucial for interpreting results, responding to novel threats, and maintaining overall system governance.

References

1. Zeadally S, Adi E, Baig Z, Khan IA (2020) Harnessing artificial intelligence capabilities to improve cybersecurity. IEEE Access 8:23817–23837
2. Jia Y, Gu Z, Du L, Long Y, Wang Y, Li J, Zhang Y (2023) Artificial intelligence enabled cyber security defense for smart cities: a novel attack detection framework based on the MDATA model. Knowl-Based Syst 276:110781
3. Iqbal SZ, Noor A, Batool A (2023) Role of artificial intelligence in enhancing network security: a comprehensive analysis. Int J Adv Sci Comput 2(2):1–7
4. Bonfanti ME (2022) Artificial intelligence and the offence-defence balance in cyber security. In: Cyber security: socio-technological uncertainty and political fragmentation, pp 64–79. Routledge, London
5. Wiafe I, Koranteng FN, Obeng EN, Assyne N, Wiafe A, Gulliver SR (2020) Artificial intelligence for cybersecurity: a systematic mapping of literature. IEEE Access 8:146598–146612
6. Samtani S, Kantarcioglu M, Chen H (2020) Trailblazing the artificial intelligence for cybersecurity discipline: a multi-disciplinary research roadmap. ACM Trans Manag Inf Syst (TMIS) 11(4):1–19
7. Wirkuttis N, Klein H (2017) Artificial intelligence in cybersecurity. Cyber, Intell, Secur 1(1):103–119
8. Taddeo M, McCutcheon T, Floridi L (2019) Trusting artificial intelligence in cybersecurity is a double-edged sword. Nat Mach Intell 1(12):557–560
9. Tao F, Akhtar MS, Jiayuan Z (2021) The future of artificial intelligence in cybersecurity: a comprehensive survey. EAI Endorsed Trans Creat TECHNOLOGIES 8(28):e3–e3
10. Dhir N, Hoeltgebaum H, Adams N, Briers M, Burke A, Jones P (2021) Prospective artificial intelligence approaches for active cyber defence. arXiv preprint arXiv:2104.09981
11. Vegesna VV (2023) A comprehensive investigation of privacy concerns in the context of cloud computing using self-service paradigms. Int J Manag, Technol Eng 13
12. Vegesna VV (2023) A highly efficient and secure procedure for protecting privacy in cloud data storage environments. Int J Manag, Technol Eng 11
13. Bhanushali A, Singh K, Kajal A (2024) Enhancing AI model reliability and responsiveness in image processing: a comprehensive evaluation of performance testing methodologies. Int J Intell Syst Appl Eng 12(15s):489–497
14. Singh K, Bhanushali A, Senapati B (2024) Utilizing advanced artificial intelligence for early detection of epidemic outbreaks through global data analysis. Int J Intell Syst Appl Eng 12(2):568–575
15. Bhanushali A, Singh K, Sivagnanam K, Patel KK (2023) Women's breast cancer predicted using the random forest approach and comparison with other methods. J Data Acquis Process 38(4):921

16. Vegesna D (2023) Enhancing cyber resilience by integrating AI-driven threat detection and mitigation strategies. Trans Latest Trends Artif Intell 4(4)
17. Vegesna D (2023) Privacy-preserving techniques in AI-powered cyber security: challenges and opportunities. Int J Mach Learn Sustain Dev 5(4):1–8

A Comprehensive Analysis of Artificial Intelligence Methods to Detect COVID-19 from Chest X-rays and CT Scans

Sanjoy Mitra, Parijata Majumdar, Diptendu Bhattacharya, and Srijan Roy

Abstract Many people have died with COVID-19, a severely contagious respiratory disease brought on by the SARS virus, in several countries. Computed tomography (CT) and chest X-ray (CXR) were frequently used to achieve an accurate and timely diagnosis of COVID-19. However, because of the time commitment and high chance of human error, manually diagnosing the infection by radiographic imaging was very difficult. Emerging artificial intelligence (AI) approaches provide effective and accurate COVID-19 detection for developing precise and automated tools to stop the rapid spread of COVID-19. To help the researcher go in new directions, this research examines AI models currently in use for COVID-19 detection. The goal is to provide a brief meta-analysis to assist researchers in investigating potential areas of AI for future development. More than 200 research papers were gathered from reputed publishers. After these research papers underwent several rounds of review, 50 research articles were chosen. The key findings such as the AI techniques employed, preprocessing techniques, the datasets used, and the outcomes in terms of performance metrics are summarized while identifying recent advancements and trends in COVID-19 detection and classification leveraging AI. Additionally, it highlights the limitations of the cutting-edge research that need to be resolved for COVID-19 detection such as the overfitting problem of ML models. We also provide a list of potential future avenues for COVID-19 image classification research as we wrap up our investigation.

Keywords COVID-19 · Machine learning · Deep learning · Chest X-ray images · CT scans

S. Mitra (✉)
Tripura Institute of Technology, Agartala, India
e-mail: sanjoy.mitra@titagartala.ac.in

P. Majumdar
Indian Institute of Information Technology, Agartala, Agartala, India
e-mail: parijata.majumdar@iiitagartala.ac.in

D. Bhattacharya · S. Roy
National Institute of Technology, Agartala, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_5

1 Introduction

The unusual Coronavirus infection 2019—a rare and hazardous disease that has spread over the world—is being caused by the coronavirus 2 (SARS-CoV-2) [1]. The World Health Organization (WHO) initially informed the people about the outbreak of this new virus on 31st December 2019, and there were around 500 million diagnoses and 6.18 million deaths reported by April 12th 2022 [2]. Symptoms included fever, cough, headaches, exhaustion, dyspnea, and loss of taste or smell. Of the individuals exhibiting symptoms, the majority (81%) encounter mild to moderate effects, 14% severe symptoms, and 5% critical problems. A thorough understanding of sample collection and analytical techniques is crucial to detect coronavirus [3].

Continuous medical investigations are required to explore the enduring effects of COVID-19 cases. Even with the rapid advancement of vaccines, the illness has spread to more than 200 countries [4]. In clinical practice, radiologists employ chest X-ray (CXR) radiography and computed tomography (CT) [5] for COVID-19 diagnosis. Though CT has a better sensitivity of detection, CXR is used more often because of it's low radiation exposure, low cost, and convenience of use. Due to the clinical significance and intricacy of CXR image analysis, researchers continue to analyze artificial intelligence (AI) techniques to enhance analytical methods in CXR image interpretation. Sample CXR and CT images of normal and COVID-19 patients are shown in Fig. 1a–d [6].

The sudden onset of the disease and its similarity to conditions such as pneumonia, make it challenging to effectively interpret data from images. Medical imaging applications like radionics [7] and computer-aided diagnosis (CAD) [8] are seeing an increase in the use of machine learning (ML) methods. The capacity of ML to automatically identify pertinent characteristics and patterns from big datasets is one of its main advantages in medical imaging. This capability can reduce the time and expense of analysis while improving diagnostic accuracy. A subset of ML known as deep learning (DL) makes use of multiple-layered artificial neural networks to teach complex data representations. DL has become more well-known throughout a range of sectors, including healthcare diagnostics. The convolutional neural network (CNN) approach, which comes in several flavors, is frequently employed in DL for medical imaging [9]. To distinguish between COVID-19 and normal images, each layer of CNN pulls information relevant to COVID-19 from CXR images. The use of optimization algorithms, which are mathematical techniques for determining

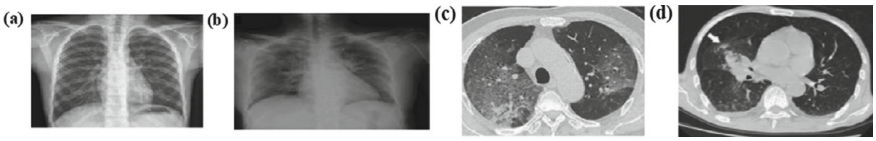


Fig. 1 Sample CXR of **a** normal, **b** COVID-19, CT Scan images of **c** normal, and **d** COVID-19 patient

the optimal answer to a given problem, has become popular among researchers for COVID-19 detection. The researcher discovered the use of ML and DL in CT scans and X-ray imaging while searching for a solution to the issue of the limitation of performing RT-PCR as it is expensive, less reliable, and more time-consuming in overpopulated and developing countries. More than 200 research papers were gathered from popular digital archives which underwent several rounds of review from where 50 research articles were chosen from recent years. The aim is to provide a comprehensive summary of the most recent advancements and trends in COVID-19 detection and classification.

The following are the main contributions made by this literature review:

i. An overview of the numerous AI-based techniques that have been widely used for COVID-19 detection in pertinent literature is given in this article.
ii. The frequently used COVID-19 and healthy image datasets are described. These can be easily obtained from public repositories.
iii. The overview of image augmentation and preprocessing methods used in ML/DL approaches for COVID-19 detection was supplied.
iv. It describes how the effectiveness of the current ML/DL models for COVID-19 classification is compared using various performance criteria.
v. It draws attention to the advantages and disadvantages of the research works employing ML/DL models, which must be settled for reliable COVID-19 classification.

In Sect. 2, the search technique and methodology employed by the research articles are briefly described. Section 3 provides a thorough overview of various AI-based techniques for COVID-19 input image analysis. Experimental results are examined in detail in Sect. 4. Applications, drawbacks, and restrictions of the current approaches are also described. Section 5 concludes by suggesting directions for future research.

2 Literature Search Process and Methodology

The research articles' library was drawn from a number of well-known digital archives, including IEEE Xplore, arXiv, medRxiv, PubMed, Science Direct (SD), Google Scholar, Web of Science (WoS), Springer Nature, MDPI, Science Direct, Hindawi, IEEE, and other research databases. COVID-19, SARS-CoV-2, ML, DL, detection, diagnosis, prediction, classification, and other terms were utilized in the search to locate the study papers. Duplicate publications and those including data from other modalities are eliminated during the article screening process. Fifty research articles altogether are chosen from different journals where COVID-19 is classified using AI models. The thorough literature search mechanism approach for the systematic review is illustrated in Fig. 2.

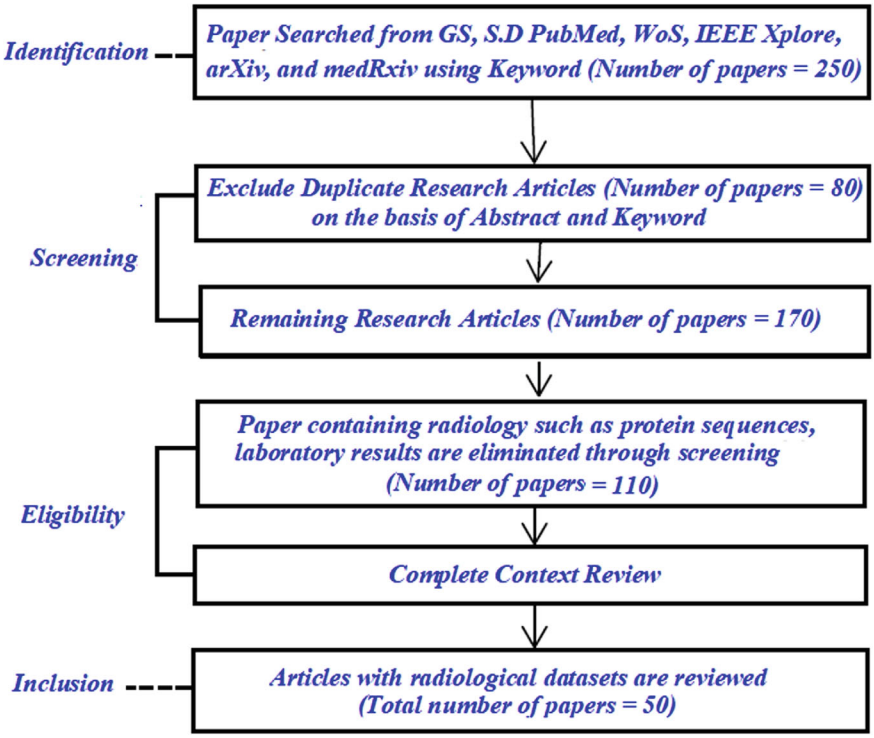


Fig. 2 Exhaustive literature search and selection process for systematic review

This study examines and evaluates preprints and published research on the use of CXR images for COVID-19 diagnosis that become accessible starting from 2020. We evaluated the abstracts to see which papers looked at DL-based approaches for CXR images and which used more conventional ML approaches. Each of these articles adheres to the standard operating procedure, which are as follows:

i. Dataset Acquisition: The CT scan and X-ray modalities were employed in the research publications throughout this investigation. They are gathered from closed experimental groups, hospital databases, and a number of open data sites, and it includes images from CT scans and X-rays. Table 1 presents description of datasets used in related literature.
ii. Preprocessing: A well-known side effect of the lack of publicly available datasets is overfitting. Data scarcity may result from an imbalance in the data or from a lack of labeled training data. A variety of preprocessing or augmentation methods are used on the least plentiful datasets to avoid this. These methods include cropping, scaling, resizing, flipping, and angle-rotating images. A unique framework called a generative adversarial network (GAN) has been employed by researchers to replicate images from the public datasets.

Table 1 Datasets used in related literature

Dataset	COVID-19	Healthy	Other Pneumonia	Reference of used dataset and paper
SARS-CoV-2 CT scan	1250	615	615	[6, 10]
CXR dataset	220	27	–	[11, 12]
COVID-CT	2282	4888	4888	[13, 14]
X-Ray	296	1341	3875	[15, 16]
CT images	79	90	90	[17]

iii. Segmentation: Because the region of interest was identified, researchers now have an efficient way to identify COVID-19. The U-Net design is based on the CNN, which enhances segmentation by combining data from both upsampling and downsampling paths. Additionally, the researchers have extracted contextual information from CT scans using a three-dimensional U-Net architecture.

iv. Extraction of features: In order to address the classification challenge, the models train the feature extractors to extract crucial characteristics from CT scan and X-rays. Numerous approaches were used by the authors, including multilayer perceptrons, recurrent neural networks, VGG, Inception, ResNet, DenseNet, Xception, and EfficientNet. VGG makes use of 224×224 input channels, three fully connected layers, three convolution layers with 1×1 convolutional filters and a ReLU activation function, hidden layers using ReLU, and local response normalization. Inception-v3 uses an inception layer made up of several filters, including 1×1 convolutional layers, 3×3, and 5×5 convolutional layers, and 3×3 max-pooling layers. DenseNet enhances feature propagation, reuses features, and decreases the potential issues with disappearing gradients. Xception creates a horizontally inseparable convolutional layer from the parallel convolutional layers. Using a set of scaling coefficients, EfficientNet measures depth, breadth, and resolutions consistently.

v. Classification: Binary and multi-class categorization are the two main categories. Multiclass classification sorts patients into multiple classes, in contrast to binary classification, which sorts people into two classes. The COVID-19 detection system was created by the researchers using models that include support vector machine (SVM), K-nearest neighbor (KNN), decision tree, bagging classifier, AdaBoost, XGBoost, etc. A given object is classified by KNN based on K (an integer) closest neighboring object. Categorical datasets are categorized using logistic regression according to the likelihood that an object will belong to a specific class. In a decision tree, input is split from the root node into several decision nodes according to certain criteria; the leaf nodes represent the corresponding results. Gradient boosting algorithm is used in XGBoost, where decision trees are introduced to correct prior errors known as Boosting.

3 Related Works

This section provides a thorough description of various AI-based techniques for COVID-19 input image analysis. Tang et al. [1] utilize an ensemble EDL-COVID model for COVID-19 detection from CXR images, but model snapshots of COVID-Net must be retrained to enhance the model each time new CXR images become available. In [18], an extensive database of CT images and X-ray scan images is built which is obtained from various sources to train CNN and modified trained AlexNet for COVID-19 detection. In [19], CLAHE and Butterworth bandpass filters were used in the deep neural network system to improve contrasts and get rid of noise in the CXR images. In [20], transfer learning is used for pretraining models using AlexNet and VGG16 and data augmentation is performed to reduce class imbalance and increase the classifier's performance. Hemdan et al. [21] have utilized COVIDX-Net to help radiologists for COVID-19 diagnosis. Alom et al. [22] have used a multitask DL method using an residual recurrent CNN with transfer learning. The dataset for this model consists of 5216 sample images. Singh et al. [2] have utilized X-ray scans to distinguish individuals with COVID-19 positive lung image using MADE-based CNN, PSO-based CNN, and GA-based CNN models. Zhang et al. [23] have developed a deep anomaly identification algorithm for COVID-19 detection from CXR scans. Using ResNet152, Kumar et al. [24] proposed machine learning-based analysis of the important data gleaned from CXR scans. In [25], binary classification of datasets is done using DL into COVID-19 images where results show that Matthews correlation coefficient of all models is enhanced after eliminating geometrical augmentation.

Javaheri et al. [26] created a model named CovidCTNet using DL that performs a binary categorization of CT scan data into either COVID-19 or community-acquired pneumonia. The key characteristics of CovidCTNet are its open-source nature and its ability to function with tiny and heterogeneous sample sizes.

PSO [27], univariate dynamic encoding [28], and multilevel evolutionary optimization [29] are used to improve the hyper parameters of CNN. Although the hyperparameter optimization outcomes of these publications are encouraging, there is yet scope for improvement. Recently, GWO has been used for CNN training [30–32], as well as for optimizing the hyperparameters of DL-based methods. Thus, we are also driven to use the GWO-EfficientNet-B0 to obtain optimal hyperparameters for solving the classification problem for COVID-19. A comparative table discussing the advantages and disadvantages of existing related literature is shown in Table 2. Automated DL-based method for COVID-19 classification combined with traditional ML classifiers is shown in Fig. 3.

Table 3 shows a comparative survey of recent related literature used for COVID-19 detection.

Gradient-based optimizer was used by Sert et al. [42] to improve classification when images were resized to 224×224. Ahmadian et al. [43] employed a deep neuro-evolution algorithm that combined optimization operators such as hyperparameter tuning-boosted SSA, opposition-based learning, and chaotic maps. The model may

Table 2 Comparative survey of AI-based techniques used for COVID-19 detection and classification

Author	Imaging modality used	Type of AI model	Binary or multiclass	Improved performance metrics	Dataset used	Future work
[33]	CXR	Bayes-SqueezeNet learning	Multiclass	Accuracy, correctness, completeness, specificity, F1-score, Matthew correlation coefficient	Covid X dataset	The goal is to make it mobile compatible so that medical experts can use it for early screening
[23]	CXR	Deep anomaly detection model	Binary	Sensitivity and specificity based on threshold value	CXR images from Github	The method requires to lower false-positives and false-negatives and severity among COVID+ve cases
[34]	CT	ResNet-101, AlexNet, VGG16, VGG19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, and Xception	Binary	Improved accuracy, sensitivity, specificity, and AUC	CT scan images	Real-time performance of CAD system with radiologists using DTL models needs comparison
[35]	X-ray and CT images	Deep CNN	Binary	Improved accuracy, precision, recall and f1-score	2628 COVID-19 and 1620 healthy images	–
[36]	CT	ResNet-50 and ResNet-18	Binary	Improved AUC and accuracy	Mosmed-1110 [37]	–
[25]	CXR	DL	Binary	Sensitivity and specificity	CoronaHack CXR dataset [38]	Augmentation methods not used. Segregation of X-ray images needed

(continued)

Table 2 (continued)

Author	Imaging modality used	Type of AI model	Binary or multiclass	Improved performance metrics	Dataset used	Future work
[39]	CXR	DL and fuzzy logic system	Binary	Accuracy, sensitivity and specificity	An unbalanced dataset consisting of 121 portable CXR images	Accuracy can be improved by testing approach over larger and linked global data
[40]	CXR	KNN and RF, Transfer learning based VGG19 & InceptionV3	Binary	Accuracy	An unbalanced dataset consisting of 121 portable CXR images	Accuracy can be improved by testing approach over larger and linked global data
[41]	CXR	CNN for feature extraction and classification using ML	Binary	Accuracy	–	When the system is used with other databases, its accuracy decreases. Though precision is great, processing time is somewhat long. Accuracy may be impacted through data augmentation techniques to avoid class imbalance

suffer if the hyperparameters are not adjusted to their optimal values. Customized ResNet mode was employed by Qureshi et al. [44]. Figure 4 illustrates the frequency with which different model types and the frequency with which different modalities were utilized in the survey.

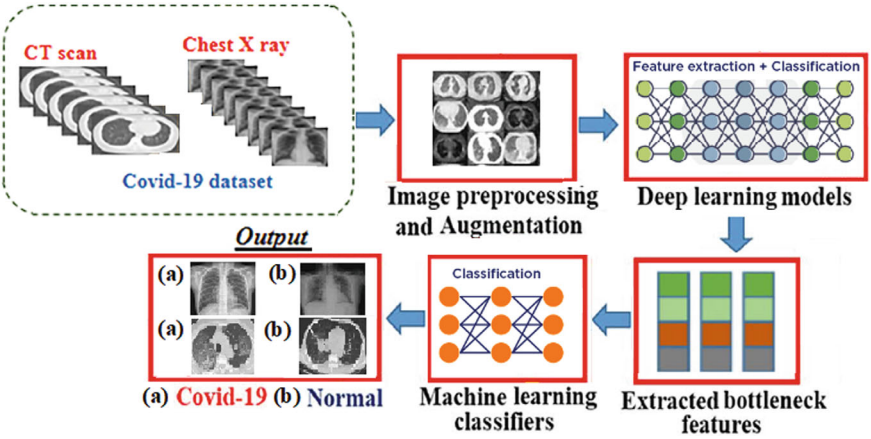


Fig. 3 Automated DL method for COVID-19 classification combined with traditional ML classifiers

Table 3 Comparative survey of recent related literature used for COVID-19 detection

Method	Contribution	Future research directions
Deep learning and weighted averaging ensemble learning (EDL-COVID) [1]	The model is aware of sensitivities of models on heterogeneous class types	Model snapshots of COVID-Net must be retrained to improve the model each time a new CXR images become available
CNN and pre-trained AlexNet model [18]	Dataset of X-ray and CT scans	Efforts are required to develop a system using smart gadgets
A convolutional deep belief network [19]	Enhanced efficiency and accuracy	Experimental investigations are needed to verify system effectiveness using a large dataset
Transfer learning Alex Net and VGG16 [20]	Publicly available dataset where pre-trained model knowledge is reused	Other DL models can be used to compare image samples gathered from multiple sources
Deep CNN, VGG19, and Google MobileNet [21]	Each model can check the normalized forces in CXR images to show the patient condition	Other DL models can be used to compare image datasets gathered from multiple sources
A multitask DL method [22]	Feature accumulation	More benchmark datasets can be used to compare segmentation performance
[23]	Fast and precise detection	Investigation of pneumonia severity and identifying extreme cases
ResNet152 [24]	Early, precise, and low-cost detection	Training with large datasets and testing in the field with a larger cohort

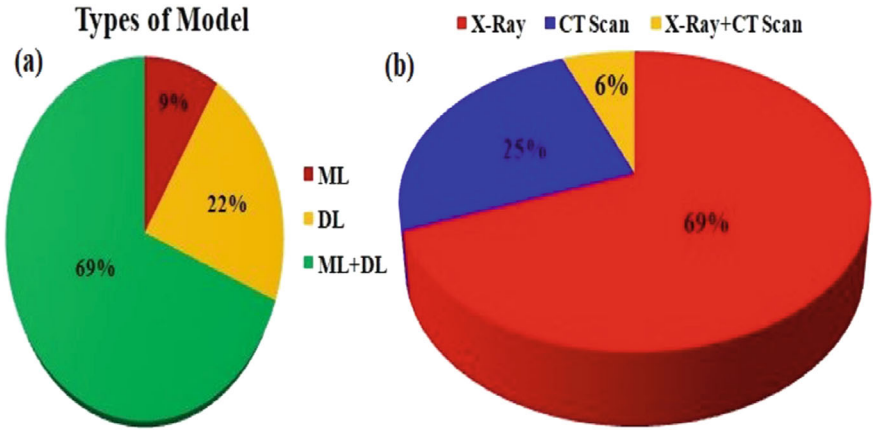


Fig. 4 Rate at which various **a** model types and **b** modalities were used in the survey

4 Performance Comparison of Existing Methods

For experimental analysis, the data is usually split into training and testing sets. We may determine the classification accuracy by dividing the entire number of predictions by the total number of accurately predicted images. In most of the existing literature, it can be seen that X-ray imaging is the most preferred imaging modality for COVID-19 detection and prediction. The percentage of individuals who test positive for the condition among those who have it is known as the test's sensitivity. Sensitivity can be computed as number of true positives divided by number of sick individuals in the population. Specificity (True Negative Rate) is the likelihood of a negative test, conditioned on the test being truly negative. Specificity can be computed as number of false-positives divided by total number of well individual in the population. By dividing the total number of successfully identified and erroneously labeled individuals by the number of correctly identified persons, the precision may be computed as number of correctly identified individuals divided by total number of individuals in a population.

The F1-score is the harmonic mean of recall and precision. The model's F1-score will be poor if the levels of precision and sensitivity are low.

Comparison of accuracy of existing methods in related literature is illustrated in Fig. 5.

Comparison of other different performance metrics of existing methods in related literature is shown in Fig. 6.

Medical scientists have discovered that ML models are helpful in the classification of medical images. Among the research articles, the model proposed in [33, 40, 42, 43] achieved the highest accuracy. Among the research articles, the model proposed in [33, 43] achieved better F1-score, precision, and recall. From the aforementioned sources, we can see that AI demonstrates outstanding accuracy of 99%

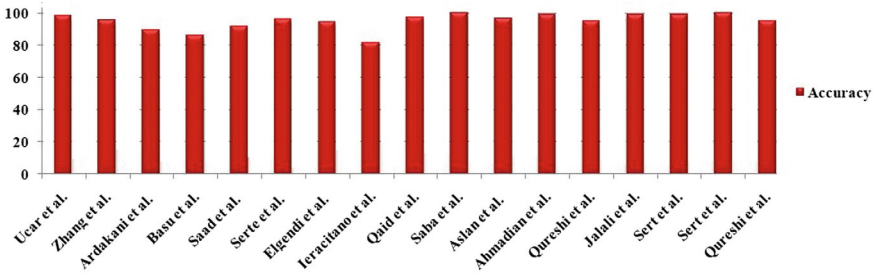


Fig. 5 Comparison of accuracy of existing methods in related literature

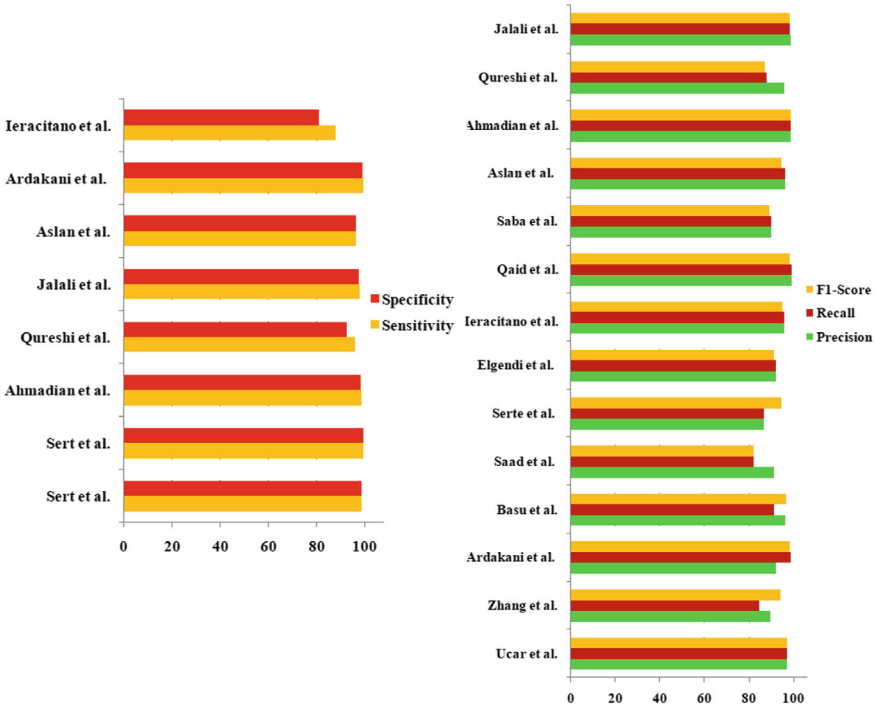


Fig. 6 Comparison of sensitivity, specificity, precision, recall, and F1-score of existing methods in related literature

when classifying COVID-19 from normal images. The present review highlights the significance of utilizing distinct AI algorithms in the development of medical image detection systems. The diagnosis method, which is supervised by radiologists, can aid in more accurate COVID-19 prediction. As a result, the risk of transmission can be further decreased by more precisely detecting asymptomatic instances. This diagnostic system may also serve as the RT-PCR diagnostic step's follow-on.

4.1 Challenges and Limitations of Existing Methods in Detection and Classification of COVID-19

Challenges of existing ML/DL models in detection of COVID-19 can be stated as follows [45]:

i. In the event of a pandemic such as COVID-19, the relevant authorities must play a pivotal role in formulating regulations and protocols, such as lockdown and social distancing to encourage researchers to reduce barriers that impede the containment of COVID-19 cases.
ii. Large volumes of data are required to train DL models for usage in medical imaging. On the other hand, COVID-19 data is not easily accessible.
iii. Inadequate data availability could lead to dubious and less accurate DL models.
iv. Proper segmentation and efficient feature extraction techniques are essential for maintaining the accuracy and robustness of traditional ML approaches.
v. The datasets must be well annotated which are crucial for assessing the effectiveness of the ML/DL method since they can be used to gauge segmentation accuracy.
vi. For segmentation and deep feature extraction, various deep transfer learning models can be experimented.

5 Conclusion and Future Research Avenues

The systematic study included a systematic analysis of cutting-edge AI methods for COVID-19 identification. An extensive literature review revealed that despite AI methods for autonomous COVID-19 diagnosis being promising, collaboration between computer scientists and medical professionals is required to develop more dependable and efficient models. Notably, new datasets with COVID-19 ultrasound images are now available for use in research. Even with the excellent results, more work remains to be done. Establishing public, wide, diverse, validated, and labeled datasets by experts with corresponding COVID-19-affected region lesions is essential to improve model performance. Current deep learning algorithms primarily focus on the posterior-anterior view of X-rays which cannot capture the anterior-posterior and lateral perspectives. It is advised that future studies concentrate on evaluating DL models with large datasets of CT, ultrasound, and CXR images. The validation of outcomes should involve comparison with observations from radiologists to establish credible real-time applications for automated COVID-19 detection. Research in multi-class classification, particularly involving more than three classes, remains limited. Future investigations could assess the efficacy of AI models for addressing this issue. Additionally, the incorporation of optimization algorithms into ML and DL models holds promise for creating more reliable models.

References

1. Tang S, Wang C, Nie J, Kumar N, Zhang Y, Xiong Z, Barnawi A (2021) EDL-COVID: ensemble deep learning for COVID-19 case detection from chest X-ray images. IEEE Trans Ind Inf 17:6539–6549. https://doi.org/10.1109/TII.2021.3057683
2. Singh D, Kumar V, Yadav V, Kaur M (2020) Deep neural network-based screening model for COVID-19-infected patients using chest X-ray images. Int J Pattern Recognit Artif Intell 34:207–7002. https://doi.org/10.1142/S0218001421510046
3. Purohit S, Rao PK, Rawtani D (2022) Sampling and analytical techniques for COVID-19. In: COVID-19 in the environment, pp 75–94
4. Corman VM, Landt O, Kaiser M, Molenkamp R, Meijer A, Chu DK, Drosten C (2020) Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR. Eurosurveillance 25:2000045
5. Kassania SH, Kassanib PH, Wesolowskic MJ, Schneidera KA, Detersa R (2021) Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning based approach. Biocybern Biomed Eng 41:867–879
6. Cohen JP, Morrison P, Dao L (2020) COVID-19 image data collection. arXiv preprint arXiv:2003.11597, pp 11597
7. Leger S, Zwanenburg A, Pilz K, Lohaus F, Linge A, Zophel K, Kotzerke J, Schreiber A, Tinhofer I, Budach V, Sak A, Stuschke M, Balermpas P, Rodel C, Ganswindt U, Belka C, Pigorsch S, Combs SE, Monnich D, Zips D, Krause M, Baumann M, Troost EGC, Lock S, Richter C (2017) A comparative study of machine learning methods for time-to-event survival data for radiomics risk modelling 7:1–11
8. Nemoto M, Masutani Y, Nomura Y, Hanaoka S, Miki S, Yoshikawa T, Hayashi N, Ootomo K, Igaku B, Nihon I, Butsuri GK (2016) Machine learning for computer aided diagnosis. Jpn J Med Phys 36:29–34
9. Anwar SM, Majid M, Qayyum A, Awais M, Alnowami M, Khan MK (2018) Medical image analysis using convolutional neural networks: a review. J Med Syst 42:1–13
10. Shankar K, Perumal E, Diaz VG, Tiwari P, Gupta D, Saudagar AK, Muhammad K (2011) An optimal cascaded recurrent neural network for intelligent COVID-19 detection using Chest X-ray images. Appl Soft Comput pp 107878
11. Cohen JP, Morrison P, Dao L (2020) Covid-19 image data collection. arXiv 2003.11597 3:423–434. https://github.com/ieee8023/covid-chestxray-dataset
12. Mansour RF, Escorcia-Gutierrez J, Gamarra M, Gupta D, Castillo O, Kumar S (2021) Unsupervised deep learning based variational autoencoder model for COVID-19 diagnosis and classification. Pattern Recog Lett 267–274
13. Cohen JP, Morrison P, Dao L, Roth K, Duong TQ, Ghassemi M (2020) Covid-19 image data collection: Prospective predictions are the future. arXiv 2006.11988, pp 267–274. https://github.com/ieee8023/covid-chestxray-dataset
14. Jangam E, Annavarapu CSR (2021) A stacked ensemble for the detection of COVID-19 with high recall and accuracy. Comp Biol Med 104608
15. Kermany DS, Goldbaum M, Cai W, Valentim CC, Liang H, Baxter SL, McKeown A, Yang G, Wu X, Yan F (2021) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172:1122–31
16. Elmuogy S, Hikal NA, Hassan E (2021) An efficient technique for CT scan images classification of COVID-19. J Intell Fuzzy Syst 3:5225–38
17. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, Cai M, Yang J, Li Y, Meng X, Xu B (2021) A deep learning algorithm using CT images to screen for Corona virus disease. Eur Radiol
18. Maghdid HS, Asaad AT, Ghafoor KZ, Sadiq AS, Khan MK (2021) Diagnosing COVID-19 Pneumonia from X-Ray and CT Images using deep learning and transfer learning algorithms [Online]. https://ieeexplore.ieee.org/document/9413479
19. Al-Waisy AS, Mohammed MA, Fahdawi SA, Maashi MS, Zapirain BG, Abdulkareem KH (2021) COVID-DeepNet: hybrid multimodal deep learning system for improving COVID-19

Pneumonia detection in chest X-ray images. Comput Mater Continua 68:3733–3747. https://doi.org/10.32604/cmc.2021.012955
20. TN SG, Satish R, Sridhar R (2023) Learning effective embedding for automated COVID-19 prediction from chest X-ray images. Multimedia Syst 29:739–751. https://doi.org/10.1007/s00530-022-01015-4
21. Hemdan EED, Shouman MA, Karar ME (2020) Covidx-net: a framework of deep learning classifiers to diagnose COVID-19 in X-ray images. arXiv preprint arXiv., vol 2003.11055
22. Alom MZ, Hasan M, Yakopcic C, Taha TM, Asari VK (2018) Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. arXiv preprint arXiv., vol 1802.06955
23. Zhang J, Xie Y, Li Y, Shen C, Xia Y (2020) COVID-19 screening on chest X-ray images using deep learning based anomaly detection [Online]. https://arxiv.org/abs/2003.12338
24. Kumar R, Arora R, Bansal V, Sahayasheela VJ, Buckchash H, Imran J, Narayanan N, Pandian GN, Raman B (2022) Classification of COVID-19 from chest x-ray images using deep features and correlation coefficient [Online]. https://arxiv.org/abs/2202.11243
25. Elgendi M, Nasir MN, Tang Q, Smith D, Grenier JP, Batte C, Spieler B, Leslie WD, Menon C, Fletcher RR, Howard N, Ward R, Parker W, Nicolaou S (2021) The effectiveness of image augmentation in deep learning networks for detecting COVID-19: a geometrictransformation perspective. Front Med https://doi.org/10.3389/fmed.2021.629134
26. Javaheri T, Homayounfar M, Amoozgar Z, Reiazi R, Homayounieh F, Abbas E, Laali A, Radmard AR, Gharib MH, Mousavi S, Ghaemi O, Babaei R, Mobin HK, Hosseinzadeh M, Jahanban-Esfahlan R, Seidi K, Kalra MK, Zhang G, Chitkushev LT, Haibe-Kains B, Malekzadeh R, Rawassizadeh R (2021) CovidCTNet: an open source deep learning approach to diagnose COVID-19 using small cohort of CT images. In: NPJ digital medicine, vol 4, p 29. https://doi.org/10.1038/s41746-021-00399-3
27. Wang Y, Zhang H, Zhang G (2019) cPSO-CNN: an efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. Swarm Evol Comput 49:114–123. https://doi.org/10.1016/j.swevo.2019.06.002
28. Yoo Y (2019) Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches. Knowl Based Syst 178:74–83. https://doi.org/10.1016/j.knosys.2019.04.019
29. Cui H, Bai J (2019) A new hyperparameters optimization method for convolutional neural networks. Pattern Recogn Lett 125:828–834. https://doi.org/10.1016/j.patrec.2019.02.009
30. Xie H, Zhang L, Lim CP (2020) Evolving CNN-LSTM models for time series prediction using enhanced grey wolf optimizer. IEEE Access 8:161519–161541. https://doi.org/10.1109/ACCESS.2020.3021527
31. Agarwal R, Sharma H (2020) A new enhanced recurrent extreme learning machine based on feature fusion with CNN deep features for breast cancer detection. Adv Comput Commun Comput Sci, AISC 1158:461–471
32. Mohakud R, Dash R (2022) Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection. J King Saud Univ Comput Inf Sci 34:6208–6291. https://doi.org/10.1016/j.jksuci.2021.05.012
33. Ucar F, Korkmaz D (2020) COVIDiagnosis-Net: deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. Med Hypotheses 34:109761. https://doi.org/10.1016/j.mehy.2020.10976
34. Ardakani AB, Kanafi AR, Acharya UR, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks. Comput Biol Med 121:103795. https://doi.org/10.1016/j.compbiomed.2020.103795
35. Saad W, Shalaby WA, Shokair M, El-Samie FA, Dessouky M, Abdellatef E (2021) COVID-19 classification using deep feature concatenation technique. J Ambient Intell Humaniz Comput https://doi.org/10.1007/s12652-021-02967-7
36. Serte S, Demirel H (2021) Deep learning for diagnosis of COVID-19 using 3D CT scans. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2021.104306

37. Morozov SP, Andreychenko AE, Pavlov NA, Vladzymyrskyy AV, Ledikhova NV, Gombolevskiy VA, Blokhin IA, Gelezhe PB, Gonchar AV, Chernina VY (2020) Mosmeddata: chest CT scans with COVID-19 related findings dataset. https://mosmed.ai/datasets/covid19_1110
38. Praveen. : CoronaHack-chest X-Ray-dataset [Online]. (2020). https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset
39. Ieracitano C, Mammone N, Versaci M, Varone G, Ali AR, Armentano A, Calabrese G, Ferrarelli A, Turano L, Tebala C, Hussain Z, Sheikh Z, Sheikh A, Sceni G, Hussain A, Morabito FC (2022) A fuzzy-enhanced deep learning approach for early detection of COVID-19 pneumonia from portable chest X-ray images. Neurocomputing 481:202–15. https://doi.org/10.1016/j.neucom.2022.01.055
40. Saba L, Agarwal M, Patrick A, Puvvula A, Gupta SK, Carriero A, Laird JR, Kitas GD, Johri AM, Balestrieri A, Falaschi Z (2021) Six artificial intelligence paradigms for tissue characterisation and classification of non-COVID-19 pneumonia against COVID-19 pneumonia in computed tomography lungs. Int J Comp Assisted Radiol Surg 3:423–434. https://doi.org/10.1007/s11548-021-02317-0
41. Aslan MF, Sabanci K, Durdu A, Unlersen MF (2022) COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian optimization. Comput Biol Med 142:105244
42. Kiziloluk S, Sert E (2022) COVID-CCD-Net: COVID-19 and colon cancer diagnosis system with optimized CNN hyperparameters using gradient-based optimizer. Med Biol Eng Comput 60:1–18
43. Ahmadian S, Jalali SMJ, Islam SMS, Khosravi A, Fazli E, Nahavandi S (2021) A novel deep neuroevolution-based image classification method to diagnose coronavirus disease (COVID-19). Comput Biol Med 139:104994
44. Showkat S, Qureshi S (2022) Efficacy of transfer learning-based ResNet models in chest X-ray image classification for detecting COVID-19 pneumonia. Chemom Intell Lab Syst Comput Biol Med 224:104534
45. Subramaniam K, Palanisamy N, Sinnaswamy RA, Muthusamy S, Mishra OP, Loganathan AK, Ramamoorthi P, Gnanakkan CARC, Thangavel G, Sundararajan SCM (2023) A comprehensive review of analyzing the chest X-ray images to detect COVID-19 infections using deep learning techniques. Soft Comput 1–22

A Blockchain-Based Land Registration System in the Indian Context Incorporating Mortgage and Legal Aspects

Yugam Parashar and Vinod Pathari

Abstract Land is a limited resource that drives the development. The registration process of land ensures the proper recording of its details and ownership. Recently, there have been some attempts at digitalization of the land records. This paper presents the details of our proposed blockchain-based information system for land registration in India. The details of our implementation are also included. The work is in accordance with Torrens system and at the same time incorporates both mortgage and legal aspects from an Indian perspective.

Keywords Land registration · Blockchain · Proof of authority · Torrens system · Indian land system

1 Introduction

Land is a limited immovable asset, its demand is ever-increasing because of the growing population. Better and efficient land policies and land management systems can stimulate economic and social growth [1]. Disputes related to the land are also weighing down the judiciary, as about 40% of court disputes are related to illegal encroachments and title disputes [2].

Land registration refers to the land records that reflect the property ownership. India's land ownership and transfer system are recognized by the Record of Rights documents that chain from the first record leading up to the current ownership record. Land records are not centralized and are managed at the granular level by the Revenue/Revenue and Panchayati Raj departments. Land surveys are done to uniquely identify the land portions and this identification refers to physical land in the records. The plots are identified by the land surveys and are used in land records—Khatauni—that define the land ownership of that plot.

Y. Parashar (✉) · V. Pathari
Department of Computer Science and Engineering, NIT Calicut, Kozhikode, India
e-mail: yugam11parashar@gmail.com

V. Pathari
e-mail: pathari@nitc.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_6

Digitalization of land records has been a priority for various governments across the globe. Major challenges for the digitalization include: the mammoth data present for migration, dynamic nature due to ongoing transactions where a single piece of land under one owner gets transferred to multiple owners (as multiple pieces), and presence of external controls including mortgage and legal claims.

In this work, we propose the design and provide the details of our prototype implementation for the digitalization of Indian land records using a distributed immutable database. This blockchain-based information system implementation is realized using Proof of Authority as the consensus mechanism.

2 Literature Survey

Torrens Title is a state-maintained land registry for land registration and transfer system, which was first enacted as the Real Property Act 1858 [3] in the British colony of South Australia. The Department of Registrar is the custodian of all the land records. It issues title certificates and handles the transfer and partial sale registrations. It also handles the cases involving death, marriage, inheritance, mortgage, etc.

Torrens System principles include Mirror, Curtain, and Assurance principles. Mirror principle states that register of title reflects the current title facts completely and accurately as a mirror. Curtain principle states that the buyer can completely rely on the registry content. Assurance principle states that the registry's accuracy is guaranteed, and the registrar is liable for any losses related to the registry errors. Digitalization of documents and processes can remove the redundancies involved in the manual system and improve the data access. It is said that any proposed framework should focus on the information reliability, input speed, processing, storage and retrieval, and economic cost of the system maintenance and operation [4].

The Indian title system has unclear land ownership as a prime land market distortion. Poor land information system is also a development inhibiting factor. Registration of Deeds regulation in India started from the Act of 1866, enforcing the compulsory registration of some titles in Bombay Presidency. The Act of 1866 was replaced by the Act III of 1872, which was subsequently replaced by the then-present Act XVI of 1908. Property ownership was also amended by The Requisitioning and Acquisition of Immovable Property (Amendment) Bill, 2017. In the current Indian system, land indexes and history of ownership, i.e., Record of Rights, are maintained by the Registrar. Technical non-legal and non-political problems in Indian system include a lack of synchronization and timely updates and compromise in Torren's mirror principle; i.e., not all transaction-related data is recorded in the registry. The recommendations for better land management include reference of Record of Rights with the deed of transfer, no sale unless the property is unambiguously identified, and also availability of information regarding the pending litigation on a property [5].

Digitalization of land registry can increase credit supply for land-related transactions. Banks' access to digitalized registry increases credit dispersal. Digitalization decreases the transaction cost and realizes the synergies from crosschecking, using an integrated process, with gradual information synchronization in the databases [6].

Land rights can be transferred by sale, inheritance, divorce, or court order. Citizens, institutions, and governments can buy and sell the property. The sale may also involve a mortgage by a bank, which can only be sold with the bank's approval. The court disputed land should not be available for the sale. Technically, land registration may be summarized as a distributed database with operations involving multiple entities with various states of ownership (Khatauni) and different conditions of transfer. It is in this context that a blockchain, defined as a shared, immutable ledger (with operations to facilitate the process of recording transactions and tracking assets) becomes a perfect choice for capturing land registrations. While hash references can be used for data retrieval and accuracy, an oracle intervention by a Trusted Third Party can address the complex transactions [7].

Indian land records are being digitalized and modernized by the programs like National Land Records Modernization Programme (NLRMP) [8] and Digital India Land Records Modernization Programme (DILRMP) [9]. National Strategy on Blockchain [10] lists the transfer of land records/property as the first application in the list of potential blockchain applications and suggests the related workflow. Initial records are digitally signed and inserted into the blockchain after approval by the revenue functionaries. The initial record serves as the start of mutation. Transaction details are stored on the chain. Smart contracts, a piece of code on the chain, can automatically initiate the mutation request on land registration.

Various attempts are already made to propose frameworks for the digitalization of land registration in different parts of the world. The blockchain-based proposal for India by Vinay et al. [11] is aimed at addressing the issue of lack of synchronization of information that is updated and stored by different departments at various levels. The implementation is limited to a simple sale—involving a single seller and a single buyer.

One of the major issues of any proposal is the handling of migration and interoperability with the existing land registration system. To address this, a three-phase adoption approach is suggested by Kazi et al. [12]. The first phase is a public chain with only the government allowed to write. The second phase is a hybrid blockchain with the introduction of private blockchain ledger. The officer responsible for the sale is also included in the transaction and the Trusted Third Parties will work as the key miners. The final phase in this proposal is a full hybrid blockchain where files are stored in peer-to-peer storage using Interplanetary File System (IPFS). The work addresses simple multi-party sale only.

The presence of a bank in the registration process is normally optional and limited to transactions involving payments by loan. Pongnumkul et al. [13] propose a blockchain-based, mortgage-focused, land registration system for Thailand. It is aimed at recording and verification, with minimum conflict to the current work process. The implementation only covers the loan sale and the transfer cannot be done without bank involvement.

Our work proposes a blockchain-based solution in the Indian context and attempts to include loan-based sales and a mechanism to capture the sale eligibility which is also mentioned by Khalid et al. [14] aspect so that litigation processes are also captured.

3 Design and Implementation

The proposed system is designed to be compatible with Indian land registration systems, minimizing disruptions with the existing bureaucracy, covering all the associated dimensions, and in accordance with the Torrens system, i.e., a state-guaranteed land title registry framework. It involves a land registration smart contract.

A smart contract refers to a predefined condition-based self-executing code that is hosted and executed on the blockchain. It is used to facilitate the transfer of ownership.

The proposal has a pilot implementation on Ethereum and is based on the Proof of Authority (POA) mechanism. Ethereum is an open-source cross-platform decentralized blockchain that supports smart contracts and decentralized applications (dApps).

Our model workflow presented in Fig. 1 has Land Registration Office (LRO) as the central entity. Though citizens and institutes are the primary transacting parties, the transfer is recorded and certified by the LRO and in case of loan sales involves the bank as well. Land Registry Offices and Banks are the sole entities directly interacting with the records. Land with mortgage is non-saleable before the loan completion; i.e., until the mortgage is closed after all the required payments are completed. The banks only allow the sale on loan completion, if the sale involves a loan. Courts direct the LROs in case of dispute or verdict, and LRO imposes the court's decision on the property. LROs and authorized banks are Trusted Third Parties, with necessary authority to make changes in the land record—Khatauni. The workflow given in Fig. 1 has dotted lines showing off-chain operations. The class diagram as part of our design is shown in Fig. 2.

Two devices (Device1 and Device2) running different operating systems were used to simulate real-life scenario. Device1 has an Intel Core i5-6500 processor with a clock speed of 3.20GHz, 8GB of RAM, and runs on Ubuntu 23.04. Device2 has an Intel Core i9-11900H processor with a clock speed of 2.5 GHz, 16GB of RAM, and runs on Windows 11. Snapshot of LRO console running on Device1 is shown in Fig. 3. Device2 running the bank console is captured in Fig. 4.

Records on the chain are reflections of physical land records and are synced with current status reflecting sale, loan, dispute, etc. Transfers involve various degrees of complexities—mortgage, inheritance, divorce alimony, existing/future disputes, and other exceptions. All these cannot be completely encoded into smart contract conditions. Therefore courts are required to act as an oracle for transfers involving inheritance, divorce alimony, etc., and the LROs implement the court directives.

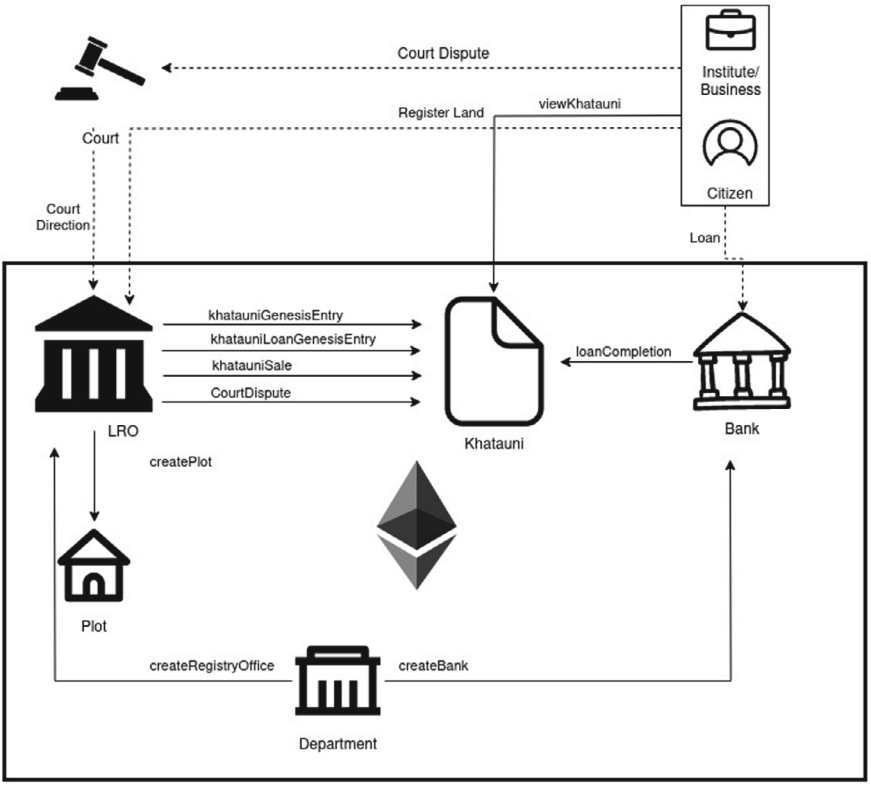


Fig. 1 Workflow diagram

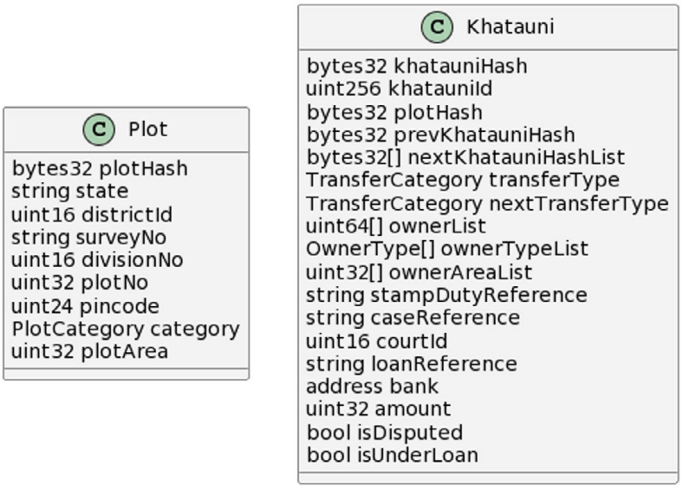


Fig. 2 Class diagram

```
yugam@Yugan: ~/Documents/mtp/chain_test
}
> var bac = "0x264A0C337adf7118cA39B066005A3aa272cD9cC3";
undefined
> contractInstance.KhatauniLoanGenesisEntry.sendTransaction(301, pl101, [789], [1], [700], bac, 'L12', 10000, {from: lr
a_1, gasPrice: 1});
ReferenceError: pl101 is not defined
        at <eval>:1:64(4)

> var pl101 = '0xe2fc4c5336b69ea49dd36d372699e4e9054bd5fd442416f61613b7791c81eeea'
undefined
> contractInstance.KhatauniLoanGenesisEntry.sendTransaction(301, pl101, [789], [1], [700], bac, 'L12', 10000, {from: lr
a_1, gasPrice: 1});
"0x80bb3be4360b93472e374537ebaaa54be08f6873c8f7b4b1b7067fdd268aaed5"
> eth.getTransactionReceipt('0x80bb3be4360b93472e374537ebaaa54be08f6873c8f7b4b1b7067fdd268aaed5')
{
  blockHash: "0x241989d4f087f47d2fd39406295011355d3942f7a62a45966a122dc4538dbb83",
  blockNumber: 4430,
  contractAddress: null,
  cumulativeGasUsed: 333718,
  effectiveGasPrice: 1,
  from: "0x60c33b475c958a4ee5681956f286ddbf9f296671",
  gasUsed: 333718,
  logs: [{
      address: "0xcb0466814a7c8a68bb9471faaa3c43a5541e25b5",
      blockHash: "0x241989d4f087f47d2fd39406295011355d3942f7a62a45966a122dc4538dbb83",
      blockNumber: 4430,
      data: "0x",
      logIndex: 0,
      removed: false,
      topics: ["0x997f3c70911d736ad311edda86c017044b934985c948cd1d200c0452d1f75957", "0xc1010d7cb7d70251e3924808a529333
fd2f36512da4f047240c2ebab0ebe62bf"],
      transactionHash: "0x80bb3be4360b93472e374537ebaaa54be08f6873c8f7b4b1b7067fdd268aaed5",
      transactionIndex: 0
  }],
  logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00004000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
10000080000000000000000000000000000000000000000000400000000000000000000000000001000000000000000000000000000000000000000000000000000
00000000000000000400000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000110000000000000000000400000000000000000000000000",
  status: "0x1",
  to: "0xcb0466814a7c8a68bb9471faaa3c43a5541e25b5",
  transactionHash: "0x80bb3be4360b93472e374537ebaaa54be08f6873c8f7b4b1b7067fdd268aaed5",
  transactionIndex: 0,
  type: "0x0"
}
>
```

Fig. 3 Device1 LRO genesis loan entry

3.1 Accounts

- **LRO**: Land Registration Office (LRO) is responsible for all the transactions/ changes in the Khatauni. In India, the Tehsil office is responsible for the land records of its administrative area.
- **Bank**: The bank marks loan completion after loan is paid on the property under loan.
- **Department:** The department is responsible for contract deployment, and LRO and bank accounts creation. Department of Land Resources, Government of India is responsible for land information management system.

3.2 Entity

- **Plot**: A piece of land that is uniquely identified by land survey. It maps physical land to the land record.

```
MINGW64:/c/Users/yugam/Documents/MTP/device_2
undefined
> var bac = "0x264A0C337adf7118cA39B066005A3aa272cD9cC3";

undefined
> var kh101 = '0xc1010d7cb7d70251e3924808a529333fd2f36512da4f047240c2ebab0ebe62bf'

undefined
> contractInstance.loanCompletion.sendTransaction(kh101, { from: bac, gasPrice: 1 });

"0xd1a4204c4321c73f64370e64b3e590243887c83c1eaaa0fe2b0ca63f28d32739"
> eth.getTransactionReceipt('0xd1a4204c4321c73f64370e64b3e590243887c83c1eaaa0fe2b0ca63f

{
  blockHash: "0x8ebdca511472cdf8431368831bd31f5730cb44a60b209f6ea7c47b21aeaa06f5",
  blockNumber: 4509,
  contractAddress: null,
  cumulativeGasUsed: 23983,
  effectiveGasPrice: 1,
  from: "0x264a0c337adf7118ca39b066005a3aa272cd9cc3",
  gasUsed: 23983,
  logs: [{
      address: "0xcb0466814a7c8a68bb9471faaa3c43a5541e25b5",
      blockHash: "0x8ebdca511472cdf8431368831bd31f5730cb44a60b209f6ea7c47b21aeaa06f5",
      blockNumber: 4509,
      data: "0x",
      logIndex: 0,
      removed: false,
      topics: ["0x0dc9f80dcec238dd4f87c46f90161e4a33d3eb66c00dd2d6c3a92a638f006456", "0
      transactionHash: "0xd1a4204c4321c73f64370e64b3e590243887c83c1eaaa0fe2b0ca63f28d32
      transactionIndex: 0
  }],
  logsBloom: "0x0000000000000000000000000000000000000000100000000000000000000000000000
0000000000000000000000000000000000000000
```

Table 1 Function table

Function	Caller account	Functionality
viewKhatauni	All	View Khatauni details
createRegistryOffice	Department	Create LRO account
createBank	Department	Create bank account
createPlot	LRO	Create plot with unique plot hash
khatauniGenesisEntry	LRO	First Khatauni record entry
khatauniLoanGenesisEntry	LRO	First loan Khatauni record entry
sourceKhatauniIssue	Internal only	Returns true if Khatauni is not sellable
khatauniSale	LRO	Sale/loan sale, Transfer of Khatauni
courtDispute	LRO	Mark or remove dispute on a Khatauni
loanCompletion	Bank	Mark loan completion on a Khatauni
courtOrder	LRO	Execute court order on a Khatauni

Plot and Khatauni are mapped by plotHash and khatauniHash, respectively. The latest Khatauni is tracked from the starting (genesis entry) of that Khatauni record in the blockchain. Any changes to existing Khatauni add up to that same Khatauni chain. These functions are implemented in our solution and are presented in Table 1.

4 Results and Conclusion

The smart contract is written in Solidity [16]. The contract was first tested with Truffle [17], a suite of tools for the smart contract development. It was then deployed on an Ethereum private POA chain. The chain has a block time of 5 s and block gas limit of 8,000,000. Department and LRO nodes are running on Device1. Bank nodes are running on Device2. The transaction analysis on deployed smart contract is shown in Table 2. The solidity code of our proposed smart contract is made available in GitHub [18].

The benchmark data to compare our implementation with existing system is estimated using Uttar Pradesh property registration data [19] from January 1, 2023 to October 22, 2023. Indian property registration data is estimated using Uttar Pradesh's annual estimate from the available date range data (35 lakhs) and then extrapolating using population proportions to the country level. Even with an estimation of 2.5

Table 2 Transaction analysis

Parameter	Value
Average gas used per transaction	194,392
Gas limit per block	8,000,000
Average no. of transactions per block	41
Block period (s)	5
No. of transactions per second	8
Transactions per year	252,288,000

Table 3 Related works—comparison table

Proposal	Land transfer type	Banks	Litigation
Vinay et al. [11]	Simple sale	Never	Yes
Pongnumkul et al. [13]	Loan Sale only	Always	No
Khalid et al. [14]	Full sale only	Never	Yes
Kazi et al. [12]	Simple sale	Never	No
Our proposal	Full, partial, loan sale, etc.	if loan sale	Yes

crore registrations in the country, our proposal is stable as it can handle ten times the estimated count. Further scaling can be achieved by increasing the block gas limit.

In Table 3, we compare our work with the other related proposals. It can be seen that our proposal is a step toward a comprehensive scalable solution involving the various stakeholders. The immutability aspects are well captured by the blockchain model and proof of authority ensures that the State does not lose its required control to ensure the authenticity of the land holdings. A major challenge to be addressed is the effective migration of legacy data with ongoing live transactions to the new implementations proposed.

References

1. Azadi H, Vanhaute E (2019) Mutual effects of land distribution and economic development: evidence from Asia, Africa, and Latin America. Land 8(6):96
2. Debroy B, Jain S (2016) Strengthening arbitration and its enforcement in India–resolve in India. Research Paper of the Niti Ayog
3. Real property act 1858 transcript, South Australia parliament. https://www.foundingdocs.gov.au/resources/transcripts/sa8_doc_1858.pdf. Accessed 2 Nov 2023
4. McCormack JL (1992) Torrens and recording: land title assurance in the computer age. William Mitchell Law Review 18(1)
5. Agarwal BK (2018) Conclusive land title system for India. Dissertation for Master of Philosophy in Social Sciences

6. Deininger K, Goyal A (2012) Going digital: credit effects of land registry computerization in India. J Develop Econ 99(2):236–243. https://doi.org/10.1016/j.jdeveco.2012.02.007
7. Vos J, Lemmen C, Beentjes B (2017) Blockchain based land administration feasible, illusory or a panacea. In: Netherlands Cadastre, land registry and mapping agency. Paper prepared for presentation at the 2017 world bank conference on land and Povertry. The World Bank, Washington, DC
8. National land records modernization programme (nlrmp). https://dolr.gov.in/en/document/success-stories-national-land-records-modernization-programme-nlrmp. Accessed 2 Nov 2023
9. Digital india land records modernization programme (dilrmp). https://dolr.gov.in/en/programme-schemes/dilrmp/digital-india-land-record-modernization-programme. Accessed 2 Nov 2023
10. National strategy on blockchain, meity government of india. https://www.meity.gov.in/writereaddata/files/National_BCT_Strategy.pdf (2021). Accessed 3 Nov 2023
11. Thakur V, Doja M, Dwivedi YK, Ahmad T, Khadanga G (2020) Land records on blockchain for implementation of land titling in India. Int J Inf Manage 52:101, 940. https://doi.org/10.1016/j.ijinfomgt.2019.04.013
12. Alam KM, Ashfiqur Rahman J, Tasnim A, Akther A (2022) A blockchain-based land title management system for Bangladesh. J King Saud Univ Comput Inf Sci 34(6, Part A):3096–3110. https://doi.org/10.1016/j.jksuci.2020.10.011
13. Pongnumkul S, Khonnasee C, Lertpattanasak S, Polprasert C (2020) Proof-of-concept (poc) of land mortgaging process in blockchain-based land registration system of thailand. In: Proceedings of the 2020 the 2nd international conference on blockchain technology, ICBCT'20, pp 100–104. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3390566.3391669. https://doi.org/10.1145/3390566.3391669
14. Khalid M, Iqbal J, Alturki A, Hussain S, Alabrah A, Ullah SS (2022) Blockchain-based land registration system: a conceptual framework. Appl Bionics Biomech 2022. https://doi.org/10.1155/2022/3859629
15. Keccak. https://keccak.team/keccak.html. Accessed 22 Oct 2023
16. Solidity. https://soliditylang.org/about/. Accessed 22 Oct 2023
17. Truffle. https://trufflesuite.com/. Accessed 22 Oct 2023
18. Indian land registration contract, github. https://github.com/yug-am/lrscontract
19. Uttar pradesh stamp and registration department , property registration dashboard. https://igrsup.gov.in/igrsup. Accessed 22 Oct 2023

From Sparsity to Accuracy: Enhancing Recommendation Systems with Item-Based Collaborative Filtering and SVD Matrix Factorization

Arkajit Banerjee and Bam Bahadur Sinha

Abstract An investigation was undertaken to assess the effectiveness of item-based collaborative filtering (IBCF) within recommendation systems, particularly in scenarios where user preferences are significantly influenced by item attributes. The Pearson Correlation served as the chosen similarity metric in the User-Item Interaction Matrix. The efficiency of the recommender engine was adversely affected by a notable matrix sparsity of 93.7%, a computation based on the MovieLens 100K dataset. To mitigate this challenge, the study implemented the Singular Value Decomposition Matrix Factorization technique. After sparsity reduction and missing value imputation within the MovieLens 100K dataset, recommendations were regenerated. Notably, the proposed model demonstrated superior performance compared to the prior sparsity-based model. The efficiency increased from 75% pre-sparsity reduction to a noteworthy 79% following the implementation of the SVD algorithm.

Keywords Collaborative filtering · MovieLens 100K · Singular value decomposition · Pearson correlation coefficient · Information filtering

1 Introduction

In the expansive realm of digital content and online experiences, the copiousness of information poses both a boon and a challenge. The digital age has witnessed an unparalleled surge in data, encompassing a spectrum from movies, music, and books to products, articles, and social interactions. Efficiently navigating this deluge of information parallels the challenge of locating a needle in a haystack, often leaving users inundated by the sheer abundance of options. As users traverse an array of

A. Banerjee
Department of Statistics, Acharya Prafulla Chandra College, New Barrackpore, Kolkata, India
e-mail: arkajit@apccollege.ac.in

B. B. Sinha (✉)
Computer Science and Engineering, National Institute of Technology Sikkim, Ravangla, India
e-mail: bahadurbam43@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_7

choices, the imperative for effective guidance and personalized assistance becomes increasingly evident. Recommendation systems emerge as a solution to address this information overload, proffering tailored suggestions aligned with individual preferences, behaviors, and interests. They play a pivotal role in reshaping the dynamics of content discovery, consumption, and engagement across diverse platforms.

Recommendation systems function as intelligent intermediaries between users and the expansive array of choices prevalent in the digital domain. Whether it is streaming services proposing the subsequent binge-worthy series, e-commerce platforms suggesting products or social networks presenting pertinent connections and content, these systems employ advanced algorithms to craft personalized experiences [1]. As users pursue more customized and pertinent interactions, businesses and platforms endeavor to furnish content and services that resonate at an individual level. The incorporation of recommendation systems not only amplifies user satisfaction but also carries substantial implications for engagement, retention, and the overarching success of digital platforms within an increasingly competitive milieu [2].

Through the utilization of data analytics, machine learning, and artificial intelligence, recommendation systems contribute to a user journey characterized by enhanced seamlessness, enjoyment, and efficiency [3]. The inaugural recommender system, denoted Grundy, was conceptualized by Elaine Rich in 1979. Her initiative involved formulating a system that posed targeted questions to users, subsequently determining applicable stereotypes based on their responses and offering book recommendations accordingly [4]. Another early recommender system, referred to as a "digital bookshelf," was elucidated by Jussi Karlgren of Columbia University in a 1990 technical report. Karlgren proceeded to implement this system on a substantial scale and continued refining it through subsequent technical reports and publications, commencing in 1994 [5, 6].

Recommendation systems are categorically classified into three types, as depicted in Fig. 1 The collaborative filtering technique exploits the collective intelligence of a user community to furnish personalized recommendations. Introduced by Resnick et al. in 1994 [7], the fundamental concept behind collaborative filtering involves identifying similarities between users or items based on their historical interactions within a system. Through the analysis of user behaviors such as ratings, purchases, or clicks, collaborative filtering algorithms discern patterns and relationships, facilitating the prediction of user preferences for items they have not encountered. Two categories of Collaborative Filtering Algorithms exist: memory-based and model-based.

The content-based filtering technique focuses on the inherent characteristics of items and user preferences. In contrast to collaborative filtering, which relies on the collective behavior of a user community, content-based filtering generates recommendations by scrutinizing the features or content attributes of items and aligning them with users' preferences. This approach proves particularly effective in scenarios where explicit user-item interactions are sparse, posing challenges in identifying user preferences solely based on historical behavior.

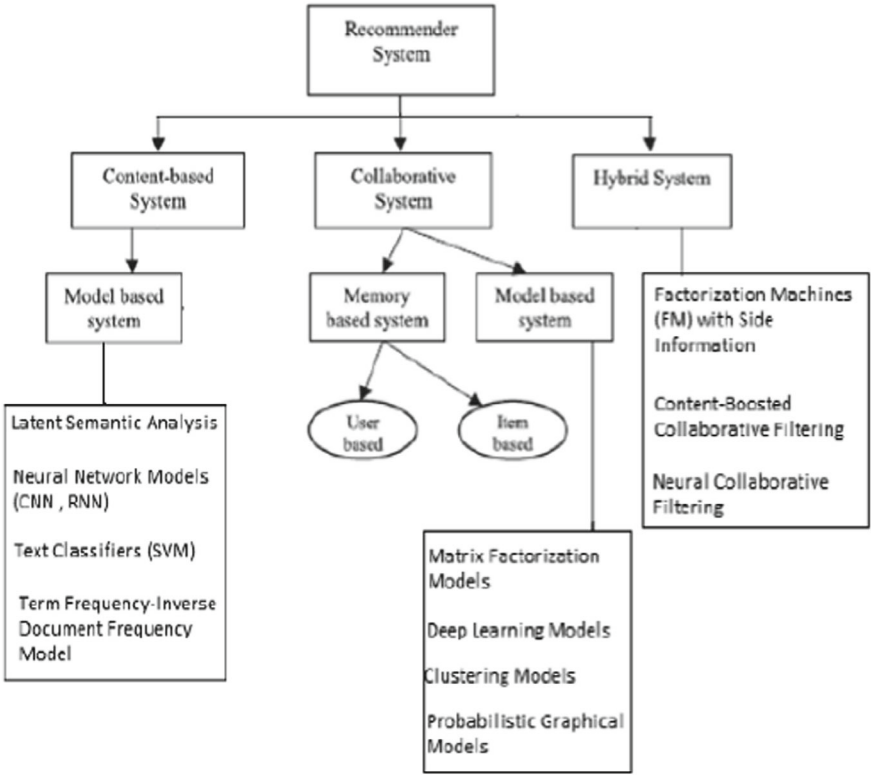


Fig. 1 Key types of recommendation systems

The Hybrid Filtering Technique integrates collaborative filtering, content-based filtering, and other approaches [8, 9]. Hybrid approaches can be implemented through various methods: by generating content-based and collaborative based predictions independently and subsequently merging them; by incorporating content-based capabilities into a collaborative-based approach (and vice versa); or by amalgamating the approaches into a unified model.

This paper centers on the Item-Based Collaborative Filtering (IBCF) approach. When engaging with IBCF, challenges such as sparsity are encountered, necessitating the utilization of Singular Value Decomposition (SVD). It is observed that the introduced model outperforms conventional Collaborative Filtering (CF) after the implementation of SVD.

Major Contributions:

- Imputation of sparsity values in the User-Item Interaction Matrix.
- Elaboration on the application of Singular Value Decomposition (SVD), a model-based technology, for dimensionality reduction in recommender systems to generate predictions.

- Utilization of low-dimensional representation for computing neighborhoods to generate recommendations.
- Presentation of experimental results with SVD on the test dataset MovieLens.

The paper's structure is delineated as follows: Sect. 2 delves into the pertinent background and literature survey. In Sect. 3, the experimental methodology is expounded, encompassing the elucidation of theories, methods, approaches employed, and challenges encountered. Section 4 outlines the proposed framework, detailing how the challenge of low efficiency attributable to sparsity was addressed through Singular Value Decomposition. Section 5 elucidates the results and graphs derived from the experimental procedures. Section 6 provides insights into the Conclusion and Future Plans.

2 Literature Survey

Sparsity in recommender systems refers to the prevalence of missing or incomplete data within the user-item interaction matrix. This phenomenon arises when users interact with only a small fraction of the available items, leading to sparse matrices. The significance lies in the challenges it poses to the accuracy and efficiency of collaborative filtering models. Sparsity significantly impacts collaborative filtering models, which rely on user-item interactions to infer preferences.

The sparse nature of the data can lead to biased recommendations and reduced model performance, necessitating effective sparsity reduction techniques.

- To address the primary limitations inherent in recommender systems, namely scalability and sparsity, dimensionality reduction and ontology techniques have been employed by the authors in [10]. Subsequently, ontology is applied to enhance the recommendation accuracy of the CF component. Additionally, Singular Value Decomposition (SVD), serving as another dimensionality reduction method, is implemented in the CF section to identify the most comparable items and users within each cluster, thereby significantly Item CF and SVD-based recommender system 5 enhancing the scalability of the recommendation method. To demonstrate the efficacy of the method, it has been tested on two real-world datasets, and the outcomes have been compared with those of other methods reported in the literature. The results unequivocally showcase the effectiveness of the approach in addressing the sparsity and scalability issues inherent in Collaborative Filtering.
- The primary aim of the work done in [11] is to elucidate the movie recommendation system using two CF algorithms in Apache Mahout. Furthermore, the utilization of Python's Matplotlib libraries is employed for data analysis, facilitating a comprehensive understanding of the movie dataset. The machine learning algorithm, Log Likelihood Similarity, is applied for calculating item similarity. The secondary objective involves a comparative assessment of the effectiveness and efficiency of item-based and user-based recommender systems. The authors achieved a zero average absolute difference, thereby demonstrating

a 100% accuracy in predictions based on the ratings of the suggested items. The major drawback of this proposed framework is its failure to generate a diverse set of recommendations which ultimately reduced user satisfaction.

- In [12], an algorithm for personalized recommendations is proposed by the authors, centered around approximating singular value decomposition (SVD). Singular value decomposition is identified as an effective method for dimensionality reduction. Unfortunately, it is not fit for practical use with big datasets because of its high computing costs and poor performance with big sparse matrices. As a solution, the authors provide the ApproSVD method. Using the MovieLens dataset, an empirical investigation is carried out to evaluate the suggested method's prediction quality in comparison to Drineas's LINEARTIMESVD algorithm and the conventional SVD technique. The findings reveal that the ApproSVD method outperforms the others. Furthermore, for the suggested ApproSVD, an optimal parameter pair is determined, with '*c*' (number of rows sampled from the original matrix) set to 600, and '*k*' (number of top singular values to be approximated from the original matrix) set to 600.
- In [13], a novel collaborative filtering recommendation algorithm is introduced, leveraging evolutionary clustering and user correlation. To generate denser score data, the score matrix undergoes initial preprocessing involving dimension reduction and normalization. Subsequently, dynamic evolutionary clustering is implemented, and the clustering principle is derived from the processed data. Secondly, the identification of the nearest neighbors, sharing the highest level of interest, is undertaken. A metric for user correlation, measuring the relationship between users, is proposed, incorporating user satisfaction and potential information. User correlation is then applied within each user group to select the closest neighbors for rating prediction. The Movielens dataset is employed to evaluate the proposed approach. The results from the diversity experiment indicate exceptional performance of the suggested method in terms of recommended precision and predicted accuracy.

3 Experimental Methodology

3.1 Collaborative Filtering

Algorithms propose things (the filtering portion) based on preferences from a large number of users (the collaborative part). This method learns to anticipate future interactions by comparing the similarity of user preference behavior to prior interactions between users and products. These recommender systems construct a model based on a user's prior behavior, such as things bought previously or ratings assigned to such items, as well as comparable judgements made by other users. The theory is that if several individuals have made similar judgements and purchases in the past, such as selecting a movie, there is a good chance they will agree on additional future choices. For example, if a collaborative filtering recommender recognizes me and

another user with similar movie interests, it may recommend a movie to me that it already knows this other person like. There are two kinds of CF approaches described in the literature: memory-based and model-based.

Model-Based Collaborative Filtering is a recommendation system technique that builds predictive models to understand and capture user-item preferences. These models are then used to make predictions about user preferences for items they have not yet interacted with. However, in this paper, memory-based collaborative filtering has been employed which is a recommendation system technique that relies on the direct comparison of user-item interactions to generate recommendations. This approach leverages the collective preferences of users by identifying similarities between users or items based on historical interaction data. Memory-based methods can be broadly categorized into two types: User-Based CF and Item-Based CF.

User-Based CF compares a target user's preferences with those of other users who have exhibited similar tastes in the past. The system identifies users with comparable behavior and recommends items liked by these like-minded individuals. This approach is effective in capturing user preferences based on social or taste similarities. However, it can face challenges in scenarios where user-item interactions are sparse or when dealing with the cold start problem for new users.

Item-Based Collaborative Filtering, on the other hand, focuses on the inherent characteristics of items. It identifies items that are similar to those a user has shown interest in and recommends items based on this similarity. This approach is particularly robust in scenarios where items have stable characteristics and can be easily compared. Item-based collaborative filtering often performs well in overcoming the cold start problem for new items.

The choice between IBCF and UBCF depends on various factors, and both approaches have their strengths and weaknesses. Here are some scenarios in which Item-Based Collaborative Filtering might be preferred over User-Based Collaborative Filtering:

- Stable Item Characteristics: Item-Based Collaborative Filtering is often more suitable when the characteristics of items remain relatively stable over time as they can be effective in capturing these stable attributes.
- Scalability: Item-Based Collaborative Filtering tends to be more scalable than User-Based Collaborative Filtering, especially as the number of users in a system grows. Calculating item-item similarity matrices is generally computationally less demanding than user-user similarity matrices.
- Sparse User-Item Interaction Matrix: In scenarios where the user-item interaction matrix is highly sparse, meaning users have interacted with only a small subset of items, IBCF can be more robust. It leverages the characteristics of items, and since items are usually less sparse than users, it helps in providing more reliable recommendations.
- Handling Cold Start for New Items: IBCF can handle the cold start problem for new items more effectively than UBCF. If a new item is introduced, its characteristics can be compared to existing items, and recommendations can be made based on item similarity.

- Item Quality Matters: When the quality of items is a significant factor in recommendations, IBCF can be advantageous. It recommends items similar to those the user has enjoyed, potentially leading to a higher likelihood of user satisfaction with the suggested items.

3.2 Pearson Correlation

The Pearson correlation coefficient is calculated using Eq. 1.

$$p(X1, Z1) = \frac{\sum_{i=1}^{n}(X1_i - \widehat{X1})(Z1_i - \widehat{Z1})}{\sqrt{\sum_{i=1}^{n}(X1_i - \widehat{X1}^2(Z1_i - \widehat{Z1})^2}} \tag{1}$$

A statistical measure of the linear connection between two variables, Pearson correlation, frequently written as ρ, is computed using Eq. 1. One typical use of Pearson correlation in recommendation systems and collaborative filtering is to measure the degree to which users or objects are similar using their past interactions as a foundation for personalized suggestions. You may set it anywhere from –1 to + 1. When one variable goes up, the other usually goes up too, according to a positive connection. When one variable goes up, the other usually goes down; this is called a negative correlation. Use in CF: Pearson correlation is a common tool in collaborative filtering for gauging how similar people or things are according to their ratings or interactions. Collaborative filtering based on things rather than individuals uses the correlation between items to find related items, and vice versa for user-based collaborative filtering.

3.3 Singular Value Decomposition (SVD)

SVD serves as a mathematical method to address sparsity and reduce dimensionality in recommender systems. This technique proves particularly effective when dealing with sparse matrices inherent in collaborative filtering applications, where user-item interaction data is often incomplete.

Mathematically, consider a user-item interaction matrix 'M' representing user ratings for items. The matrix 'M' is decomposed into three matrices, 'U', 'S', and V^T, through SVD (Eq. 2).

$$M = U.S.V^T \tag{2}$$

Here, 'U' denotes the user-feature relationships, 'S' is a diagonal matrix containing singular values, and V^T represents item-feature relationships. In the recommender

system context, the singular values in 'S' capture the strength of user-item relationships. The matrices 'U' and V^T offer reduced dimensional representations of users and items, respectively. The decomposition enables the approximation of the original matrix 'M' by selecting the top 'k' singular values and their corresponding vectors from 'U' and V^T (Eq. 3).

$$\widehat{M_k} = U_k S_k V_k^T \tag{3}$$

Here, 'U_k', S_k, and ${V_k}^T$ consist of the top 'k' columns of 'U', the top $k \times k$ submatrix of S, and the top 'k' rows of V^T, respectively. By appropriately choosing 'k', the approximation M_k retains the essential features of the original matrix while mitigating sparsity. This allows for more efficient and accurate recommendations in situations characterized by sparse user-item interaction data. The selection of 'k' involves finding a balance between computational complexity and the desire for a compact representation that captures crucial data patterns.

4 Proposed Framework

Understanding the theory of Singular Value Decomposition (SVD) in the context of Item-Based Collaborative Filtering, utilizing the MovieLens 100 k dataset is paramount. A matrix $A_{m \times p}$ represents items with 'p' columns and users with 'm' rows. The primary objective of employing SVD is to decompose the matrix $A_{m \times p}$, with a large dimension, into three matrices with potentially smaller dimensions.

Matrixes 'U' and 'V' are orthogonal, and the SVD assertion is mathematically expressed as $A = USV^T$. Each column of 'U' contains the eigenvectors of the AA^T matrix, which have an order of $m \times r$. The eigenvectors of the matrix A^TA, order $p \times r$, make up the columns of the vector space 'V'. The singular values, ordered by decreasing magnitude, are the diagonal elements of the matrix 'S', which have an order of $r \times r$.

The matrices obtained via Singular Value Decomposition (SVD) are very valuable for practical purposes since they provide the most accurate lower-rank approximations based on the Frobenius Norm. To generate a matrix S_k, where $k < r$, the $r \times r$ matrix 'S' is modified to retain just the 'k' biggest diagonal values. By suitably lowering matrices 'U' and 'V', the reconstructed matrix $A_k = U_k S_k {V_k}^T$ becomes the closest matrix of rank $-k$ to A. Put simply, A_k minimizes the Frobenius norm $\|A_k - A\|$ across all matrices of rank$-k$.

The process of predicting ratings using SVD entails constructing a matrix based on the dataset, where users are represented as rows, films as columns, and ratings as individual values. The system fills empty spaces by using latent associations between users and movies. The underlying premise is that the rank-k SVD is capable of capturing the collection of 'k' elements that have an impact on a user's rating for a movie.

To generate recommendations, a high correlation with the target movie is identified. The list is arranged in decreasing order of correlation, and the number of ratings for each movie is considered. Recommendations are generated by selecting the top 10 strongly correlated movies, with an additional condition of a significantly high number of ratings, preferably above 30. This ensures recommendations are based on popular movies with a considerable number of user ratings. Figure 2 depicts the proposed flow of the recommendation system.

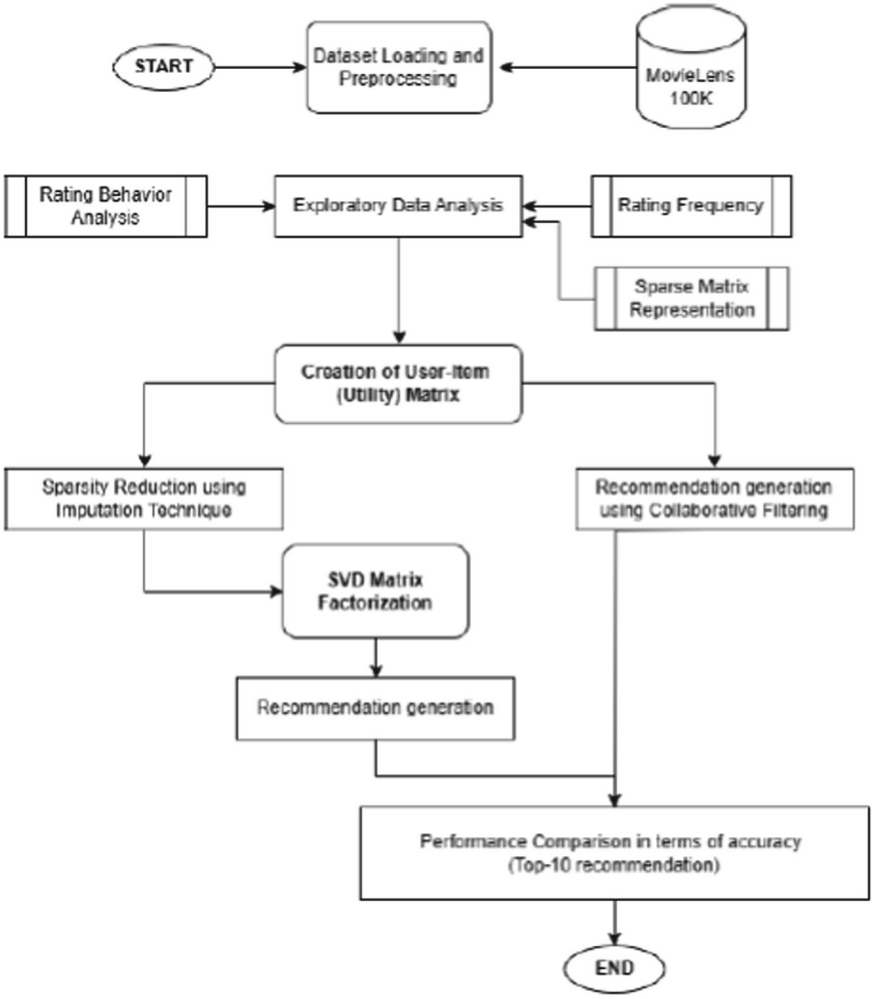


Fig. 2 Proposed flow of recommendation system

5 Results and Discussion

This section highlights the results obtained at each stage of the proposed framework. Figure 3 illustrates a joint plot depicting the relationship between average ratings and the number of ratings for movies. The histogram of average ratings reveals a central density concentrated between values 3.5 and 4, indicating that users, on average, have rated movies within this range. Conversely, the histogram for the number of ratings displays skewness, suggesting that fewer users have rated a larger number of movies.

Figure 4 presents a bar plot where the highest frequency is observed for a rating value of 4, indicating that most individuals have assigned this rating. Conversely, the lowest frequency is recorded for rating 1. The graph is organized in decreasing order based on the magnitude of frequency. The user-item interaction matrix is depicted, featuring rows corresponding to user IDs and column names representing movie names. The presence of "NaN" in numerous cells is observed, indicating empty values. A logical inference can be drawn: not all users rate all movies due to the inherent limitation that individuals only rate movies they have seen. Consequently, a highly sparse matrix is evident, as it is impractical for every person to rate every movie. This signifies the importance of applying Singular Value Decomposition (SVD) to the matrix.

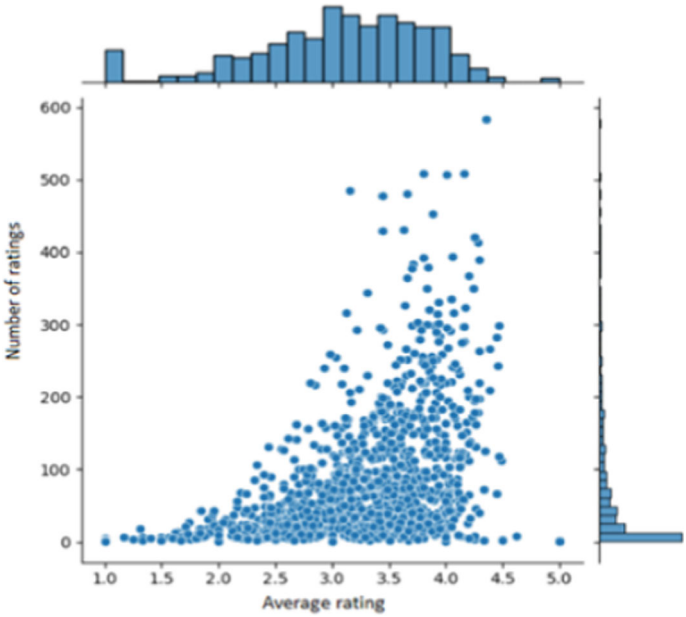


Fig. 3 Rating behaviour—average rating versus number of ratings

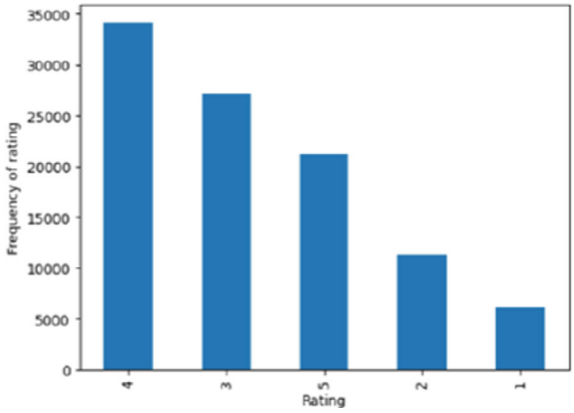


Fig. 4 Rating frequency—MovieLens 100K

Figure 5 provides a visual representation of the sparse user-item interaction matrix, where rows represent users, and columns represent movie names. The matrix comprises 943 rows and 1664 columns. Notably, a substantial portion of the matrix is observed to be white, indicative of sparsity or emptiness, as reported at a rate of 93.7%.

Table 1 is employed to assess the movie-wise efficacy of the recommendation system both pre and post-sparsity reduction. The evaluation involves the random selection of 10 movies for which recommendations are generated. For each of these 10 movies, 10 recommendations are generated, taking into consideration parameters such as high correlation and a high number of ratings for relevance. The determination of True or False is based on whether the generated recommendations satisfy both conditions. For instance, in the case of the movie "Scream," before sparsity reduction, the recommendation engine produced 8 True recommendations, indicating

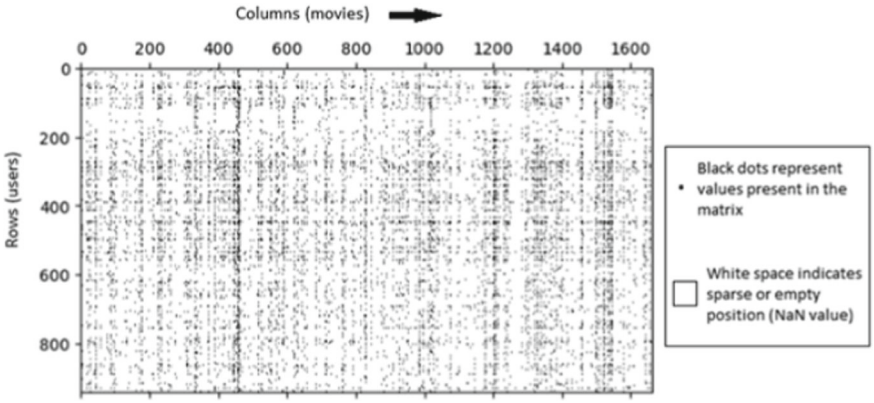


Fig. 5 Sparse matrix representation—MovieLens 100K

Table 1 Proposed f model performance (Item CF with SVD)

Movie name	Efficiency before SVD	Efficiency after SVD
Scream (1996)	80	90
Star Wars (1997)	90	100
Kicked in the Head (1997)	90	60
Enchanted April (1991)	40	100
Wonderful, Horrible Life (1993)	90	100
Kissed (1996)	80	50
Twin Town (1997)	80	50
Copycat (1995)	50	60
Outbreak (1995)	60	100
Daylight (1996)	90	80

80% accuracy. Post-sparsity reduction, the efficiency of the engine increased to 90%, resulting in the generation of 9 True Recommendations.

In Fig. 6, two bars are presented, with the x-axis representing the models utilized for recommendation generation. Before sparsity reduction via Singular Value Decomposition (SVD), conventional Item-Based Collaborative Filtering was employed, and the recorded efficiency of the model was 75%. Following sparsity reduction, the model's performance efficiency increased to 79%. The reported efficiency is an average value, computed by averaging all the efficiency percentages before and after the implementation of SVD, providing a comprehensive overview of the model's overall performance.

Figure 7 facilitates the visualization of the change in movie-wise efficiency after sparsity reduction from the user-item interaction matrix using Singular Value

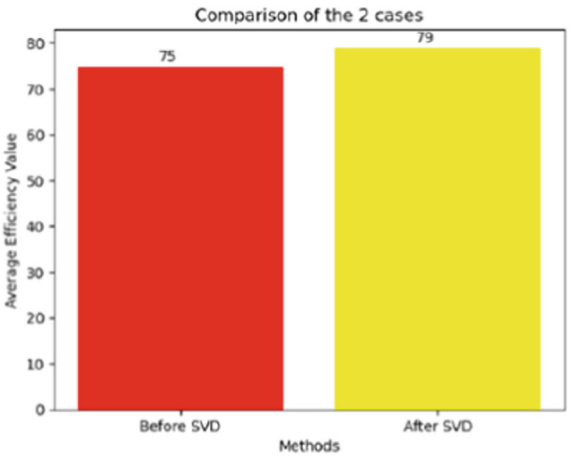


Fig. 6 Efficiency magnitude before and after SVD implementation

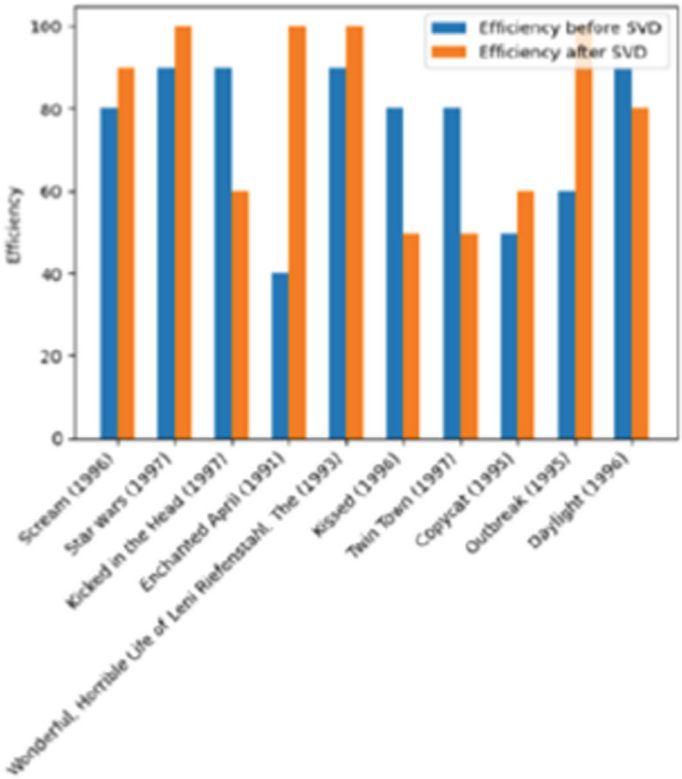


Fig. 7 Comparative analysis using 2D clustered bar plot

Decomposition (SVD). While a few movies experienced a slight decline, a significant number exhibited positive growth. Notably, some movies achieved 100% efficiency after the implementation of SVD, contributing to an enhancement in the overall average performance from 75 to 79%. This signifies an improvement in the recommendation system's effectiveness.

The depicted Fig. 8 features a two-sided bar plot designed to elucidate the movie-wise variations in efficiency following the implementation of Singular Value Decomposition (SVD). The *x*-axis represents movie names, while the y-axis illustrates the percentage increase or decrease in efficiency after applying the proposed framework. In most instances, positive spikes are evident, signifying an enhancement in the model's performance. While a few movies experienced marginal declines, the overall testing on 10 random movies demonstrated an increase in efficiency from 75 to 79%.

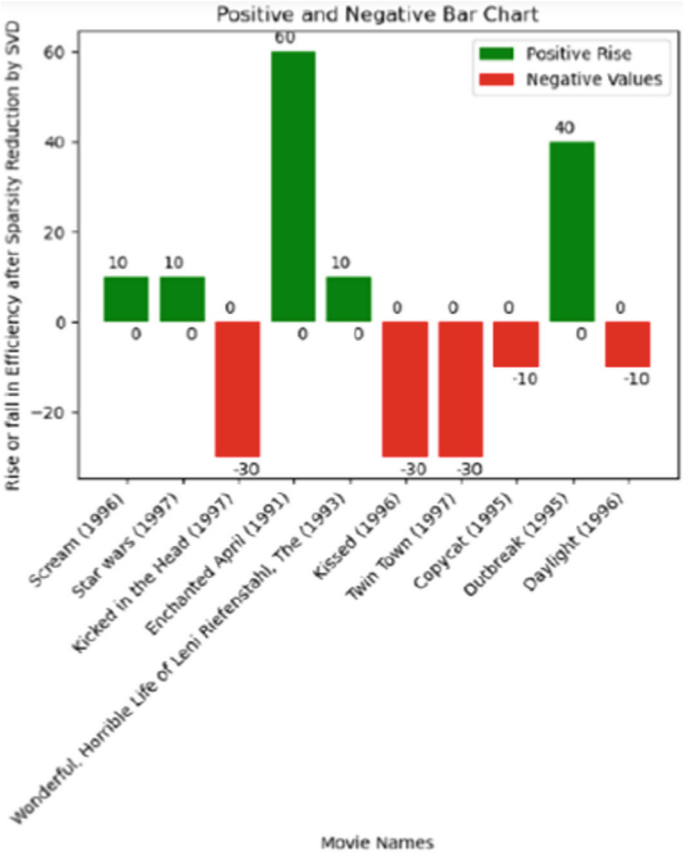


Fig. 8 Movie-wise rise and fall of efficiency post SVD implementation

6 Conclusion and Future Work

Item-Based Collaborative Filtering (IBCF) serves as a prevalent and potent technique within recommendation systems, particularly in contexts where item characteristics significantly influence user preferences. The Pearson Correlation was employed as the similarity measure for the User-Item Interaction Matrix. Subsequent sparsity computation revealed a substantial matrix sparsity of 93.7%, adversely impacting the efficiency of the recommender engine. In response, the Singular Value Decomposition Matrix Factorization method was employed to mitigate this sparsity issue. Following sparsity reduction and missing value imputation, recommendations were regenerated, demonstrating the enhanced efficiency of the proposed model compared to its predecessor afflicted by sparsity.

The present focus is on the continual improvement and advancement of the recommendation system utilizing Item-Based Collaborative Filtering (IBCF), achieving an efficiency of 79%. Future directives aim to further enhance the system through:

- Improving Similarity Metrics: This involves incorporating contextual information, semantic analysis, or employing deep learning techniques to refine item similarity calculations, thereby elevating recommendation accuracy.
- Reducing Forceful Decrease of Sparsity: The objective is to boost efficiency without compromising on data integrity, emphasizing a balanced approach to sparsity reduction.
- Adjusting Important Features of Items: This entails adapting the significant features or aspects of items based on evolving user preferences and contextual factors, achieved through the introduction of weighted adjustments.

References

1. Sinha BB, Dhanalakshmi R (2019) Evolution of recommender system over the time. Soft Comput 23(23):12169–12188
2. Sinha BB, Dhanalakshmi R, Regmi R (2020) TimeFly algorithm: a novel behavior-inspired movie recommendation paradigm. Pattern Anal Appl 23:1727–1734
3. Rich E (1979) User modeling via stereotypes. Cogn Sci 3(4):329–354
4. Leite WL, Roy S, Chakraborty N, Michailidis G, Huggins-Manley AC, D'Mello S, Jing Z (2022) A novel video recommendation system for algebra: an effectiveness evaluation study. In: LAK22: 12th International learning analytics and knowledge conference, pp 294–303
5. Wang G, Zhang X, Tang S, Zheng H, Zhao BY (2016) Unsupervised clickstream clustering for user behavior analysis. In: Proceedings of the 2016 CHI conference on human factors in computing systems, pp 225–236
6. Aslan I, Murer M, Primessnig F, Moser C, Tscheligi M (2013) The digital bookshelf: decorating with collections of digital books. In: Proceedings of the 2013 ACM conference on pervasive and ubiquitous computing adjunct publication, pp. 777–784
7. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on computer supported cooperative work, pp 175–186
8. Nilashi M, Ibrahim O, Bagherifard K (2018) A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Expert Syst Appl 92:507–520
9. Wu CSM, Garg D, Bhandary U (2018) Movie recommendation system using collaborative filtering. In: 2018 IEEE 9th international conference on software engineering and service science (ICSESS). IEEE, pp 11–15
10. Sinha BB, Dhanalakshmi R (2021) Building a fuzzy logic-based McCulloch-Pitts neuron recommendation model to uplift accuracy. J Supercomput 77:2251–2267
11. Sinha BB, Dhanalakshmi R (2022) DNN-MF: deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems. Neural Comput Appl 34(13):10807–10821
12. Zhou X, He J, Huang G, Zhang Y (2012) A personalized recommendation algorithm based on approximating the singular value decomposition (ApproSVD). In: 2012 IEEE/WIC/ACM International conferences on web intelligence and intelligent agent technology, vol 2. IEEE, pp 458–464
13. Chen J, Zhao C, UlijiChen L (2020) Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering. Complex Intell Syst 6:147–156

A Comparative Analysis of Sorting Algorithms for Large-Scale Data: Performance Metrics and Language Efficiency

Cosmina-Mihaela Roșca and Mădălina Cărbureanu

Abstract In the era of artificial intelligence and real-time applications, the demand for swift data processing on large volumes has become paramount. This article delves into the critical role of programming languages and frameworks in meeting these requirements. With a focus on web development, the study conducts a comparative analysis of eight classic sorting algorithms—Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort, Heap Sort, Introspective Sort, and Shell Sort—evaluated in two prominent programming languages, PHP and C#. The research emphasizes the importance of selecting programming tools based on performance criteria in the context of modern applications. Next, a new formula for assessing the complexity of sorted words was introduced, aiming to scrutinize its impact on sorting time. Furthermore, a comparative analysis was conducted, evaluating the implementation of sorting in C# using arrays versus lists. The article sought to delineate the intricacies associated with each approach, considering computational efficiency and resource utilization. The discourse delved into the nuanced aspects of word complexity measurement and its consequential effects on the performance of sorting algorithms in the context of C# programming, contributing valuable insights to the ongoing discourse on algorithmic optimization and language-specific implementation choices.

Keywords Sorting · Time complexity · Computer languages · Datasets · Tools

C.-M. Roșca (✉) · M. Cărbureanu
Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, 39 Bucharest Avenue, 100680 Ploieşti, Romania
e-mail: cosmina.rosca@upg-ploiesti.ro

M. Cărbureanu
e-mail: mcarbureanu@upg-ploiesti.ro

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_8

1 Introduction

Within the domain of algorithmic efficiency, the selection of sorting algorithms significantly influences application performance. This article undertakes an exhaustive comparative analysis encompassing eight classical sorting algorithms—namely Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort, Heap Sort, Introspective Sort, and Shell Sort. Extensive testing has been carried out on a substantial volume of data, both in PHP and C# programming languages. By scrutinizing their intricacies and evaluating their performance on large datasets, this exploration aims to shed light on the strengths and weaknesses of each algorithm, offering valuable insights for developers navigating the landscape of sorting algorithms in these two prominent programming languages.

The selection of PHP and C# for algorithmic testing stems from their prowess in web application development, where user interaction demands swift data processing and display. Both languages are renowned for their capacity to create dynamic, responsive interfaces, crucial in web environments. Given the paramount importance of speed in user-facing applications, assessing sorting algorithms in PHP and C# becomes not just a technical exploration but a practical necessity for developers aiming to optimize data handling in web-based user interfaces. This study thus addresses the imperative need for efficient sorting algorithms in the context of real-world, performance-sensitive web applications.

Over time, various approaches to data sorting have been tested [1–4]. However, the scale of data analyzed has remained limited compared to the data requirements of artificial intelligence (AI) and the current demands of the industry.

The paper [5] introduces a novel approach by merging Bubble and Insertion sort methods, estimating a reduced computational complexity of $O(N\sqrt{N})$. The technique involves segmenting the input array, employing a modified Bubble sort on individual blocks, and merging them using a modified Insertion sort. This innovative hybrid outperforms traditional methods, offering a promising alternative for real-world applications with a computational complexity of $O(n^2)$.

The effectiveness of sorting methods can significantly vary depending on the specific characteristics and requirements of an application. Tailoring the choice of sorting algorithm to the unique demands of the application can yield superior results in terms of efficiency and performance [6, 7].

The implementation of classical sorting algorithms on Field-Programmable Gate Arrays (FPGA) involves leveraging the parallel reconfigurable architecture of FPGAs to enhance computational efficiency. This approach aims to exploit parallelism for independent data operations, potentially providing a significant speedup compared to traditional sequential implementations [8–11]. Paper [12] introduces a novel approach to enhancing the efficiency of the Bubble sort algorithm by leveraging parallel swapping on Field-Programmable Gate Arrays (FPGA). The key contribution lies in exploiting the parallel reconfigurable architecture of FPGA to implement Bubble sort, demonstrating a significant speedup compared to the traditional sequential implementation. The methodology involves a left-to-right scan of consecutive

numbers, with swaps occurring based on specific conditions. The results indicate that the parallel implementation outperforms the sequential approach by nearly tenfold across 20 different data inputs. The sequential Bubble sort algorithm is detailed, emphasizing the stepwise comparison and swapping process.

While FPGAs may offer significant advantages in terms of parallelism and energy efficiency, their implementation cost and complexity in conventional server systems can be limiting factors. In practice, more commonplace solutions for high-speed processing and real-time data manipulation in web applications involve the use of multicore processors and hardware accelerators, such as Graphics Processing Units (GPUs), which are easier to integrate into modern server infrastructures. These alternatives provide a balance between performance and practicality, catering to the evolving demands of web-based applications for rapid data processing and responsiveness.

The utilization of large volumes of data in multicore environments constitutes a pivotal domain in contemporary computing. Multicore processors, characterized by multiple processing units on a single chip, present a scalable solution for parallel processing, catering to the intensifying demands of data-intensive applications. In the specific context of large-scale data sorting, this paradigm is exemplified. Sorting large datasets involves intricate algorithms, such as parallel implementations of well-known sorting algorithms (e.g., Quick Sort or Merge Sort), leveraging the concurrent processing capabilities of multiple cores. This strategic use of parallelism optimizes computational efficiency, addressing the challenges posed by the burgeoning scale of data in scientific, industrial, and computational fields. Multiple research studies delve into various facets of sorting algorithms and their optimization for contemporary computing architectures. In [13], three innovative sorting algorithms—Ins-merge, Quick-merge, and Binary Search Tree with duplicate keys (BSTD)—are introduced, specifically designed for efficient large dataset processing on multicore processors. The study in [14] investigates the energy-efficient performance of sorting algorithms on multicore systems, revealing a correlation between improved performance and energy conservation. In [15], a hybrid comparison-based sorting algorithm is proposed, leveraging both multicore CPU and manycore GPU, demonstrating superior results compared to existing approaches. Paper [16] presents Hybrid Dual Pivot Sort, an empirical parallel sorting algorithm for multicore/manycore CPU systems, showcasing significant speedups over traditional methods. In article [17], Parallel Sorting by Approximate Splitting, an enhanced partition method with improved flexibility and efficiency, is introduced. Lastly, the paper [18] explores the application of parallel processing and multicore utilization in sorting and searching applications, highlighting the benefits of shared workloads and faster execution. Together, these works provide valuable insights into the optimization and efficiency of sorting algorithms in the context of modern computing architectures. Despite theoretical advantages, the complexities of implementing and optimizing parallel sorting algorithms on multicore architectures pose practical obstacles, suggesting limitations in their straightforward application for large-scale data sorting.

The majority of research studies often utilize array-based implementations for performance testing [19]. While arrays offer advantages in terms of contiguous

memory access and cache efficiency, the optimization benefits of using lists may vary. Lists provide dynamic memory allocation and ease of insertion/deletion, but their non-contiguous structure can lead to less favorable cache behavior. The choice between arrays and lists depends on the specific requirements of the algorithm and the nature of the data, and empirical testing is crucial to determine the optimal data structure for performance improvements.

Sorting algorithms find various applications. For instance, in addressing Fully Homomorphically Encrypted (FHE) cloud data sorting [20], the proposed Lazy sort algorithm efficiently combines Bubble sort and Insertion sort showcasing feasibility for encrypted processors. In networking protocols [21], the QB sorting approach, inspired by Quick sort and Bubble sort, enhances efficiency, proving adaptable to applications like the Selective Repeat flow control protocol. These innovations highlight the broader implications and advancements in sorting algorithms within computer engineering.

The reference to the relevant study for this work can be found in the paper [22], which utilized a significant number of samples (11 k entries). Through comparative analysis of time and memory usage, the research, conducted using Visual Studio Code and Python, reveals Introspective Sort's (Intro Sort's) superiority in time efficiency, while Heap Sort excels in optimal memory usage. The experimentation involves multiple iterations, demonstrating robust findings on algorithm performance.

Within the scope of this study, sorting algorithms undergo evaluation on a notably larger dataset, focusing on values ranging from 10 to 100 k entries. These entries represent specific Romanian language words extracted from the explanatory dictionary [22]. The objective is to scrutinize the algorithmic performance within the context of varying dataset sizes, allowing for a comprehensive analysis of their efficiency and scalability. This analysis offers valuable insights into how algorithms perform when applied to a dataset with linguistic specificity, enriching our broader understanding of the sorting algorithms' performance characteristics.

2 Sorting Algorithm and Performance Metrics

The need for rapid data processing is particularly evident in the domains of AI and real-time applications. As these applications become integral to various industries, the choice of programming language and framework becomes crucial for achieving optimal performance. To address this challenge, our study analyzes the performance of eight classic sorting algorithms in PHP and C#, two widely used languages in web development.

Through rigorous analysis and testing, it is revealed the limitations and strengths of each sorting algorithm in the chosen programming languages. The obtained results provide valuable insights into the trade-offs associated with algorithmic choices, aiding developers in selecting the most suitable approach for their specific application requirements.

Furthermore, the article addresses the broader implications of these findings for modern web applications. It underscores the significance of optimizing data processing for enhanced user experiences, especially in the context of web-based interfaces. The study advocates for a holistic approach to programming language selection, considering not only syntactical preferences but also performance characteristics crucial for real-time and AI-driven applications.

Sorting algorithms play a fundamental role in computer science, influencing the efficiency of different applications that have to do with large-scale datasets. The present research carries out a comparative analysis of eight sorting algorithms—Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort, Heap Sort, Introspective Sort, and Shell Sort—implemented in both C# and PHP programming languages. The focus is on evaluating the performance metrics and language efficiency of these algorithms across dataset sizes ranging from 10,000 to 100,000 entries.

2.1 Algorithms Implementation

The selected sorting algorithms are essential components in the toolkit of every programmer. Bubble Sort, Insertion Sort, and Selection Sort are classical methods, while Merge Sort and Quick Sort are known for their efficiency in larger datasets. Algorithms are implemented based on the following description:

- Bubble Sort, a simple sorting algorithm, iteratively traverses the list, comparing adjacent elements and exchanging them if they are out of order. This cyclic process continues until the entire list achieves a sorted state. Despite its conceptual simplicity, Bubble Sort exhibits a time complexity of $O(n^2)$, rendering it less efficient for extensive datasets.
- Insertion Sort incrementally constructs the sorted array by selecting each element from the list and placing it into its proper position within the already sorted section of the array. Particularly effective for smaller datasets, Insertion Sort operates in place with a time complexity of $O(n^2)$, yet it can surpass other quadratic algorithms in practical performance scenarios.
- Selection Sort, a sorting algorithm through which an element (minimum or maximum) is placed on its final position in the list. In case of ascending sorting, starting from the first element, the minimum value is identified, value that is placed on the first position by an exchange between the element on the first position and the minimum element. The process continues until the last element (until the list is sorted). Despite sharing a time complexity of $O(n^2)$ with Bubble and Insertion Sorts, Selection Sort generally exhibits less efficiency for most datasets compared to Insertion Sort.
- Merge Sort follows the divide-and-conquer strategy. Merge Sort dissects the unsorted list into 'n' sub-lists, each containing a singular element. Through successive merging of sub-lists, the algorithm generates new sorted sub-lists until only

one remains—the fully sorted list. With a time complexity of $O(n \log n)$, Merge Sort demonstrates stability and consistency across diverse datasets.

- Quick Sort, another divide-and-conquer algorithm, selects a pivotal element and partitions the array into two subsets—elements less than and greater than the pivot. This recursive process culminates in a fully sorted array. While boasting an average time complexity of $O(n \log n)$, Quick Sort may degrade to $O(n^2)$ in the worst case. Renowned for its efficiency and adaptability, it is notable for various application scenarios.
- Heap Sort, a comparison-based sorting algorithm, utilizes a binary heap data structure to construct a heap. It subsequently sorts elements by iteratively extracting the maximum (for max-heap) or minimum (for min-heap) element and placing it at the terminus of the sorted array. Heap Sort achieves a time complexity of $O(n \log n)$ and is characterized by its suitability for a diverse array of input scenarios.
- Introspective Sort, a hybrid sorting algorithm, amalgamates various sorting techniques based on input data characteristics. Primarily commencing with Quick Sort, it seamlessly transitions to Heap Sort when recursion depth surpasses a predefined threshold. This adaptive approach aims to deliver efficient performance across diverse input scenarios, contributing to the ongoing discourse on algorithmic optimization and adaptive sorting strategies.
- Shell Sort, colloquially known as diminishing increment sort, is a comparison-based sorting algorithm that initially sorts pairs of elements with considerable spacing. The algorithm progressively reduces the gap between elements, culminating in a gap of 1, transforming the array into a partially sorted state. The final phase employs an insertion sort to efficiently sort the array. Despite its time complexity of $O(n^2)$, Shell Sort's adaptive nature often renders it competitive in practice, particularly for moderate-sized datasets.

The mentioned algorithms were implemented in both C# and PHP to explore language particularities. Notably, the same pseudocode is implemented in both PHP and C# programming languages. This ensures a consistent and language-agnostic representation of the algorithms, allowing for a fair and comparable evaluation of their performance across the specified datasets.

2.2 Dataset Characteristics

The datasets utilized in this analysis consist of linguistic elements extracted from the Romanian language's explanatory dictionary [22]. Ranging from 10,000 to 200,000 entries, these datasets represent a diverse and sizable collection of words, allowing for a comprehensive examination of algorithmic behavior.

2.3 Performance Metrics

The evaluation criteria include time complexity, space complexity, and overall execution time. By scrutinizing the algorithms' performance across increasing dataset sizes, we aim to provide insights into their scalability and efficiency in handling large-scale data. In each test iteration, the analysis included an examination of both the sample size and the time required for the sorting process. The time is measured in seconds.

The complexity of the sorting algorithm entries inherently influences the time complexity of a sorting operation. Even when utilizing the same programming language, method, and input size, the time can vary based on the inherent complexity of the sorted words. To illustrate, consider two scenarios with a dataset of equal size, but different complexities:

- In a scenario where the words are simple and consist of a limited number of characters, a sorting algorithm may exhibit a faster execution time; for instance, sorting a list of three-letter words in straightforward alphabetical order would likely result in quicker sorting times.
- Conversely, if the dataset contains complex words with varying lengths and intricate character combinations, the sorting algorithm may require more computational effort and time; sorting a list of words with varying lengths (including special characters and numerals), may showcase increased sorting times due to the additional processing required to establish a precise order.

Therefore, the apparent similarity in programming language, method, and input size might not fully encapsulate the time complexity, as the intrinsic complexity of the sorted words plays a pivotal role in determining the efficiency of the sorting process.

The authors of this paper propose a comprehensive formula for calculating dataset complexity that incorporates both the words complexity in the dataset and their lengths. This mathematical expression aims to provide a unified metric for evaluating the computational efficiency of sorting algorithms, taking into account not only the dispersion of the first letters but also the average length of the words within the given dataset. The proposed mathematical formula for calculating dataset complexity in terms of the distance between positions in the alphabet of the first letters and the average lengths of words in a set of words can be expressed as follows (1):

$$\mathrm{DS}_c = \frac{1}{2}\left(\frac{\sum_{i=1}^{n-1}|D_{i+1} - D_i|}{n-1} + \frac{\sum_{i=1}^{n}\mathrm{Average_Length}_i}{n}\right) \tag{1}$$

where:

DS_c is the dataset complexity indicator;

D_i—the position of the first letter of the word i in the alphabet;

$Average_Length_i$—the average length of the word i.

In the following, we will illustrate the process of calculating time complexity using two datasets: Dataset1—{"programmer", "languages", "tool", "framework"} and Dataset2—{"programmer", "program", "set", "setup"}.

For Dataset 1: {"programmer", "languages", "tool", "framework"}, the DS_c value is:

$$DS_1 = \frac{1}{2}\left(\frac{|12-16|+|20-12|+|6-20|}{3} + \frac{(10+9+4+9)}{4}\right) = 8.33 \quad (2)$$

For Dataset 2: {"programmer", "program", "set", "setup"}, the DSc value is:

$$DS_2 = \frac{1}{2}\left(\frac{|16-16|+|19-16|+|19-19|}{3} + \frac{(10+7+4+5)}{4}\right) = 1.28 \quad (3)$$

Comparing the results, Dataset 1 has a Dataset Complexity of 8.33, meantime Dataset 2 has a Dataset Complexity of 1.28. These values provide a quantitative measure of the Dataset complexity for each dataset, considering both the dispersion of the first letters and the average length of the words.

Dataset Complexity serves as a comparative metric for assessing the complexity level of datasets sorted by algorithms. This allows for a proper analysis of Time Complexity for an equal number of entries.

2.4 Language Efficiency

Implementing the algorithms in both C# and PHP enables a comparative analysis of language efficiency. Differences in syntax, memory management, and other language-specific features may influence the algorithms' overall performance. This exploration sheds light on the impact of language choice in sorting algorithm implementations.

In the following section, relevant tests and results will be presented to assess the eight sorting analyzed algorithms.

3 Tests and Results

In this research, the eight sorting algorithms were implemented in both C# and PHP languages. The datasets were extracted from a database containing 1,187,025 words in the Romanian language [22]. It is noteworthy that the Romanian language includes special characters, adding complexity to the sorting process. The tests were conducted on data samples ranging from 10 to 100 k words. Additionally, all tests were executed in the same environment, utilizing an 11th Gen Intel(R) Core(TM) i5-1135G7@2.40 GHz 2.42 GHz processor with 8.00 GB of RAM.

3.1 C# Versus PHP Performance Across Sorting Algorithms

A number of 1440 tests were conducted for datasets comprising words ranging from 10 to 100 k. Table 1 displays the average time required for sorting datasets grouped by intervals using the C# programming language. In these results, it can be observed that the Quick Sort method yields the best outcomes in terms of execution time. It is important to emphasize that these sorting methods were implemented using array data structures.

In Tables 1 and 2, the algorithms are abbreviated as follows: BS—Bubble Sort, IS—Insertion Sort, SS—Selection Sort, MS—Merge Sort, QS—Quick Sort, HS—Heap Sort, IPS—Introspective Sort, and SH—Shell Sort. The presented results highlight the performance of sorting algorithms implemented in both C# and PHP languages, with time measurements in seconds.

The C# analysis provides the following interpretation of sorting algorithm performance:

- Bubble Sort (BS) shows an increasing trend in execution time with larger datasets; it becomes less efficient as the dataset size grows;
- Insertion Sort (IS) exhibits better performance than Bubble Sort but still struggles with larger datasets;
- Merge Sort (MS) demonstrates consistent and efficient performance, even with larger datasets; it stands out as a reliable sorting option;
- Quick Sort (QS) shows remarkable efficiency, with the lowest execution times across all dataset sizes; it emerges as the top-performing algorithm in C#;
- Selection Sort (SS) presents a steady increase in execution time; it performs adequately but is outperformed by Quick Sort;
- Heap Sort (HS) exhibits consistent and relatively efficient performance across the dataset; the gradual increase in execution times indicates a stable and predictable behavior.

Table 1 Performance metrics for sorting algorithms in C# on different datasets sizes

No. samples (k)	Algorithm							
	BS	IS	MS	QS	SS	HS	IPS	SH
10–20	15.849	6.647	0.0428	0.0314	13.055	0.0570	0.5300	0.0350
20–30	34.790	14.845	0.0614	0.0468	28.691	0.0660	0.3250	0.0770
30–40	86.330	35.158	0.1088	0.0823	67.350	0.0980	0.3440	0.1120
40–50	126.06	56.269	0.1249	0.1067	104.90	0.1170	0.3850	0.1530
50–60	183.28	79.131	0.1588	0.1125	146.53	0.1560	0.5250	0.1880
60–70	246.63	113.63	0.1795	0.1535	222.35	0.1880	0.6110	0.2500
70–80	305.59	132.13	0.1994	0.1725	252.42	0.2210	0.7220	0.2600
80–90	392.54	169.84	0.2064	0.1802	322.26	0.2600	3.1690	0.2890
90–100	623.26	239.42	0.2582	0.2361	461.02	0.2860	1.3280	0.3160

Table 2 Performance metrics for sorting algorithms in PHP on different datasets sizes

No. samples (k)	Algorithm							
	BS	IS	MS	QS	SS	HS	IPS	SH
10–20	41.826	18.033	29.106	0.2164	33.646	0.4153	0.2035	0.2277
20–30	366.64	159.43	223.74	0.5691	198.23	0.6188	0.6111	0.5749
30–40	470.08	207.42	47.747	0.5707	271.58	0.9694	0.8633	0.6726
40–50	624.43	277.73	228.59	0.8416	509.47	1.2702	0.9109	0.8880
50–60	1942	356.68	260.72	0.9844	1762.2	1.6600	1.2787	1.0453
60–70	1998	477.25	365.37	1.1491	1791.7	1.9606	1.3399	1.2664
70–80	2009	646.73	401.54	1.2732	1960.0	2.4099	1.5710	1.5596
80–90	2365	1101.1	451.91	1.6048	2117.6	2.7447	1.7941	1.7496
90–100	2662	1161	474.21	1.8796	2312.2	3.4711	1.9767	2.2858

- Introspective Sort (IPS) demonstrates varied execution times across the dataset;
- Shell Sort (SH) indicates competitive performance, particularly for datasets with moderate sizes.

The same algorithms were executed in the PHP command line. Table 2 presents the average time required for sorting various datasets grouped by intervals, allowing for performance analysis in comparison to the C# language.

The PHP analysis provides the following interpretation of sorting algorithm performance:

- Bubble Sort (BS) exhibits high execution times, indicating suboptimal performance;
- Insertion Sort (IS) struggles with efficiency as the dataset size increases;
- Merge Sort (MS) displays varying performance, with efficient execution for mid-sized datasets; it faces challenges with larger datasets;
- Quick Sort (QS) maintains competitive efficiency, with lower execution times than Bubble and Insertion Sort; it shows a consistent performance across different dataset sizes;
- Selection Sort (SS) yields varying results, with more favorable performance for mid-sized datasets; it also faces challenges with larger datasets, similar to Bubble and Insertion Sort;
- Heap Sort (HS) exhibits an overall increasing trend; as the dataset size increases, the algorithm's performance becomes less favorable;
- Introspective Sort (IPS) showcases a general increasing trend in execution times;
- Shell Sort (SH)—the execution times are conspicuously higher in certain cases, suggesting a potential performance drawback under certain circumstances.

The indisputable leader in terms of efficiency is Quick Sort (QS), which exhibits exceptional performance in both C# and PHP languages, but the average times for QS algorithm in C# are smaller than the ones obtained in PHP.

So, C# demonstrates superior performance with Quick Sort. This implies that C# users can benefit from an enhanced sorting experience, especially when dealing with extensive datasets. On the other hand, PHP showcases acceptable performance levels but tends to lag behind C# in terms of algorithmic efficiency.

Overall, the choice of programming language significantly influences the performance of sorting algorithms. C#, particularly when coupled with Quick Sort, emerges as a robust option for achieving optimal efficiency.

3.2 Dataset Complexity Impact on Quick Sort

The Quick Sort's adaptability to different data distributions contributes to its widespread usage. Quick Sort is known for performing exceptionally well on partially sorted or almost sorted arrays, making it advantageous in real-world scenarios where data may not be uniformly distributed. Despite its numerous advantages, Quick Sort is not without its limitations. The sensitivity to the initial ordering of the array and the potential for a worst-case time complexity of $O(n^2)$ necessitates careful consideration of the dataset characteristics.

In this study, 1000 tests were conducted in C# to evaluate the performance of the Quick Sort method under different dataset complexities using the formula (1) proposed by the authors and keeping constant the number of words at 10 k.

The initial expectation was that as the dataset complexity increased, the sorting time would proportionally increase, reflecting the algorithm's sensitivity to dataset complexities. However, the results challenged this assumption. In all 1000 tests, Quick Sort demonstrated consistent and efficient sorting times, regardless of dataset complexity.

The associated histogram, presented in Fig. 1, exhibits a relatively uniform distribution. The minimal variations observed in this histogram can be attributed to the limited dataset size, where the sorting efficiency of the algorithm is less influenced by dataset complexity.

Upon delving into the intricacies of Quick Sort, its divide-and-conquer approach emerged as a crucial factor contributing to this robust behavior. The algorithm's efficiency lies in its ability to break down the sorting process into smaller subproblems, mitigating the impact of word complexity on the overall time complexity.

As a result, the unexpected consistency in Quick Sort's performance across different word complexities underscores its robustness. This behavior can be attributed to its divide-and-conquer strategy and pivot-based partitioning, enabling it to navigate efficiently through datasets with varying word lengths. Quick Sort's stability across diverse scenarios reaffirms its status as a reliable and adaptable sorting algorithm, showcasing its resilience even in the face of varying word complexities.

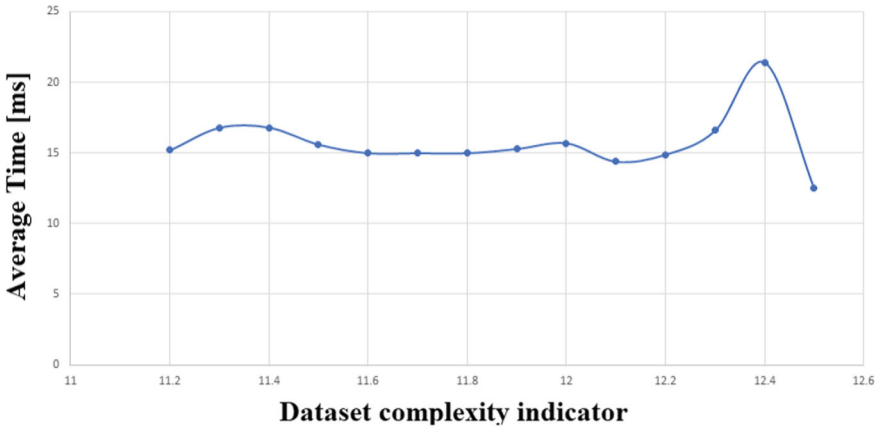


Fig. 1 1000 tests histogram of the complexity datasets

3.3 Array Versus List Performance on Quick Sort

The array-based implementation of Quick Sort relies on the direct access of elements through indices. In arrays, memory is contiguous, enabling quick and efficient access to elements. Quick Sort exploits this feature, making it well-suited for array-based implementations. Sorting a range of elements involves swapping values directly within the array, enhancing locality, and reducing the number of memory accesses.

Conversely, linked lists offer a dynamic structure where elements are stored in nodes with pointers to the next element. The absence of direct indexing in linked lists poses a challenge for Quick Sort's pivot-based partitioning. Navigating through the list requires traversing pointers, resulting in a less straightforward access pattern compared to arrays. This inherent characteristic introduces potential variations in execution time.

When comparing execution times, arrays generally exhibit faster sorting with Quick Sort. The direct access capability allows for efficient manipulation of elements during the partitioning phase, contributing to reduced time complexity. However, it is essential to acknowledge that linked lists possess advantages in scenarios where dynamic memory allocation and flexibility in inserting and deleting elements are paramount.

In terms of space complexity, arrays and linked lists differ significantly. Arrays allocate memory based on the anticipated size, while linked lists dynamically allocate memory as elements are added. This difference influences cache efficiency and impacts overall performance.

In the context of this research, a comprehensive examination was conducted by implementing identical C# code using both arrays and lists. Subsequently, 3000 tests were executed across three distinct datasets: 10, 10 k, and 1000 k words. The results, presented in Table 3, reveal superior execution times for array implementation compared to lists.

Table 3 Comparative average execution times for array and linked list

No. words	Quick sort	
	Array method—Execution time [ms]	List method—Execution time [ms]
10	0.009	0.009
10 k	15.225	17.762
1000 k	3283.110	3593.675

Table 3 provides a succinct overview of the average execution times, expressed in milliseconds, for each dataset category. The employment of arrays consistently yields superior results across all three test scenarios, comprising sets of 10, 10 k, and 1000 k words. This pattern underscores the efficiency of arrays-based implementations compared to lists when executing sorting algorithms, particularly Quick Sort.

These sorting algorithms face limitations in applications designed for graphical user interfaces, as sorting large volumes of data may result in long wait times or application crashes. Overcoming this challenge involves identifying methods to handle large datasets by sorting them into smaller sub-packages and displaying data accordingly. Existing algorithms may struggle to efficiently manage datasets in real time, making them less suitable for user-friendly graphical interfaces. To address this issue, it is essential to explore innovative approaches for sorting and presenting data in more manageable subsets. This challenge will optimize the sorting process, ensuring a smooth and responsive user experience even when dealing with significant data volumes.

As a result, for the Quick Sort algorithm, utilizing array structures yields superior results in terms of execution time, especially for large datasets.

4 Conclusions

In the present paper, a comprehensive comparative analysis was conducted between two prominent programming languages within the Web development domain. The study focused on the evaluation of eight sorting algorithms, presenting an examination of their performance across 1440 tests. The findings unequivocally demonstrated the superiority of the C# programming language in terms of speed, with the Quick Sort algorithm emerging as the most suitable for handling large volumes of data.

Subsequently, an in-depth exploration was undertaken to assess the impact of dataset complexity on the sorting speed. Remarkably, it was revealed that the complexity of the dataset exerted no discernible influence on the temporal aspects of the sorting process. This result suggests a robust and consistent performance of the chosen algorithms across different datasets.

Investigations delved into the ramifications of data structure choices on the speed tests. The analysis pinpointed a disadvantageous outcome associated with the utilization of arrays in conjunction with the Quick Sort method when compared to the deployment of lists. This observation underscores the significance of thoughtful data structure selection in optimizing the efficiency of sorting algorithms.

Future investigations will analyze sorting methods within frameworks compared to conventional approaches. As future research, the exploration of utilizing multiple cores for data sorting aims to enhance algorithmic performance. This direction responds to the escalating demand for efficiency in processing vast and intricate datasets. Leveraging multicore technology allows the concurrent distribution of sorting tasks, potentially accelerating execution speed. This technological evolution aligns with the trend of harnessing modern hardware resources to efficiently handle computationally intensive tasks, maximizing data sorting efficiency in academic and industrial realms.

References

1. Min W (2010) Analysis on bubble sort algorithm optimization. In: 2010 international forum on information technology and applications. Kunming, China, pp 208–211. https://doi.org/10.1109/IFITA.2010.9
2. Rashidy R, Yousefpour S, Koohi M (2011) Parallel bubble sort using stream programming paradigm. In: 2011 5th international conference on application of information and communication technologies (AICT). Baku, Azerbaijan, pp 1–5. https://doi.org/10.1109/ICAICT.2011.6111024
3. Setiawan R (2016) Comparing sorting algorithm complexity based on control flow structure. In: 2016 international conference on information management and technology (ICIMTech). Bandung, Indonesia, pp 224–228. https://doi.org/10.1109/ICIMTech.2016.7930334
4. Rosiene JA, Rosiene CP (2019) Design patterns for sorting algorithms. In: 2019 IEEE frontiers in education conference (FIE). Covington, KY, USA, pp 1–4. https://doi.org/10.1109/FIE43999.2019.9028379
5. Paul T (2022) Enhancement of bubble and insertion sort algorithm using block partitioning. In: 2022 25th international conference on computer and information technology (ICCIT). Cox's Bazar, Bangladesh, pp 412–417. https://doi.org/10.1109/ICCIT57492.2022.10055404
6. Hayaran I, Khanna P (2016) Couple sort. In: 2016 fourth international conference on parallel, distributed and grid computing (PDGC). Waknaghat, India, pp. 390–393. https://doi.org/10.1109/PDGC.2016.7913226
7. Yang Y, Yu P, Gan Y (2011) Experimental study on the five sort algorithms. In: 2011 second international conference on mechanic automation and control engineering. Inner Mongolia, China, pp 1314–1317. https://doi.org/10.1109/MACE.2011.5987184
8. Lobo J, Kuwelkar S (2020) Performance analysis of merge sort algorithms. In: 2020 international conference on electronics and sustainable communication systems (ICESC). Coimbatore, India, pp 110–115. https://doi.org/10.1109/ICESC48915.2020.9155623
9. Fahad Alif AFM, Islam SMR, Deb P (2019) Design and implementation of sorting algorithms based on FPGA. In 2019 international conference on computer, communication, chemical, materials and electronic engineering (IC4ME2). Rajshahi, Bangladesh, pp 1–4. https://doi.org/10.1109/IC4ME247184.2019.9036675
10. Ul Sadad N, Afrin A, Islam Mondal MN (2021) Implementation of scalable non-recursive merge sort on FPGA. In: 2021 international conference on computer, communication, chemical,

materials and electronic engineering (IC4ME2). Rajshahi, Bangladesh, pp 1–4. https://doi.org/10.1109/IC4ME253898.2021.9768498
11. Abdelrasoul M, Shaban AS, Abdel-Kader H (2021) FPGA based hardware accelerator for sorting data. In: 2021 9th International japan-africa conference on electronics, communications, and computations (JAC-ECC). Alexandria, Egypt, pp 57–60. https://doi.org/10.1109/JAC-ECC54461.2021.9691432
12. Lipu AR, Amin R, Islam Mondal MN, Al Mamun M (2016) Exploiting parallelism for faster implementation of Bubble sort algorithm using FPGA. In: 2016 2nd international conference on electrical, computer & telecommunication engineering (ICECTE). Rajshahi, Bangladesh, pp 1–4. https://doi.org/10.1109/ICECTE.2016.7879576
13. Suresh A, George AK (2018) Performance analysis of various combination sorting algorithms for large dataset to fit to a multi-core architecture. In: 2018 second international conference on inventive communication and computational technologies (ICICCT). Coimbatore, India, pp 51–56. https://doi.org/10.1109/ICICCT.2018.8472956
14. Zecena I, Zong Z, Ge R, Jin, T, Chen, Z, Qiu, M (2012) Energy consumption analysis of parallel sorting algorithms running on multicore systems. In: 2012 international green computing conference (IGCC). San Jose, CA, USA, pp 1–6. https://doi.org/10.1109/IGCC.2012.6322290
15. Banerjee DS, Sakurikar P, Kothapalli K (2013) Fast, scalable parallel comparison sort on hybrid multicore architectures. In: 2013 IEEE international symposium on parallel & distributed processing, workshops and Phd Forum. Cambridge, MA, USA, pp 1060–1069. https://doi.org/10.1109/IPDPSW.2013.129
16. Taotiamton S, Kittitornkun S (2017) Parallel hybrid dual pivot sorting algorithm. In: 2017 14th international conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON). Phuket, Thailand, pp 377–380. https://doi.org/10.1109/ECTICon.2017.8096252
17. Wu W, Wu P, Shang W, Fang Y (2010) Parallel sorting by approximate splitting for multi-core processors. In: 2010 third international joint conference on computational science and optimization. Huangshan, China, pp 308–311. https://doi.org/10.1109/CSO.2010.72
18. Sujatha K, Rao PVN, Rao, AA, Sastry VG, Praneeta V, Bharat RK (2015) Multicore parallel processing concepts for effective sorting and searching. In: 2015 international conference on signal processing and communication engineering systems. Guntur, India, pp 162–166. https://doi.org/10.1109/SPACES.2015.7058238
19. Bijoy MHI, Hasan MR, Rabbani M (2020) RBS: a new comparative and better solution of sorting algorithm for array. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT). Kharagpur, India, pp 1–5. https://doi.org/10.1109/ICCCNT49239.2020.9225685
20. Chatterjee A, Sengupta I (2015) Windowing technique for lazy sorting of encrypted data. In: 2015 IEEE conference on communications and network security (CNS). Florence, Italy, pp 633–637. https://doi.org/10.1109/CNS.2015.7346878
21. Saxena S (2018) QB sort: an efficient computational algorithm for advance networks. In: 2018 3rd international conference on internet of things: smart innovation and usages (IoT-SIU), Bhimtal, India, pp 1–4. https://doi.org/10.1109/IoT-SIU.2018.8519911
22. Dexonline. https://wiki.dexonline.ro/wiki/Info_rma%C8%9B_ii#%20D%20esc%20%C4%83rcare, Last accessed 18 Nov 2023

Genetic Algorithm-Based Probability Distribution Fitting for Improved Classification of Echocardiogram Regions

Dennis Thomas, K. M. Subhash, and K. V. Shihabudheen

Abstract Ultrasound imaging has emerged as a pivotal modality in medical diagnostics due to its safety, real-time capabilities, and absence of ionizing radiation. This paper presents a comprehensive analysis of echocardiogram image segments obtained from tissue, blood, valve, and combined tissue and blood regions, using curve-fitting techniques utilizing genetic algorithm. The genetic algorithm convergence data, including generation number, cumulative fitness function evaluations, R^2 values, and fitted curve parameters, are documented and analyzed. Furthermore, the optimization results for various probability distribution functions (PDFs) are presented, offering insights into the mean and standard deviation of generation counts, function evaluations, R^2 values, and fitted distribution parameters for Gamma, Gaussian, exponential, and bimodal distributions. The findings show the suitability of the Gamma distribution for modeling the histograms of ultrasound image segments of blood regions, supported by R^2 values ranging from 0.71 to 0.99. Additionally, the study evaluates the goodness of fit for three PDFs—Gaussian, exponential, and bimodal—for the tissue, valve, and combined tissue and blood region, providing insights into the performance of these distributions. Metrics such as R^2, MAE, AIC, and BIC are presented for each case and compared to evaluate their ability for classification tasks. The work underscores the potential for more specialized PDF models, incorporating hybrid approaches and machine learning techniques, to further enhance the analysis of cardiac ultrasound image segments.

Keywords Echocardiogram · Ultrasound imaging · Genetic algorithms · Probability distribution functions · Image segmentation

D. Thomas (✉) · K. M. Subhash · K. V. Shihabudheen
Department of Electrical Engineering, National Institute of Technology, Calicut, Kerala, India
e-mail: dennisthom@gmail.com

K. M. Subhash
e-mail: subhash@nitc.ac.in

K. V. Shihabudheen
e-mail: shihabudheen@nitc.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_9

1 Introduction

Ultrasound has become indispensable in medical diagnostics due to its safety, expediency, and non-ionizing nature, making it suitable for various applications, including pediatric monitoring as well as liver, kidney, and thyroid examinations [1]. In medical image analysis, a critical objective involves precise object delineation through segmentation and accurate label descriptions via classification [2].

Although cardiac function is routinely assessed using ultrasound, its lower image quality compared to CT and MRI poses challenges for automated analysis [3]. Echocardiography, utilizing ultrasound waves, is vital for cardiac assessment. Speckle noise significantly impacts image resolution and contrast [4]. Although blood region generally produces minimal backscatter, a speckle-like texture is often present. It could be backscatter from tissue into the side-lobes of the transducer and is referred as stationary speckle [5].

Analyzing 2D echocardiographic images is crucial for cardiac assessment, relying on low-level processing like segmentation and tracking. For example, precise delineation of the left ventricular endocardium in both end diastole (ED) and end systole (ES) is crucial for extracting parameters such as the ejection fraction (EF) of the left ventricle (LV) [6]. Accurate segmentation enables precise measurement of parameters like ejection fraction, stroke volume, and wall motion abnormalities, aiding in identifying pathological changes.

2 Related Work

Current research in echocardiography aims to advance imaging techniques and analysis methods to enhance diagnostics and patient outcomes. Studies explore refining image acquisition protocols, developing automated segmentation algorithms, and utilizing statistical models for valuable information extraction [7].

The application of probability density functions (PDFs) to heart-echo images, particularly using the Gamma distribution, enhances the analysis of cardiac ultrasound data [5]. Such PDF-based approaches offer a systematic and quantitative analysis of echo images, facilitating better understanding and diagnosis of cardiac and other pathologies [8].

Recent literature delves into PDF applications in various medical imaging domains. Studies leverage PDF-based approaches for characterizing tissue properties, anomaly detection, and quantifying physiological parameters [9]. Examples include a PDF modeling technique for liver ultrasound texture analysis to detect fibrosis [10] and the use of Gaussian mixture models (GMMs) for cardiac structure segmentation and myocardial deformation parameter estimation [11].

3 Preliminaries

Segmenting cardiac ultrasound images relies on establishing robust statistical models for inherent speckle characteristics. Literature proposes various PDFs [5], informed by empirical observations and statistical inferences, to capture diverse statistical behaviors in cardiac ultrasound data. Assessing the goodness of fit is crucial, using metrics derived from comparing modeled PDFs to actual histogram distributions. Higher values indicate a more favorable alignment between chosen PDFs and speckle statistics. Efficacious segmentation hinges on minimizing misclassification errors, necessitating evaluation of individual PDF fits and intra-regional comparisons. A low cumulative misclassification rate across the entire image is essential for reliable segmentation outcomes, extending beyond individual regions.

4 Proposed Approach

Effective cardiac ultrasound segmentation, especially when using maximum likelihood (ML) and maximum a posteriori (MAP) approaches, depends on tailored statistical models for speckle characteristics. Zhong Tao et al. [5] assess goodness of fit in tissue and blood regions using four curve-fitting models, employing a generalized likelihood ratio test (GLRT) for classification. Genetic algorithms (GA) for curve fitting offer advantages in global optimization and adaptability over multiple iterations. Incorporating GAs with metrics like R^2, mean squared error, AIC, and BIC enables a nuanced and globally optimized fit, promoting robust analysis. Evaluating goodness of fit is crucial, minimizing misclassification errors, and selecting PDFs adaptable to diverse cardiac regions.

This paper categorizes cardiac ultrasound regions using Gamma, Gaussian, exponential, and bimodal PDFs, creating a framework capturing variations in speckle statistics. GAs optimize PDF selection and parameterization for diverse cardiac ultrasound regions, aiming for effective modeling strategies and accurate segmentation outcomes through systematic experimentation and evaluation.

5 GA-Based PDF Model Parameter Estimation

5.1 Data Collection and Preprocessing

The cardiac ultrasound video, acquired using a standard IE 33 (Philips ultrasound, Bothell, USA) ultrasound machine, with a phased-array transducer (X7-2t TEE xMATRIX) operating at a frequency of 3–7 MHz, consisted multiple sequences of 150 frames (600 × 800 pixels, 8-bit grayscale). The video was converted to individual images, preprocessed by cropping to 500 × 400 pixels to exclude machine

labels, ensuring focused analysis on the heart echo portion. All images were resized uniformly to 500 × 400 pixels for dataset consistency.

5.2 Image Sub-segments and Histogram Representation

The resulting image was manually delineated into four regions (blood, tissue, valve, and tissue–blood) using visual observation, creating bounding rectangles around each portion. Each segmented region was approximately 50 × 50 pixels. This manual approach was chosen for accurate identification and isolation of distinct regions of interest within the image.

The blood region, with darker grayscale pixels, is modeled using the Gamma PDF, suitable for bounded data exhibiting skewness. For the tissue region, covering the full grayscale spectrum, we opt for the Gaussian PDF, ideal for bell-shaped distributions. The valve region, characterized by a few brighter pixels and mostly 0 intensity, is modeled using the exponential PDF for its decay-like distribution. The tissue and blood region, featuring both darker and brighter pixels, employs the bimodal Gaussian PDF with two peaks to capture multimodal distributions. Chosen PDFs accurately depict pixel intensity characteristics within each region. Utilizing MATLAB, a trial and error approach revealed that a 20-bin histogram accurately depicts pixel intensity distribution, which is adjustable by the user. Midpoint values of bins are calculated, normalized to form a probability density distribution, and linearly interpolated to a 256-point array for precise representation. The resulting curve, reflecting probability density against pixel intensity values (0–255), is pivotal for subsequent analysis and curve-fitting algorithm accuracy. The user-flexible bin adjustment ensures adaptability to varying image characteristics.

5.3 Genetic Algorithm

Curve fitting on histograms from different cardiac ultrasound segments is executed using the genetic algorithm (GA), a stochastic optimization technique inspired by biological evolution. GA (Fig. 1) iteratively refines solutions through crossover, mutation, and selection, aiming to minimize the squared error between the histogram and the fitted PDF. GA optimizes the curve-fitting process by evolving candidate solutions toward the best fit. Representations of solutions, whether as vectors for continuous values or bit strings for discrete bits, are pivotal in guiding the selection of genetic operators.

Crossover. GAs employ crossover operators like the 2-point crossover, which mixes genetic material from parents. The 2-point crossover splits candidates at 2 positions, assembling them alternately into a new solution, motivated by the potential to combine successful parts from both parents. Extensions to more points are possible,

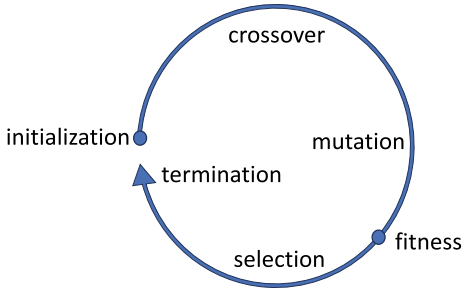


Fig. 1 Mechanics of arriving at optimal solution by GA

with various approaches like arithmetic mean, dominant, and uniform. In our GA, we use a "scattered" n-point crossover, where a random binary vector determines gene selection from parents to compose offspring genetics.

Mutation. Mutation operators induce alterations through random perturbations, guided by requisites like reachability, unbiasedness, and scalability. We adopt Gaussian mutation, adding a random number from a Gaussian distribution (mean 0) to each parent vector entry, with a default standard deviation set to 1.

Selection. Genetic algorithms involve choosing the best offspring solutions as parents for the next generation. In our algorithm, we opt for stochastic uniform selection, representing each parent proportionately on a line based on their scaled value. The algorithm progresses in uniform steps, designating parents at each step.

Termination. Termination criteria include a maximum of 3000 generations and a fitness function tolerance of 10^{-8}, ensuring an optimal solution within an acceptable timeframe in our study. We use four different PDFs for fitting the histograms.

Gamma distribution. *The Gamma distribution is commonly used to model the amplitude of a signal that is affected by random noise.* In the context of medical imaging, it has been used to model the distribution of speckle pixel intensities in blood regions [5]. The PDF of the Gamma distribution is given by [12]:

$$\text{gamma}(x, k) = k3 * \frac{1}{\Gamma(k2) * k1^{k2}} x^{k2-1} \mathrm{e}^{\frac{-x}{k1}} \tag{1}$$

where x is the random variable, $k2 > 0$ is the shape parameter, $k1 > 0$ and $k3 > 0$ are the scale parameters, and $\Gamma(\alpha)$ is the gamma function.

Gaussian distribution. In this study, the Gaussian distribution is used to model the distribution of pixel intensities in tissue regions. The PDF of the Gaussian distribution is given by:

$$g(x, k) = k1 * \mathrm{e}^{\frac{-(x-k2)^2}{2*k3^2}} \tag{2}$$

where x is the random variable, and $k1$, $k2$, $k3$ are the scale, mean, and standard deviation parameters.

Exponential distribution. In this study, exponential distribution is used to model the distribution of pixel intensities in valve regions. The PDF of the exponential distribution is given by:

$$e(x, k) = k1 * \mathrm{e}^{-k2*x} + k3 \tag{3}$$

where x is the random variable and $k1$, $k2$, $k3$ are the scale, rate and shift parameters.

Bimodal distribution. The bimodal distribution is used to model data that has two distinct peaks or modes. In this study, it is used to model the distribution of pixel intensities in regions comprising both tissue and blood. The PDF of the bimodal distribution is given by:

$$b(x, k) = k1 * \mathrm{e}^{\frac{-(x-k2)^2}{2*k3^2}} + k4 * \mathrm{e}^{\frac{-(x-k5)^2}{2*k6^2}} \tag{4}$$

where x is the random variable, $k2$ and $k5$ are the means of the two modes, $k3$ and $k6$ are their standard deviations, and $k1$ and $k4$ are the corresponding scale parameters.

The GA's fitness function, based on PDFs representing the image segment's distribution, considers Gamma, Gaussian, exponential, and bimodal distributions with 3, 3, 3, and 6 unknown parameters, respectively. Solution vectors for these PDFs match the fitness function's parameter count, ensuring accurate modeling. The GA input included [0, inf] as bounds for these parameters, and a 10^{-8} tolerance for consecutive iteration differences was used. The algorithm was set to terminate at maximum 3000 generations for optimal fit to observed data. The algorithm was implemented using MATLAB.

5.4 Dataset and Evaluation Metrics

The dataset comprises 100 randomly selected frames from video recordings of trans-esophageal echocardiogram (TEE) images of the heart. From each frame, blood (B1-B100), tissue (T1-T100), valve (V1-V100), and tissue and blood (BT1-BT100) are manually segmented. Multiple frames and segments enable a better evaluation of curve fitting using GAs.

To evaluate the goodness of fit, we compute the R^2 value for the histogram obtained in Sect. 5.2. It is a statistical measure of how well the curve fits the data.

$$R^2 = \frac{\mathrm{SSR}}{\mathrm{SST}} = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} \tag{5}$$

Sum of squared regression (SSR), sum of squared total error between the histogram curve and its mean (SST), and sum of squared error/residuals between

the histogram curve and fitted curve (SSE) are employed for R^2 computation. R^2, ranging from 0 to 1, indicates fit quality, with 1 denoting a perfect fit and 0 implying no fit.

Akaike information criterion (AIC) is a measure of the trade-off between goodness of fit and the complexity of the model. Bayesian information criterion (BIC) tends to penalize complex models more heavily [13].

$$\mathrm{AIC} = -\frac{2}{N} * \log lik + 2\frac{k}{N} \tag{6}$$

$$\mathrm{BIC} = -2{*}\log lik + k{*}\log(N) \tag{7}$$

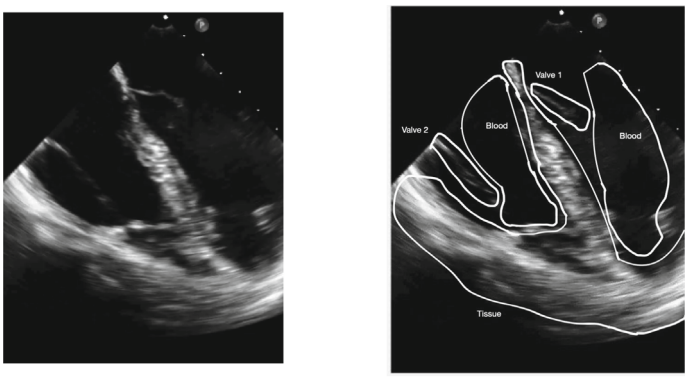
Here, N is the number of data samples, k is the number of model parameters, and $\log lik$ is the maximized log-likelihood of the estimated model. Lower values of AIC and BIC indicate better fit.

6 Results

6.1 Preprocessed Image and Image Segments

The 8-bit grayscale cardiac ultrasound image used in this study (Fig. 2) is cropped to 500 × 400 pixels, showcasing the heart's four chambers and valves.

In order to perform curve fitting on the cardiac ultrasound images, the preprocessed image was segmented into four different regions: blood, tissue, valve, and tissue plus blood, as shown in Fig. 3.



(a) (b)

Fig. 2 8-bit grayscale image of echocardiogram **a** original image **b** regions of interest

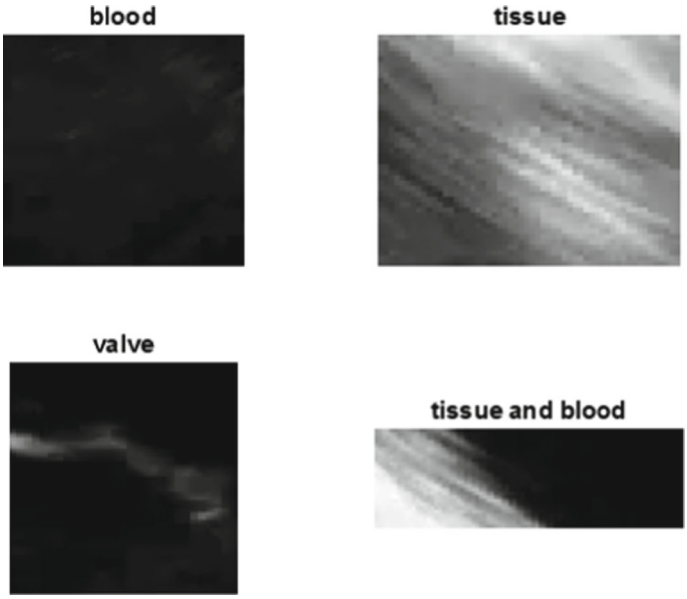


Fig. 3 Image segments—blood, tissue, valve, and tissue and blood

The delineation/segmentation process was performed manually, using bounding rectangle as explained in Sect. 5.2.

6.2 Histograms

In order to analyze the characteristics of the segmented regions, histograms (Sect. 5.2) were plotted using MATLAB code (Fig. 4a–d).

After experimentation with various bin sizes, it was determined that 20 bins were appropriate to visually depict the intensity distributions of the curves for the 50 × 50 pixel images.

6.3 Fitted Curves and Evaluation Metrics

Figure 5a–d shows curve-fit results using the GA. With parameter bounds set from 0 to infinity, GA achieved convergence well before the 3000 generation limit for most samples.

Table 1 displays results from five representative samples among the 100 samples. It includes dataset labels, the generation number at GA convergence, cumulative fitness function evaluations at convergence, R^2 values from curve fitting, and fitted

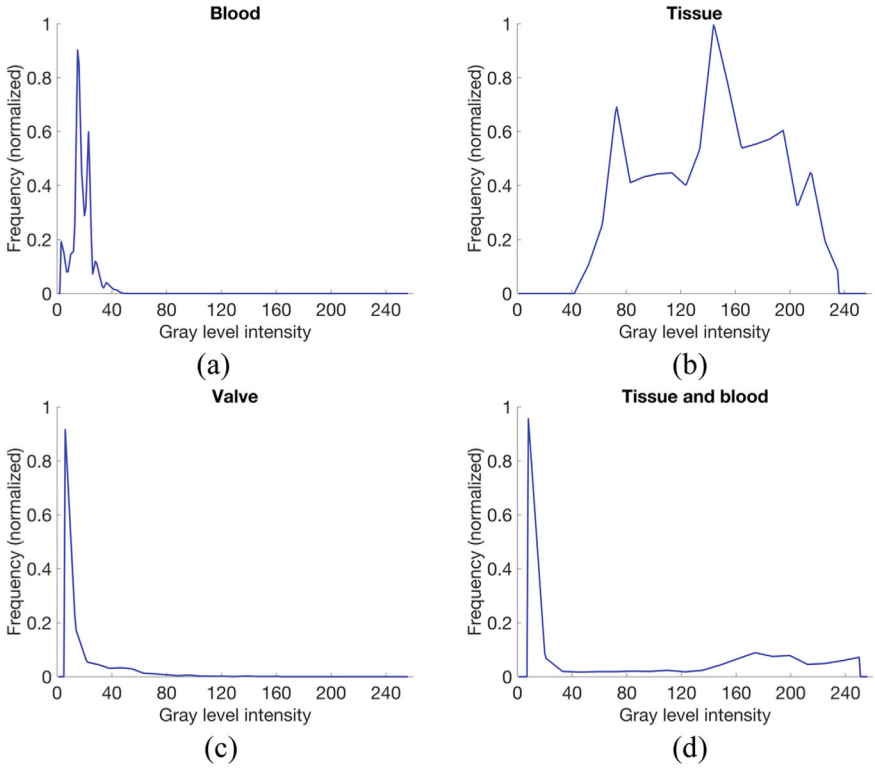


Fig. 4 Typical histograms of the 4 image segments with number of bins = 20, (**a**) blood (**b**) tissue (**c**) valve (**d**) tissue and blood

model parameters. This dataset overview facilitates a comprehensive analysis of GA performance on each dataset.

Table 2 shows the optimization results of the various PDFs studied, for all the samples. This is formed from data obtained for each of the regions as shown before in Table 1. The table gives the mean (μ) and standard deviation (σ) of generation counts, function evaluations, R^2 values, and fitted distribution parameters for Gamma, Gaussian, exponential, and bimodal distributions.

The GA demonstrates convergence well before the typical limit of 3000 generations, as evident from the observed generation counts (685, 750, 1801, and 1938), indicating efficient optimization within a considerably lower number of iterations. The high R^2 values (0.87 to 0.94) across the different distributions underscore the robustness and effectiveness of the optimization algorithm in achieving a significant goodness of fit. The large function evaluation counts showcase the comprehensive exploration and refinement carried out.

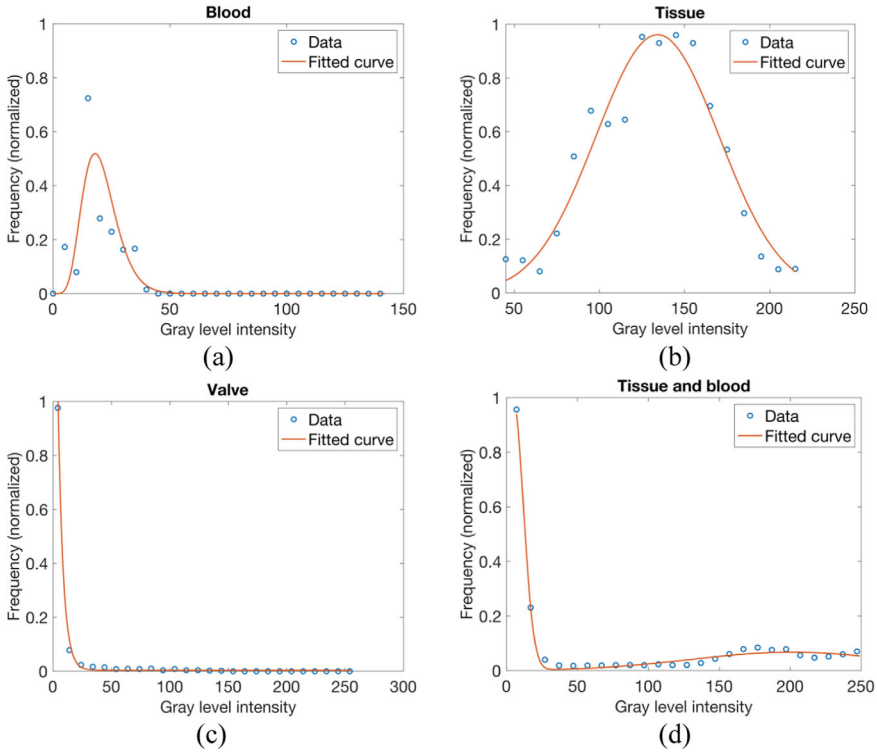


Fig. 5 Fitted curve obtained using GAs to find the optimal fit, (**a**) blood (**b**) tissue (**c**) valve (**d**) tissue and blood

Table 1 GA fitted results for five representative samples from tissue region

Tissue dataset	Generation	Function count	R^2	k1	k2	k3
T1	559	28,000	0.92	0.89	127	37
T2	737	36,900	0.96	0.96	134	35
T3	623	31,200	0.94	0.93	121	31
T4	892	44,650	0.87	0.77	152	46
T5	508	25,450	0.92	0.9	94	35

The results show that the proposed approach can accurately fit the histograms of the image segments using the appropriate distribution functions. The high R^2 values indicate that the fitted curves closely match the histogram data, thus validating the effectiveness of the proposed method. Overall, the results demonstrate the potential of using GA for curve fitting of histograms in medical image analysis.

Table 2 Summary of GA curve-fit results across different regions

	Gamma	Gaussian	Exponential	Gaussian bimodal
	(μ, σ)	(μ, σ)	(μ, σ)	(μ, σ)
Generation count	685, 332	750, 210	1801, 973	1938, 765
Function evaluations	46,400, 4300	37,545, 10,500	90,115, 5200	387,850, 15,000
R^2	0.87, 0.12	0.89, 0.06	0.9, 0.1	0.94, 0.05
k1	5.3, 9.8	0.83, 0.16	3.8, 5.6	18, 19
k2	8, 5.7	117, 35	0.16, 0.12	47, 100
k3	15, 9	43, 25	0.3, 1	95, 153
k4	–	–	–	16, 12
k5	–	–	–	19, 21
k6	–	–	–	40, 37

7 Discussion

7.1 Blood, Tissue, and Valve Samples

Studying GA curve fitting for 100 ultrasound blood image segments, the assumed Gamma distribution exhibited varying goodness of fit (R^2: 0.71–0.99). Higher R^2 values signify better fit. Shape parameter ranged widely (2.2–16.9), while scale parameter ranged narrowly (0.97–11.4). Although Gamma distribution reasonably fits, diverse shape parameter values hint at histogram variations. These GA results serve as a valuable foundation for exploring ultrasound image segment characteristics.

Table 3 (a) compares goodness of fit for Gamma, Gaussian, and exponential PDFs on 100 blood region segments. Bimodal was excluded as it would not be a proper description of data, and it was overfitting. R^2 values for Gamma (0.7291) and Gaussian (0.7029) suggest good fits, while exponential (0.163) does not. MAE for Gamma and Gaussian implies similar predictive accuracy. A lower AIC and BIC value indicates a better fit. AIC values indicate similar fits for Gamma and Gaussian, while exponential is less suitable. BIC values confirm Gamma and Gaussian as similar, with exponential less favorable. Overall it is evident that Gamma would be better fit than Gaussian here.

Table 3 (b) compares curve-fitting results for the tissue portion of cardiac echo images using three PDFs (Gamma, Gaussian, and exponential). The Gaussian distribution shows the highest R^2 (0.93) and lowest mean absolute error (0.053). AIC and BIC values are significantly lower for Gaussian compared to Gamma, suggesting its superior fit and suitability for describing the tissue distribution in cardiac echo images.

Table 3 Comparison of fits for blood, tissue, and valve regions

	Gamma	Gaussian	Exponential
(a) Blood			
R^2	0.7291	0.7029	0.163
MAE	0.0989	0.11	0.2393
AIC	−27.44	−27.17	−10.51
BIC	−27.24	−27.13	−10.68
(b) Tissue			
R^2	0.9007	0.93	0
MAE	0.0803	0.053	0.2462
AIC	−27.53	−836.8	−375.6
BIC	−27.69	−827.8	−366.5
(c) Valve			
R^2	0.8794	0.8342	0.9792
MAE	0.0237	0.0293	0.0142
AIC	−1456.44	−1374.46	−1824.83
BIC	−1445.81	−1363.82	−1814.32

For the valve regions, the R^2 values ranged from 0.4 to 0.99 for exponential fit. The mean R^2 value, as per Table 2, is 0.9, suggesting significant proportion of well-fitted curves.

The exponential PDF exhibits the highest R^2 value (Table 3(c)). It also has the smallest MAE (0.0142), indicating minimal fitting error. With the lowest AIC and BIC values, the exponential PDF demonstrates the best fit to the valve sample data.

7.2 Tissue and Blood

The R^2 values for bimodal fit ranged from 0.81 to 0.99, indicating very good fit for most of the segments. Table 4 summarizes the curve-fitting analysis for "tissue with blood" samples. The Gaussian bimodal distribution stands out with the highest R^2 value (0.9877), indicating a robust fit to the data, and the lowest MAE (0.011) compared to other distributions. Additionally, it achieves the lowest values for both AIC and BIC, underscoring its superior performance for the "tissue with blood" samples.

Table 4 Comparison of fits for tissue with blood region

	Gamma	Gaussian	Exponential	Gaussian bimodal
R^2	−0.1355	0.9761	0	0.9877
MAE	0.0978	0.0274	0.1299	0.011
AIC	−762.02	−1745.90	−790.56	−1909.89
BIC	−758.48	−1735.27	−779.93	−1888.62

8 Conclusion and Further Work

This study delves into ultrasound image analysis using statistical modeling and GAs for distinct cardiac regions. Tailored statistical models, employing various PDFs to capture diverse speckle statistics, were evaluated. Rigorous model evaluation, employing metrics like R^2, MAE, AIC, and BIC, provides quantitative insights into the models' adaptability to specific regions' statistical characteristics.

The incorporation of GAs enhances the adaptability and refinement of proposed PDFs, leading to a globally optimized fit. Convergence data, encompassing generation number, cumulative fitness evaluations, R^2 values, and curve parameters, offer valuable insights into optimization outcomes. The effectiveness of chosen distributions in modeling ultrasound image segment histograms, as indicated by R^2 and MAE values, highlights the success of this approach.

Comparative analysis of fits across different models provides insights into their relative performance, setting the stage for developing more specialized PDF models and advancing imaging techniques aiding segmentation and classification tasks. This research contributes to automated ROI detection algorithms, aiming to extract valuable information from echo images for improved diagnostic capabilities and laying the foundation for future advancements in ultrasound image analysis.

References

1. Wang S, Celebi ME, Zhang Y-D, Yu X, Lu S, Yao X, Zhou Q, Miguel M-G, Tian Y, Gorriz JM, Tyukin I (2021) Advances in data preprocessing for biomedical data fusion: an overview of the methods, challenges, and prospects. Inform Fus 76:376–421. https://doi.org/10.1016/j.inffus.2021.07.001
2. Panayides AS, Amini A, Filipovic ND, Sharma A, Tsaftaris SA, Young A, Foran D, Do N, Golemati S, Kurc T, Huang K, Nikita KS, Veasey BP, Zervakis M, Saltz JH, Pattichis CS (2020) AI in medical imaging informatics: current challenges and future directions. IEEE J Biomed Health Inform 24:1837–1857. https://doi.org/10.1109/JBHI.2020.2991043
3. Painchaud N, Duchateau N, Bernard O, Jodoin P-M (2022) Echocardiography segmentation with enforced temporal consistency. IEEE Trans Med Imag 41:2867–2878. https://doi.org/10.1109/TMI.2022.3173669
4. Pallavi K, Deepa M (2022) Adaptive thresholding method for speckle reduction of echocardiographic images. https://doi.org/10.1080/03772063.2019.1634494

5. Zhong T, Tagare HD, Beaty JD (2006) Evaluation of four probability distribution models for speckle in clinical cardiac ultrasound images. IEEE Trans Med Imag 25:1483–1491. https://doi.org/10.1109/TMI.2006.881376
6. Leclerc S, Smistad E, Pedrosa J, Ostvik A, Cervenansky F, Espinosa F, Espeland T, Berg EAR, Jodoin P-M, Grenier T, Lartizien C, Dhooge J, Lovstakken L, Bernard O (2019) Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. IEEE Trans Med Imag 38:2198–2210. https://doi.org/10.1109/TMI.2019.2900516
7. Zamzmi G, Hsu L-Y, Li W, Sachdev V, Antani S (2021) Harnessing machine intelligence in automatic echocardiogram analysis: current status, limitations, and future directions. IEEE Rev Biomed Eng 14:181–203. https://doi.org/10.1109/RBME.2020.2988295
8. Noble JA, Boukerroui D (2006) Ultrasound image segmentation: a survey. IEEE Trans Med Imag 25:987–1010. https://doi.org/10.1109/TMI.2006.877092
9. Ouzir N, Ollila E, Vorobyov SA (2020) Data-adaptive similarity measures for b-mode ultrasound images using robust noise models. IEEE J Sel Top Sign Process 14:1244–1254. https://doi.org/10.1109/JSTSP.2020.3001829
10. Hirata S, Isshiki A, Tai D-I, Tsui P-H, Yoshida K, Yamaguchi T (2023) Convolutional neural network classification of ultrasound images by liver fibrosis stages based on echo-envelope statistics. Front Phys 11:1164622. https://doi.org/10.3389/fphy.2023.1164622
11. Channoufi I, Najar F, Bourouis S, Azam M, Halibas AS, Alroobaea R, Al-Badi A (2020) Flexible statistical learning model for unsupervised image modeling and segmentation. In: Bouguila N, Fan W (eds) Mixture models and applications. Springer International Publishing, Cham, pp 325–348. https://doi.org/10.1007/978-3-030-23876-6_14
12. Montgomery DC, Runger GC (2010) Applied statistics and probability for engineers. John wiley & sons
13. Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction. Springer

Investigate the Properties of FTA Through Empirical Experimentation

Harsh Raj and Dilshad Shaik

Abstract Recently, the rapid development of artificial intelligence has led to the introduction of an array of different technologies. One such technology is fuzzy tilling activation. ReLU activation has been shown to work better for deep learning neural networks than other types of activation functions, but fuzzy tilling is a new type of activation function that has not been tried before. This work investigates the properties of FTA. Our work found that the use of tanh normalization is not dependent on bounds and can enhance the performance of FTA without any need for further tuning in the given environment. As a future direction, exploring the robustness of tanh-normalized DQN-FTA across multiple environments would be an intriguing prospect.

1 Introduction

Neural networks (NN) have been shown to perform well on several challenging problems [1–4]. Their efficacy stems from a knack for learning valid summaries of the input information through which appropriate representations are constructed. A representation of input is defined as the values of all the connection points in an artificial neural network, barring those present in the final layer of the NN when the input is passed into the NN [5]. The values of all the neurons are commonly called *features*. Unfortunately, when the NN is trained online (data received sequentially), interference can occur between the representations that are learned [6–8]. Interference is loosely defined as when a newly learned representation affects the performance of the NN on previously seen data in a negative way (i.e., the old representation becomes worse due to the newly learned one).A potentially useful intuition for when interference can occur is when the representation for some input x_1 is dense (a large number

H. Raj (✉)
SRM University, Chennai, India
e-mail: raj.harsh1906@gmail.com

D. Shaik
IIM Sambalpur, Hyderabad, India
e-mail: sheikdilshad1@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_10

of neurons are non-zero),since then, if another input x_2 is sufficiently different (uncorrelated) from x_1, its representation should be quite different. However, for a different representation to be learned for x_2, the NN must update the weights that contribute to the representation of x_1, thus making the representation for $\times$ 1 worse to learn a better representation for x_2.

A method that has been found to reduce interference is to enforce sparse representations (a large number of neurons are approximately zero) [9–12]. We revisit the example above with inputs x_1 and x_2. If the representation for x_1 is sparse, then the representation that is learned for x_2 can use weights that are unused for the representation of x_1, thus causing no interference between the representations for x_1 and x_2. The best way to enforce sparsity is an active topic of research. A detailed discussion of related work can be found in Sect. 1.1.

One way to enforce sparsity is by design, a method which does this is using a *fuzzy tiling activation* (FTA) [13]. FTA is an activation function that can be used at each layer of a NN, similar to ReLU or tanh. Different from ReLU and tanh is that FTA has a vector output instead of a scalar output. Scalar quantities are the input, whereas a one-hot vector becomes the corresponding output. The output of FTA can be thought of as mapping a scalar $z \in \mathbb{R}$ to one of k evenly spaced bins in $\mathbb{R}^k$. A critical step to using FTA effectively is setting the tiling bound parameter u. This is because z should be in $[-u, u]$ to work well. However, just satisfying this is usually not enough for good performance (as one could easily set u to be very large). For instance, if $u = 10$ and $k = 4$, then the bins would be $[-10, -5)$, $[-5, 0)$, $[0, 5)$, $[5, 10]$, and if z really takes values in $[0, 1]$, then it will always map to the same bin (bin 3 since $[0, 1] \subset [0, 5)$). Thus, as discussed by [13], setting the tiling bound is important; however, no good solution exists for setting it other than performing a search. The sensitivity to the tiling bound was especially prominent in the work of [13] when using FTA with a deep Q-network (DQN) [3] in the LunarLander RL environment.

As such, in this work we aim to do two things:

1. We aim to reproduce the FTA vs ReLU using DQN in LunarLander experiments presented in [13], to confirm that indeed FTA is sensitive to the tiling bound in this setting.
2. Next, we aim to test if the simple technique of normalizing the values (using tanh, batch norm, or range norm) passed to FTA can remove the need for tuning the tiling bound parameter in FTA.

Experiments related to the first point are shown in Sect. 3.1, indicating that the experiments were reproducible and nearly identical results were obtained as in [13].

To answer the question in the second point, we normalize the values passed into FTA using three methods: passing them into tanh, batch norm, or range norm (all to be defined in Sect. 3.2). The method of using tanh to normalize the input values was discussed in [13], but not tested. Pan et al. [13] mentioned that using this method might suffer from vanishing gradients. However, we did not see any indications of this in our experiments and obtained very positive results for tanh. We hypothesize that normalizing the values should solve the need to set tiling bounds using search,

since normalizing will ensure that the values passed into FTA are in [−1, 1], and thus, the tiling bound can always be set to [−1, 1]. We found that tanh was most effective, while batch norm and range norm did not perform as well. Experiments highlighting these results are shown in Sect. 3.2.

1.1 Related Work

Representation learning can influence learning efficiency, positively through generalization but negatively through interference [9, 14–16]. Learned representations using neural networks are especially vulnerable to interference due to aggressive generalization, as updates to one feature's estimates change the estimates of later features in unexpected ways, degrading model accuracy. Interference in features learned by neural networks is even worse when trained on temporally correlated data [17–19].

Several authors [9, 10, 20, 21] demonstrated how scattered representations tend to reduce the impact of interference for learning operations: few features stay active for each input, and thus each update is less likely to interfere with the weights of other features. Sparse feature spaces can leave an interpretable impact as outputs can be traced to the activation of the most input-discriminative features. However, learning such sparse representations in the online setting remains an open problem.

Existing strategies to ensure sparsity are to pre-train representations offline using regularizers [9] or meta-learning [10], or online methods using regularizers with replay buffers [21]. While sparsity regularizers do improve learning, they often lead to high levels of dead neurons that degrade the learned representation over time [21]. Kernel representations can also provide sparse representations online, but do not scale well to large problems due to their computational complexity [13]. Therefore, there is motivation to find a simpler method to ensure sparsity in learned representations that is easy to implement, train, and scale to larger input spaces.

FTA is an approach to achieve sparsity by design, where mechanisms to activate can be deployed to the various layers found in a selected neural network. The mechanisms provide adjustable sparsity without the need for pre-training or adjustments to the loss function. The FTA function aggregates inputs of any feature layer into bins with differentiable, overlapping zones. Much like tile coding [22], FTA allows learning to be discriminative between different input values through binning, but unlike tile coding, FTA maintains input generalization through learning activations within the differentiable regions between bins. When applied to DQN and deep deterministic policy gradient, or DDPG for short, [23], FTA repeatedly shows a likelihood of improving agent performance for discrete and continuous control environments, often outperforming activations like ReLU and tanh [13]. Furthermore, FTA-applied networks can often perform better without target networks, circumventing the sample inefficiency problem inherent in their use [13]. A recent study by [24] on transferring learned representations to networks performing auxiliary tasks found that FTA-based representations transferred better and more consistently than ReLU-based representations for less similar tasks.

2 Preliminaries

Upon introducing FTA and the rectified linear unit (ReLU) activation. Moreover, we formally define the RL setting and the deep Q-learning (DQN) algorithm which we use for our experiments.

2.1 Fuzzy Tiling Activation

First, we introduce the simpler *tiling activation* (TA) and then extend it to the FTA. A TA is a function $\phi : \mathbb{R} \to \mathbb{R}^k$, which expects inputs $z \in [-u, u]$ and maps them to one-hot vectors (standard basis) in $\mathbb{R}^k$. . The $[-u, u]$ is referred to as the tiling bound. For example, if $u = 10$ and $k = 4$, then any $z \in [-10, -5)$ would correspond to $(1, 0, 0, 0) \in \mathbb{R}^k$, any $z \in [-5, 0)$ would be linked to $(0, 1, 0, 0)$, any $z \in [0, 5)$ would be mapped to $(0, 0, 1, 0)$, and any $z \in [5, 10]$ would be mapped to $(0, 0, 0, 1)$. [13] show that the TA can be implemented efficiently as follows. Assume you want evenly spaced bins of size $\delta > 0$ and $k = 2u/\delta$, where WLOG u was chosen such that it is divisible by δ. Define the tiling vector

$$\mathbf{c} = (-u, -u + \delta, I + 2\delta/\ldots, u - 2\delta, u - \delta) \in \mathbb{R}^k \tag{1}$$

Then the TA can be defined as

$$\mathbf{c} = \phi(z) =_1 -I_+(\max(\mathbf{c} - z_1, 0) + \max((z - \delta)_1 - \mathbf{c}, 0)) \tag{2}$$

where $1 \in \mathbb{R}^k$, and $I_+()$ indicate returns equal to 1 for positive inputs and zero in other cases, and applied element-wise to vectors.

An issue with the TA is that it has zero derivatives almost everywhere. To solve this issue, [13] proposed modifying the TA for a fuzzy tiling activation (FTA). The FTA can be implemented as follows. Define a fuzzy version of the indicator function as

$$\mathbf{c} = I_\eta = I_+(\eta - x)x + I_+(X - \eta) \tag{3}$$

where $\eta \geq 0$. Then the FTA can be defined as

$$\mathbf{c} = \phi_n(z) = 1 - I_n + (\max(c - z1) + \max((z - \delta)_1 - c, 0)) \tag{4}$$

where $I_{\eta,+}$ is applied element-wise to vectors. The reason for the use of fuzzy in the name of FTA can be understood from the definition of $I_{\eta,+}$. The term $I_+(\eta - x)$ is 1 when $x < \eta$, while when $x > \eta$, $I_+(x - \eta)$ is 1. Thus, when $x < \eta$ we have that $I_{\eta,+}$ evaluates to x, giving a smoother (fuzzy) indicator function. Notice that this means

for $x < \eta$ the derivative is non-zero, allowing us to use backpropagation to train out networks. Also, when $\eta = 0$ the original indicator function I_+ can be recovered.

2.2 ReLU Activation

We will be comparing FTA to the ReLU activation later. Thus, we formally introduce it now. ReLU is defined as the positive part of its argument, formally:

$$\mathbf{c} = \phi(z) = \max(0, z) \tag{5}$$

2.3 Reinforcement Learning

Markov decision process (MDP) accurately represents a reinforcement learning situation. The tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is that state space, $\mathcal{A}$ is the action space, $P\mathbb{R}$ is the transition probability kernel, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{R} \rightarrow \mathbb{R}$ represents the reward, and $\gamma \in [0, 1]$ is the reduction. A policy $\pi : \mathcal{S} \rightarrow \mathcal{M}_1$ is a map from states to distributions over actions, where $\mathcal{M}_1$ is the set of all probability measures over actions $\mathcal{A}$. . Then, an agent (policy) interacts with an MDP as follows. At each instance $t \in 1, 2\ldots$, a state is captured as defined by $s_t \in \mathcal{S}$ The agent responds per $a_t \sim \pi(s_t)$ before moving to a position described by $s_{t+1} \sim P\mathbb{R}$ (s_t, a_t), for which the reward runs along $R(s_t, a_t, s_{t+1})$.

Given below is an action-value function with π policy.

$$\mathbf{c} = Q_\pi(s, a) := \mathbf{E}\big[G_t | S_t = s, A_t = a; A_{t+1:\infty} \sim \pi\big] \tag{6}$$

where $G_t := \sum_{t=0}^{\infty} \nu^t R(s_t, a_t, s_{t+1})$ is returned. We expect the agent to increase its reward for each state in $\mathcal{S}$.

2.4 Deep Q-Learning

When using deep Q-learning (DQN) [3] in the experiments we performed, as explained in Sect. 2.3, the agent aims to interact with the ambient settings to enhance expected rewards. The ideal and most efficient action-value function is shown as:

$$\mathbf{c} = Q^*(s, a) = \max_\pi Q_\pi(s, a) \tag{7}$$

In Q-learning, the action-value function is iteratively updated by using the following (Bellman) equation to approach an action-value function $Q_{t+1}(s, a) = \mathbf{E}\big[R(s_t, a_t, s_{t+1}) + \gamma \max_{at+1} \in AQ_t(s_{t+1}, a_{t+1})|A_t = a, S_{t+a} = s_{t+1}\big]$. This optimal function is $Q_t \to Q^*, t \to \infty$. A NN approximates with weights θ and approaches the action-value function, $Q(s, a; \theta)$. The agent's experience $(s_{t-1}, a_{t-1}, s_t, R(s_{t-1}, a_{t-1}, s_t))$ at each time step t is stored into a *replay memory*. A small sample $\mathcal{B}_t$ is selected randomly from the replay memory to calculate the loss function.

The NN training includes minimization of the given loss function at every interval t and then performing a gradient descent update

$$c = L_t(\theta_t) = \sum_{s,a,r,s' \in B_t} \left[r + \gamma \max_{a'} Q\big(s', a'; \theta_{t-1}\big) - Q(s, a; \theta_t) \right]^2 \tag{8}$$

The agent's behavior policy is an ε-greedy mechanism, which randomly selects per probability ε and is characterized by greediness with respect to $Q(s_t,; \theta t)$ otherwise.

3 Experiments

The LunarLander environment, from OpenAI Gym [25] is a trajectory rocket optimization problem, where the goal is to land a moon lander on a landing pad.

The state space of the environment $\mathcal{S}$ is comprised of 8 attributes, namely x and y coordinates, v_x and v_y velocities, θ and v_θ angular velocities, and 2 Booleans to show if either leg is touching the ground. At each timestep t the action space

$$\begin{aligned} \mathcal{A}(s_t) = \{&\text{do nothing} \\ &\text{fire left engine,} \\ &\text{fire main engine,} \\ &\text{fire right engine}\}. \end{aligned}$$

The reward for making it onto the landing pad earns 100 to 140 points, with an additional 100 points if it stops on it, and hence, a solved episode is + 200 points. Firing the main engine has a −0.3 and the side engines − 0.03 points penalty, and the lander gets −100 points if it crashes. If the agent crashes, leaves the viewport, or remains stationary without touching another body, the episode will terminate.

We use a student's t-distribution to display 95% confidence intervals for all of the plots in this section. The confidence intervals appear as the shaded regions in the plots. The validity of using such confidence intervals is investigated in Sect. 3.3.

3.1 Reproducibility Experiment

Following the same LunarLander experiment settings as in [13], we use a two-layer neural network along a different activation function in the ultimate hidden layer as a primary difference. For DQN and DQN-Large, a ReLU activation mechanism applies to each layer save the last one, which is linear. Since DQN with FTA activation function expands the number of features, its last hidden layer is $k = (l - \mathrm{u})/\delta$ times bigger than the DQN, and hence, we use DQN-Large on the final layer that is of exactly the same size of DQN-FTA with and without the target network.

For DQN-FTA in the last layer, we set $[l, u] = [-20, 20]$ and $\eta = \delta = 2.0$, and hence, $k = 40/2 = 20$.

In Fig. 1, we observe that, while DQN-FTA with and without the target network performs slightly better than DQN, DQN-Large outperforms both DQN and DQN-FTA. This is the same behavior that was observed by [13] for this experiment, and thus, we were able to reproduce the experiment.

To show the sensitivity of DQN-FTA to the tiling bounds, we repeat the experiment for DQN-FTA only, with different values of $u \in \{0.1, 1, 10, 50, 100\}$, $l = -u$ with different number of tiles $\mathrm{k} \in \{16, 24, 128\}$ for each tiling bounds.

Figure 2 depicts the results of the experiment. The DQN-FTA in LunarLander achieves its best when we set u to a reasonably small value (approximately $u = 1$). However, it performs poorly when it is set too large ($u = 10, 50, 100$) or small ($u = 0.01$). With $\mathrm{u} \in \{10, 50, 100\}$, performance improves by increasing the number of tiles k, but it gets worse when u is very small. The improved performance when

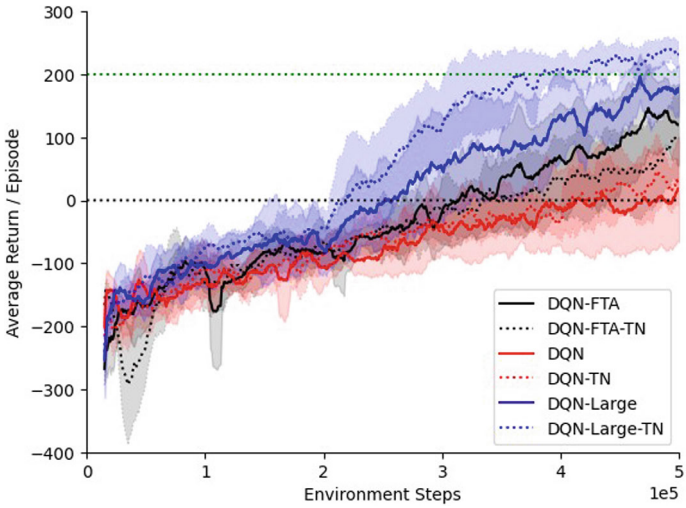


Fig. 1 Representations of **DQN-FTA(black)**, **DQN(red)**, and **DQN-Large(blue)** in LunarLander. The **sprinkled** line shows the relevant network's algorithmic training. The results were taken as an average of 10 trials

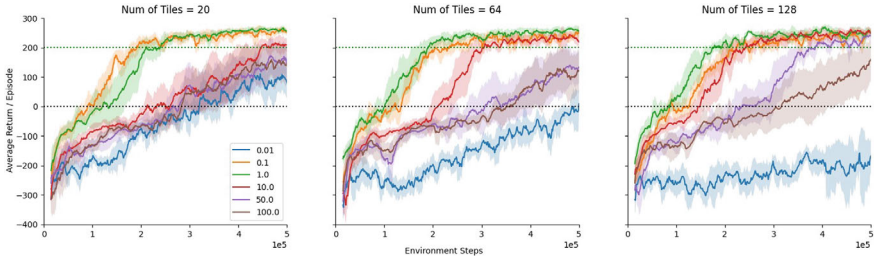


Fig. 2 Evaluation learning curves of DQN-FTA using different values of tiling bounds: $u \in \{0.1, 1, 10, 50, 100\}$, $l = -u$, with different number of tiles $k \in \{16, 64, 128\}$. Shown is the average result obtained from 10 trials

the number of tiles k is increased can be understood by looking at the case when $u = 10$ and $k = 128$. Notice that $u = 10$ is 10 times larger than $u = 1$, and $k = 128$ is approximately 10 times larger than k = 16. Thus, for the range [1] there will be approximately the same number of bins when $u = 10, k = 128$ as when $u = 1, k = 16$. This explanation explains why the green line ($u = 1$) when $k = 16$ is approximately the same as the red line ($u = 10$) when $k = 128$.

We found FTA sensitive to the tiling bound (u), and hence, only when choosing the right value of u we observed that it outperforms both DQN and DQN-Large. We found that the best value $u = 1$ for DQN-FTA without a target network was $u = 1$. Figure 3 shows a plot of the above-described DQN-FTA network compared to the untuned DQN-FTA ($u = 20$) and DQN using ReLU activation instead of FTA. This clearly indicates that when the value of u is set appropriately, DQN-FTA can outperform DQN with ReLU activation.

3.2 *Normalizing Experiment*

In our experiments, detailed in Sect. 3.1, it can be observed that the performance of FTA is quite sensitive to the tiling bound $[-u, u]$. This reinforces the observation made in [13]. In Fig. 2, we can see that the best performance on LunarLander was achieved with $u = 1$, and it got worse as we increased or decreased u. It can be inferred intuitively that for a small value of u, many inputs to FTA may be out of its range providing zero gradients. Besides, for bigger values of u, multiple points of input can nurture similar tiles. In both scenarios, the results are numerous inactive neurons and enhanced interference. This means there is a specific value of u which would work best for any specific environment. Since it is not straightforward to observe the range of outputs from a neural network layer, u cannot be set mathematically. It has to be tuned manually by sweeping over a range of values in multiple experiments, which increases the cost of a project.

We hypothesize that if the inputs to FTA were scaled to a certain range, the performance would be less sensitive to the tiling bounds. We try three different

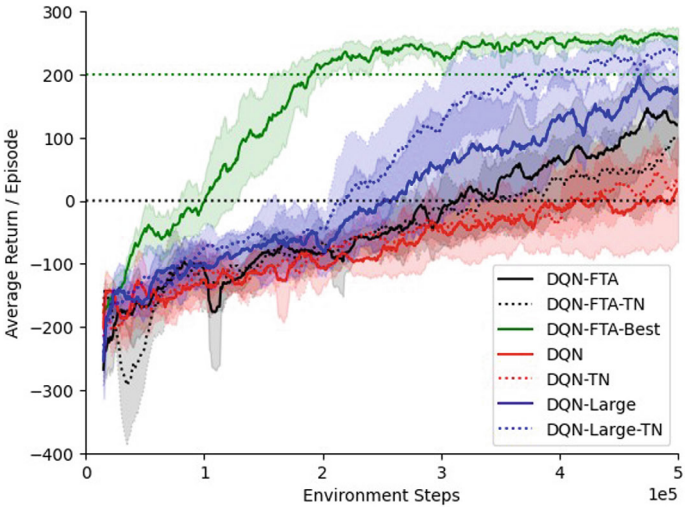


Fig. 3 Evaluation learning curves of DQN-FTA(black) ($u = 20$, $k = 40$), **DQN(red)**, **DQN-Large(blue)**, and **DQN-FTA(green)** ($u = 1$, $k = 64$). The dotted line shows the algorithms trained with target network

methodologies to achieve this. First, using a batch normalization [26] layer before FTA. Second, using a tanh activation layer before FTA. And finally, using a range normalization layer before FTA.

Batch normalization scales its input x to a learned mean β and variance γ. It is defined as

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \in}} * \gamma + \beta \tag{9}$$

By using a batch norm layer before FTA, we expect it to learn the best γ and β for any particular tiling bound of FTA.

On the other hand, tanh is an activation function that maps the input to a continuous range of values between -1 and 1. For bigger inputs, the output tends closer to 1, and in cases of smaller input instances, the output points gravitate to -1. This range of outputs can be controlled by multiplying them through non-vector functions as demonstrated below.

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

The motivation behind using tanh is to control the range of inputs to FTA so that a standard tiling bound of $u = -l = 1$ would work in every case, regardless of the agent or the environment.

Range normalization is used to scale the values to fall within a specific range. We use range normalization to scale the inputs to FTA, to the range of -1 and 1. It is

defined as

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot 2 - 1 \tag{11}$$

Similar to tanh, the reason for using range normalization is to ensure the range of inputs to FTA is between -1 and 1.

To test this hypothesis, we run experiments with tanh, batch norm, and range norm as the layer preceding FTA on LunarLander. One, with the best performing FTA tiling bound ($u = 1$) and two, with the worst-performing FTA tiling bounds ($u = 0.01, 100$). All other parameters and configurations are kept the same as those in Sect. 3.1.

Figure 4 depicts the results of these experiments in LunarLander, compared to DQN-FTA with the best FTA tiling bounds and the two worst-performing tiling bounds. When tanh is used as the preceding layer to FTA to normalize its inputs, we observe good performance with all the tested tiling bounds and a significant improvement over DQN-FTA with the worst tiling bounds ($u = 0.01, 100$). Range normalization performs poorly across all the tiling bounds. While batch normalization improves the performance of DQN-FTA with the highest tiling bound, it performs very poorly when the tiling bound is very small. Overall, it is clear that *tanh* normalization is effective regardless of the tiling bound in the LunarLander environment.

Cartpole To test the effectiveness of the tanh normalizing technique across environments, we also performed experiments on the CartPole environment from OpenAI Gym [25]. The CartPole environment starts as a pole balanced atop a cart, with the objective being to maintain the pole's stability even when the cart shifts to the left or to the right. The state space of the CartPole environment contains four values: cart stance, cart velocity, pole angle, and pole angular velocity. The directions involve moving the cart to the left and then moving it to the right. With the objective being to ensure the pole does not fall, a $+1$ reward ensues for each step, and the episode terminates when the pole is at an angle greater than $\pm 12\circ$.

We first tested to see how DQN-FTA without a target network performs with three different tiling bounds ($u = 0.01, 1, 100$) and $k = 20$. The results can be seen on the

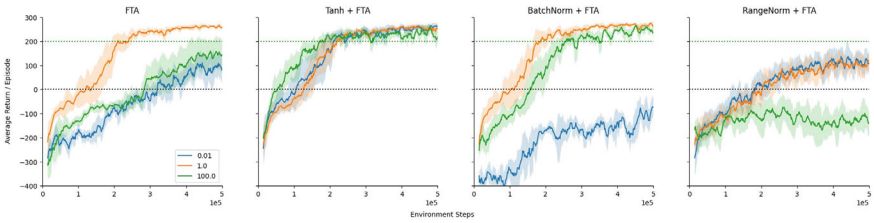


Fig. 4 Note the curves plotting periodic results against environment intervals for Batch Norm, Range Norm and tanh as layers preceding FTA, compared to DQN-FTA without a normalization layer preceding FTA

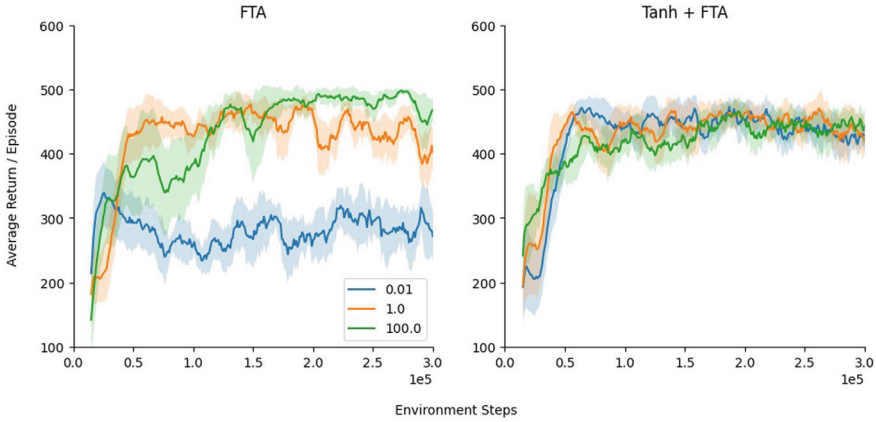


Fig. 5 Each agent learning curve is generated using a u and all are trained without target networks

left plot in Fig. 5. Unlike in LunarLander, a tiling bound of $u = 100$ performs best, while $u = 1$ performs worse. Importantly, this highlights the sensitivity of FTA to the tiling bound parameter in the CartPole environment as well as LunarLander.

We then applied tanh normalization before the FTA layer and found this improved the performance of $u = 0.01$, 1 to be similar to that of $u = 100$. The right plot in Fig. 5 illustrates that DQN-FTA with tanh normalization at all tiling bounds perform similarly well to the best tiling bounds for DQN-FTA without tanh normalization, for all the tiling bounds we tested ($u = 0.01$, 1, 100). Thus, the experimental results in the CartPole environment further confirm that using tanh normalization before an FTA layer can potentially remove the need for tuning the tiling bound u.

3.3 Distribution of Returns

We observed that DQN-FTA had noticeably different returns at 500k steps between seeds, wherein some seeds performed significantly better than others. To better understand agent performance with respect to initialization stochasticity, we investigate the distribution of mean episodic returns for ReLU, FTA, and FTA-tanh over various time steps. Additionally, student's t confidence intervals and averaging over runs in evaluation learning curves assume that mean episodic returns are uni-modal and normally distributed; violating these assumptions would misrepresent the robustness of the agent and the average return to which it converges. A bi-modal distribution of average episodic returns at the final stage of learning (i.e., 500k steps) would indicate that an agent is sensitive to weight initialization or environmental stochasticity and therefore demonstrate inconsistent learning and unreliable behavior.

Figure 6 shows the distribution of mean episodic returns for DQN, (untuned) DQN-FTA, and DQN-FTA-tanh. DQN and DQN-FTA has the same parameters as in

3.1, and DQN-FTA-tanh has the same parameters as in 3.2. The FTA agents have the same number of bins, $k = 20$, with DQN-FTA-tanh differing from DQN-FTA by a tanh activation layer before the FTA layer. We can observe some general distribution characteristics and trends; for DQN and DQN-FTA, distributions of returns at the beginning of learning appear uni-modal and shift away from the mean of their distributions near the end of learning, while the distribution of returns for DQN-FTA-tanh remain uni-modal and appear to converge toward the mean of its distribution. At 495k time steps, DQN and DQN-FTA distributions appear bi-modal while the DQN-FTA-tanh distribution appears strongly uni-modal and more normally distributed. While 50-run sample sizes per agent is relatively small hypothesis testing, our preliminary results in LunarLander show that (1) DQN and DQN-FTA distributions of returns may violate the normality and uni-modality assumptions near the end of training, an important observation on performance stability under stochasticity in the environment and weight initializations, and (2) DQN-FTA-tanh appears to maintain those assumptions throughout learning, illustrating that tanh normalization can improve FTA stability.

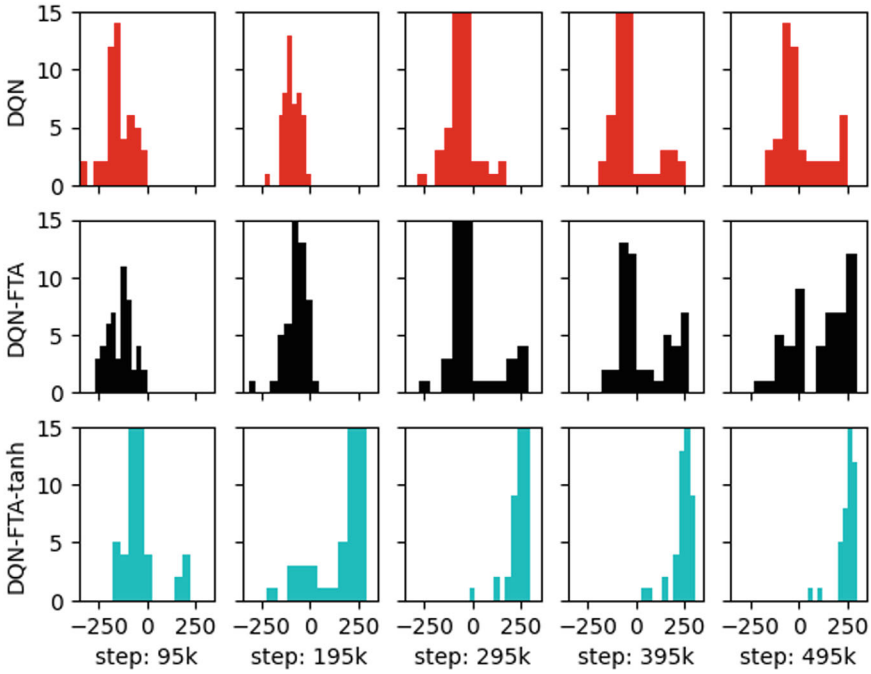


Fig. 6 Histogram of mean episodic returns for **DQN(red)**, **DQN-FTA(black)**, and **DQN-FTA-tanh(cyan)** over time step learning progression in LunarLander. There are 50 runs per agent, with the mean episodic return values at each time step divided into 10 bins between the range [−350, 350]. All three agents do not use target networks

4 Discussion

After exploring FTA through empirical experimentation as an extension to the original FTA paper by [13], we first reproduced the learning curve evaluation for DQN-FTA, DQN, and DQN-Large in LunarLander and found similar trends. When comparing the untuned FTA agent ($u = 20$) against the tuned ReLU agents, we found that FTA and DQN-Large reached higher average returns by 5×10^5 time steps than [13]. It is currently unknown why our FTA and DQN-Large implementations perform better than the original paper. There are many potential reasons: stochasticity in our runs and initialization, our specific seeds used to evaluate performance, or even the implementation framework (we used PyTorch while [13] used TensorFlow) could all influence the empirical outcome of experiments. While we only average over 10 runs compared to the 20 performed in the original paper, this should not have contributed to the performance discrepancy given our relatively tight confidence intervals between runs.

Further investigation into the plausible causes may be useful; however, we believe that since the same trends between networks were observed it is sufficient to claim we were able to reproduce the LunarLander experiments from [13].

Our second experiment reproduces the bound sensitivity experiment comparing FTA with different tiling bounds and across different numbers of tiles. Our results share the conclusion that moderately small values for u ($u \in 0.1, 1$) perform better than the extremes ($u \in 0.01, 10, 50, 100$) [13]. We also found that increasing the number of tiles improves the performance of larger tiling bounds ($u \in 50, 100$) while an extremely small u gets worse ($u = 0.01$). This is expected, as increasing the number of tiles given a large tiling bound reduces δ, thereby reducing interference between bins. In addition, increasing the number of tiles given an extremely small tiling bound greatly increases delta and results in high interference between bins.

While [13] addressed the *out-of-boundary* and *gradient vanishing* problem when $z \notin [l, u]$, we address the over-generalization problem due to incorrect bounds wherein the majority of $z \notin [l, u]$ aggregate into a small range of the available bins. For example, if the inputs to FTA are $z = x\,w \in [-1, 1]$ while the range $[l, u] = [-20, 20]$ and $\delta = \eta = 2$ such that number of bins $k = 20$ and $c = (l, l + \delta, l + 2\delta, \ldots, u - 2\delta, u - \delta, u) = (-20, -18, -16, \ldots, 16, 18, 20)$ (like the LunarLander experiment in Sect. 3.1), then $FTA(z)$ will only bin z into two bins, between the ranges [−2,0) and [0,2). When the $[l, u]$ bounds extend beyond the standard input to activation range, the sparsity provided by FTA is limited due to the number of unused bins. To address the bound selection problem, we perform normalizing experiments using range normalization, tanh activation, and batch normalization. We found that tanh normalization is bound-agnostic and improves the performance of FTA without requiring additional tuning for the environment. In Sect. 3.2, we show that tanh-normalized FTA performs the same regardless of the tiling bound [l, u] given a fixed number of bins k. When comparing against the agents in the LunarLander experiments, tanh-normalized DQN-FTA performs as well as a tuned FTA and better than all the other agents near the end of learning. tanh-normalized DQN-FTA also performs

better than untuned FTA in Cartpole, extending the robustness tanh normalization within other environments where the input range for $z \in [l, u]$ is unknown and agents are untuned.

Lastly, we evaluate the distribution of mean episodic returns for DQN, (untuned) DQN-FTA, and DQN-FTA with tanh normalization in LunarLander. We find that near the end of learning, DQN and DQN-FTA have more bi-modal distributions, whereas DQN-FTA with tanh normalization is more uni-modal. This potentially demonstrates that tanh-normalized FTA improves agent stability to randomness in the initialization of network weights and environment settings compared to agents using ReLU or untuned FTA. Untuned DQN-FTA's bi-modally distributed returns could also potentially explain the poor performance in LunarLander reported by [13]. Although our distribution experiments are preliminary, and that hypothesis testing with significantly more sample runs is required to more conclusively address distribution normality and modality, Fig. 6 illustrates how inconsistent the returns can be for an untuned FTA agent. Furthermore, our results emphasize the sensitivity of FTA tiling bounds [l, u] and the importance of a bound-agnostic solution to FTA for its reliability and widespread adoption.

Comparing against the agents in the LunarLander experiments of Sect. 3.1, tanh-normalized DQN-FTA performs as well as a tuned FTA and better than all the other agents near the end of learning. The experiments performed in the CartPole environment indicate a similar conclusion. Evaluating the robustness of tanh-normalized DQN-FTA across more environments is left as useful future work.

5 Conclusion

To summarize, our study aimed to investigate the properties of FTA through empirical experimentation. We observed that neural networks perform well on challenging problems, but interference can be reduced by enforcing sparse representations. One method to achieve this is through fuzzy tiling activation (FTA). Our research found that using tanh normalization in FTA improved its performance without the need for additional tuning, and it was found to be bound-agnostic. In conclusion, our study suggests that FTA with tanh normalization has the potential to enhance the robustness of neural networks and may be beneficial in various environments. Further research could explore its efficacy in more diverse settings.

Our research topic focused on investigating the properties of fuzzy tiling activation (FTA) through empirical experimentation. Neural networks have shown impressive performance on various challenging problems. However, one significant challenge in neural networks is the issue of interference, which can occur when multiple inputs activate the same neurons. Enforcing sparse representations has been proposed as a solution to address this issue.

One method to enforce sparsity in neural networks is through the use of FTA, to separate input field as sets of fuzzy tiles. Each tile is associated with a neuron that is

only activated when the input falls within its corresponding fuzzy region. This helps to enforce a sparse representation of the input space and can reduce interference.

In our research, we investigated the impact of using tanh normalization in FTA to improve its performance. We found that using tanh normalization was bound-agnostic and improved the performance of FTA without requiring additional tuning for the environment. This is because tanh normalization helped to keep the activations of the neurons within a reasonable range, which improved the stability and convergence of the network during training.

References

1. Brown T, Mann B, Ryder N, Subbiah N, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. Adv Neur Inform Process Syst 33:1877–1901
2. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
3. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602
4. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A et al (2017) Mastering the game of go without human knowledge. Nature 550(7676):354–359
5. Tilley MJ, Miller M, Freedman D (2023) Artificial neuronal ensembles with learned context dependent gating. arXiv preprint arXiv:2301.07187
6. Chandak Y, Theocharous G, Kostas J, Jordan S, Thomas P (2019) Learning action representations for reinforcement learning. 941–950
7. Caselles-Dupré H, Garcia-Ortiz M, Filliat D (2018) Continual state representation learning for reinforcement learning using generative replay. arXiv preprint arXiv:1810.03880
8. Madjiheurem S, Toni L (2019) Representation learning on graphs: a reinforcement learning application. 3391–3399
9. Liu V, Kumaraswamy R, Le L, White M (2019) The utility of sparse representations for control in reinforcement learning. 33:4384–4391
10. Javed K, White M (2019) Meta-learning representations for continual learning. Adv Neural Inform Process Syst 32
11. Rafati J, Noelle DC (2019) Learning sparse representations in reinforcement learning. arXiv preprint arXiv:1909.01575
12. Gai S (2023) Theory of reduced biquaternion sparse representation and its applications. Expert Syst Appl 213:119245
13. Pan Y, Banman K, White M (2019) Fuzzy tiling activations: a simple approach to learning sparse representations online. arXiv preprint arXiv:1911.08068
14. Bullinaria JA (1994) Representation, learning, generalization and damage in neural network models of reading aloud. Submitt Psychologic Rev
15. Rebuffi SA, Kolesnikov A, Sperl G, Lampert CH (2017) Icarl: incremental classifier and representation learning, pp 2001–2010
16. Le L, Kumaraswamy R, White M (2017) Learning sparse representations in rein- forcement learning with sparse coding, pp 2067–2073
17. Liu V, White A, Yao H, White M (2020) Towards a practical measure of interference for reinforcement learning. arXiv preprint arXiv:2007.03807
18. Bengio E, Pineau J, Precup D (2020) Interference and generalization in temporal difference learning. 119:767–777

19. Zhang T, Wang X, Liang B, Yuan B (2022) Catastrophic interference in reinforce- ment learning: a solution based on context division and knowledge distillation. IEEE Trans Neural Netw Learn Syst 1–15
20. Ghiassian S, Rafiee B, Lo YL, White A (2020) Improving performance in reinforcement learning by breaking generalization in neural networks. arXiv preprint arXiv:2003.07417
21. Hernandez-Garcia JF, Sutton RS (2019) Learning sparse representations incre- mentally in deep reinforcement learning. ArXiv, vol. abs/1912.04002
22. Sherstov AA, Stone P (2005) Function approximation via tile coding: automating parameter choice, pp 194–205
23. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971
24. Wang H, Miahi E, White M, Machado MC, Abbas Z, Kumaraswamy R, Liu V, White A (2022) Investigating the properties of neural network representations in reinforcement learning
25. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W (2016) "Openai gym," arXiv preprint arXiv:1606.01540
26. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift, pp 448–456

BDSLL: Biomedical Document Recommendation Using Knowledge-Empowered Semantic Intercepted Large Language Models

A. Aravind Krishnan and Gerard Deepak

Abstract There is a requirement for document recommendation frameworks focusing on certain domains linked to medical sciences and biosciences like biomedical document recommendation in the era of the Web 3.0. This paper proposes a biomedical document recommendation strategy that realizes informative term generation using the encompassment of extraction of terms and categories and the TF-IDF. The framework also generates an ontology from the informative terms harvested from which features are selected to classify the dataset using a random forest classifier. The model also uses very strong knowledge bases that are highly specific to the domain like QIAGEN. It also uses WikiData which is generic but has a full cover of world knowledge to bridge the knowledge present in the datasets and the existing structure web. Metadata generation also increases the quantity of additional knowledge that is encompassed in the model. BioBERT is a strategic large language model that focuses on biosciences and biomedicine as its domain is used to classify the metadata to make it more permeable into the model. Semantic similarity is computed using NPMI, web overlap, SimRank, and Ricklefs and Lau index at different stages in the pipeline to facilitate quantitative reasoning through semantic relatedness computation. The salient semantic analysis also accelerates the overall semantic association into the proposed framework. This framework can demonstrate a precision of 96.56%, recall of 96.93%, and an FDR of 0.04.

Keywords Document recommendation · Biomedical document · Ontology · Large language model · Ricklefs and Lau index

A. Aravind Krishnan
Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India

G. Deepak (✉)
Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bangalore, India
e-mail: gerard.deepak.christuni@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_11

1 Introduction

In the evolving landscape of the web transformation, the emergence of the Web 3.0 which inculcates semantic intelligence into the general web structure has proven to be one of the most important developments. This evolution represents a fundamental reimagination of the web, aiming to overcome the limitations of its predecessors by prioritizing interoperability and transparency into this digital transformation.

Biomedicine is an intersection of the areas of bioscience and medicine, bringing together knowledge and methodologies from both disciplines. Bioscience is a broad study of living organisms, structure, function, evolution, and interactions with their environments. The terminologies, ontological entities, and documents that arise from these two convergent areas of science are large and require frameworks that can semantically analyse and recommend documents for the same. It is of vital importance that this framework is also in compliance with the standards of Web 3.0. Such a framework would facilitate efficient information retrieval and bridge the gap between the current era and the vision for Web 3.0.

Motivation: The absence of a document recommendation framework for highly generic but rare domains like biosciences and biomedicine is the primary motivation for proposing this framework. Ontologies and recommendation frameworks for medical sciences are much more common than frameworks for biomedicine which is an intersection of medical sciences and biosciences. Apart from the domain-specific unavailability, strategic frameworks that comply with Web 3.0's standards that are knowledge-centric are rare, and it is important to formalize knowledge-centric paradigms for document recommendation for domains that are marginalized and rare in the current context of the web structure.

Contributions: A novel hybridization of a machine learning framework and an LLM-based domain-specific model, namely, the BioBERT Is carried out to classify the dataset and the metadata, respectively. The presence of an LLM-based classifier not only helps in the convergence of optimality at a quicker rate but also facilitates domain-specific classification which has increased accuracy. The encompassment of a highly domain-specific knowledge base that is the QIAGEN along with the WikiData knowledge base to harvest entities lowers the cognitive gap between the web and the framework which is a novel technique incorporated into this framework. Further, the framework computes semantic similarity using the Ricklefs and Lau index, web overlap, SimRank, and NPMI measures. The presence of an LLM-based classifier not only helps in a convergence of optimality at a quicker rate but also facilitates domain-specific classification with an increased accuracy.

Organization: Sect. 2 depicts the related works, Sect. 3 depicts the proposed system architecture, Sect. 4 depicts the implementation, performance evaluation, and results, Sect. 5 depicts a conclusion, and in the end of the chapter depicts references.

2 Related Works

Yang et al. [1] have proposed a deep learning framework for diagnostic multi-turn Q&A records, aiming to efficiently reuse medical data and documents. Tan et al. [2] have proposed improvement to existing scientific document recommendation by proposing a self-supervised coupled text pair embedding model. Ramya et al. [3] have proposed a relevant document recommendation framework that uses cosine similarity and latent Dirichlet allocation for topic modelling. Ref. [4–7] are several frameworks used for document recommendation and document-related measurements. Ref. [8, 9] are frameworks that strategically utilize BioBERT models in different application cases. Reese et al. [10] have created a metatag harvester that utilizes MarcEdit.

3 Proposed System Architecture

Figure 1 depicts the architecture diagram for the framework for the recommendation for documents using knowledge-empowered semantic interceptive large language models. The framework has two major inputs, namely the documents dataset and the user query with biomedicine as the domain. The biomedical document dataset is subjected to the extraction of categories as the dataset is categorical to yield informative terms. Subsequently, the biomedical document dataset is subjected to the TF-IDF to yield terms that are frequent within the document corpus and rare terms as well. The frequent entities, the rare entities, and the extracted categories are all merged as a single informative term set. This informative term set is further used in the framework which serves as a primary input to generate an ontology. Ontology generation is conducted by using Onto Collab as a tool of choice without any restrictions upon the number of levels and individuals. However, only the top 2% of the ontology are enlisted as features for the random forest classifier to classify the biomedical document dataset.

The model is subjected to the generation of entities using the QIAGEN knowledge base which is a biomedical knowledge store repository that is highly specialized for biomedicine as the domain of choice and also the WikiData which is a community-contributed and community-verified knowledge store repository. This yields strong entities which are further used in the model. The generated ontology is provided as input to the metadata generator using the MarcEdit metatag harvester the tool of choice. The metadata thus generated is subjected to classification by the BioBERT LLM-based classifier.

BioBERT which is a pre-trained LLM for the biomedical domain is made use of to classify the generated metadata. The classified instances are subjected of semantic similarity calculation and overlap, namely web overlap and NPMI indices. Web overlap is set to a threshold of 0.75, and NPMI is set to a threshold of 0.65. The positive terms that emerge from this pipeline are used in the model to compute the salient semantic analysis with a step deviance of 0.15. The entities that emerge

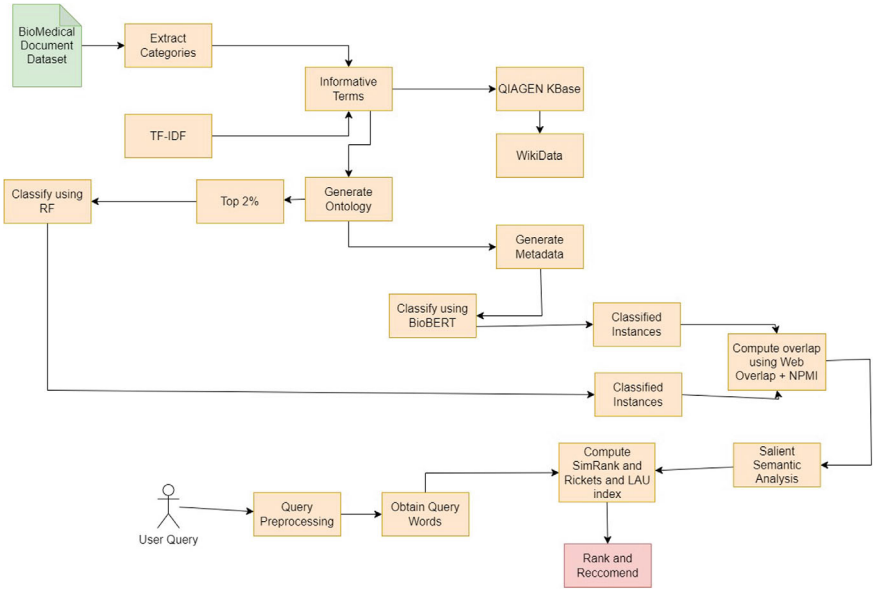


Fig. 1 Architecture of the BDSLL framework

from this pipeline are subjected to SimRank computation with the Ricklefs and Lau index. SimRank is computed for the preprocessed query words at a threshold of 0.50. Ricklefs and Lau index is set to a step deviance of 0.1. These thresholds are empirically set at a mediocre level, specifically for the SimRank owing to its strength and the computation of final similarity measures with that of the query as two levels of semantic similarity-based regulation have already been achieved. The entities that come out of this pipeline are ranked and recommended in the increasing order of the SimRank. These facets are further mapped with the categories and the keywords of the document dataset and the documents are prioritized, ranked, and recommended to the users.

Word overlap is a simple measure that can compute the count of the number of words shared between two documents. In line with this idea, we have a new technique of topic overlap called web overlap where similarity in web indices is computed for topic modelling.

BioBERT is a domain-specific LLM that works on the principles of BERT. It has been trained on a set of biomedical data and is therefore well suited for various tasks within the biomedical domain. The BioBERT is well suited for many tasks including question answering, named entity recognition and classification.

In the context of the BDSLL, the BioBERT is used to effectively classify metadata based on their categories and fields. Identifying fields related to biomedical demographics, experimental conditions, etc. It can extract and classify relevant information from metadata fields like keyword, concept, and relationship extractions. The usage

of BioBERT enhances and facilitates the discovery of new biomedical data resources if provided with a semantically rich representation of metadata.

Overall, BioBERT's ability to capture semantics and context of biomedical language makes it a valuable tool for classifying metadata in the biomedical domain. It can enhance the organization, searchability, and accessibility of biomedical data within the model.

The Ricklefs and Lau index is important in semantic relatedness computation. It is a niche overlap between two species based on the idea that similar niches will have similar resource requirements and therefore will overlap in their distribution. The Ricklefs &Lau index is calculated in Eq. 1.

$$R\&Lindex = \left(\frac{1}{2}\log 2\right)\sum_{i[}(p_i + q_i)\log(p_i + q_i) - p_i \log p_i - q_i \log q_i \quad (1)$$

where p_i is the proportion of resource i used by species 1, and q_i is the proportion of resource i used by species 2. $\sum$ is the sum of all resources.

NPMI, which stands for normalized pointwise mutual information, is an extension of the pointwise mutual information (PMI) measure that addresses the issue of bias towards rare events in co-occurrence analysis. NPMI normalizes the PMI value by dividing it by a negative logarithm of the probability of co-occurrence, resulting in a score between -1 and 1 that accounts for the informativeness of the co-occurring events relative to their individual probabilities. The NPMI function is defined by Eq. 2:

$$\mathrm{NPMI}(a, b) = \frac{PMI(a; b)}{h(a, b)} \quad (2)$$

where a and b are a pair of outcomes belonging to discrete random variables A and B which is a quantifiable entity of the discrepancy given their joint and individual distributions. PMI(a, b) is the pointwise mutual function of a and b. While $h(a, b)$ shows joint self-information, i.e. $\log_2 p(a, b)$.

4 Implementation, Performance Evaluation, and Results

Implementation of the proposed BDSLL framework was carried out using Python 3, with Google Colaboratory as the preferred IDE. Python's NLTK framework was used to perform common NLP tasks like lemmatization, stop word removal, etc. TensorFlow has been used to implement the learning models involved.

Experimentations were conducted on three distinct datasets that have been strategically integrated into a single large dataset. These datasets are the integration of 'Xindi Wang; Robert E. Mercer; Frank Rudzicz (2022) MeSHup Dataset', the

'Junseok Park (2019). Data for: Concept Embedding to Measure Semantic Relatedness for Biomedical Information Ontologies dataset', and the '3. Kush Attal; Brian Ondov; Dina Demner-Fushman (2023). PLABA Dataset'. At first, the categorical dataset is kept as is, and the non-categorical dataset values are converted into factors by category extraction using a customized annotator. The categories are then populated followed by using a customized crawler to crawl the documents for constructing the ontological dataset. A singular large dataset is thus constructed upon which the experiments were carried out for 2, 4, 6, and 8 queries. The ground truth was collected by asking several biomedical experts, including undergraduate and postgraduate students from various institutes and universities, to provide the top ten most relevant keywords for a given topic over 1–20 days. The experts were also asked to provide the corresponding research papers and their unique identifiers (DOIs). This information was used to establish the ground truth for the experiments.

The performance of the proposed framework is evaluated using precision, recall, accuracy, F-measure, and FDR as potential metrics. The primary metrics are used to measure the relevance of results while the FDR is used to measure the number of false positives.

Table 1 indicates that the proposed BDSLL model gives the highest precision, recall, accuracy, and F-measure of 96.56, 96.93, 96.74, 96.74%, and an FDR of 0.04. The comparisons are made with three other similar models: R2DCLT, DLDR, and SSDL. There models do not hold up to the proposed model in any of the primary metrics.

To evaluate the performance of the proposed BDSLL, it is compared with three similar models, namely the R2DCLT, DLDR, and the SSDL frameworks. The proposed BDSLL can produce a precision, recall, accuracy, and F-measure percentage in its average form of 96.56, 96.93, 96.74, and 96.74%, respectively, with an FDR of 0.04. This is significantly better performing than the other frameworks considered. This is because it uses domain-specific knowledge sources like QIAGEN Which are specific to the domain of biomedicine and is contributed and verified by domain experts. The quality of the sources and the quality of instances that are thus induced into the model are very good. The presence of a large language model for classification is the BioBERT that classifies the generated metadata. The BioBERT is also highly specific to the domain in question. It works competently in comparison with other deep learning approaches. In addition to these, there is also an induction of semantic understanding into the model by using several semantic similarity measures.

Table 1 BDSLL metrics against other approaches

Model	Precision %	Recall %	Accuracy %	F-measure %	FDR
R2DCLT [1]	87.34	90.07	88.70	88.68	0.13
DLDR [2]	90.12	92.45	91.28	91.27	0.10
SSDL [3]	92.73	94.81	93.77	93.76	0.08
Proposed BDSLL	96.56	96.93	96.74	96.74	0.04

The DLDR is a deep learning-based multi-turn conversation modelling for diagnostic Q&A document recommendation. This framework has a very strong deep learning model but a very shallow encompassment of knowledge incorporated into the framework. The unavailability of good data along with the absence of semantics-oriented analysis causes the DLDR to underperform in comparison with the proposed BDSLL.

The SSDR is a self-supervised scientific document recommendation using contrastive learning that uses coupling of text pair embedding with a self-supervised learning model that feeds knowledge and follows a divisive approach for classification where the constructive knowledge using intra and inter-document pairs of document embeddings giving some amount of knowledge. This knowledge, however, becomes highly specific because of its contrastive learning which acts as both an advantage and disadvantage that entirely depends on document availability and the density of knowledge. The uniformity of knowledge is not maintained statically. It can cause the dataset categories to become unbalanced which results in a classification that may require oversampling or under sampling, affecting the overall recommendation accuracy.

The R2DLT framework is the retrieving relevant documents using cosine similarity and LDA in text mining. LDA is a topic modelling framework, and these are capable of adding to the amount of knowledge in the framework, and the knowledge being highly relevant is a positive aspect of the framework. However, the amount of knowledge added using the LDA is minimal, and the LDA alone cannot satisfy the knowledge requirements of such a framework. Cosine similarity cannot sufficiently quantify the quality of semantic similarity. Additionally, instead of the datasets being compared individually with the categories, using a classification approach could improve the model. For these reasons, it is outperformed by the BDSLL.

Figure 2 depicts the precision versus number of recommendations curve. It can be observed that the BDSLL framework is at the highest position in the line graph. This is because it is a semantically inclined framework with strong knowledge bases and integration that uses an LLM as a classifier. The lack of the same, i.e. strong learning models, absence of dense knowledge bases, and a lack of semantics computation measures is the reason the other frameworks underperform in comparison.

5 Conclusion

The proposed framework for biomedical document recommendation, leveraging knowledge-empowered semantics and large language model. Through a special focus on strategic knowledge integration of knowledge in addition to the rich document corpus, such as the use of WikiData, QIAGEN biomedical knowledge base, and the usage of BioBERT classifier along with several semantics relatedness computation measures. The BioBERT being an LLM is applied as a classifier on the metadata to produce and contribute to the optimized classified instances within the model. The random forest classifier is also applied to this purpose, and therefore, the model can

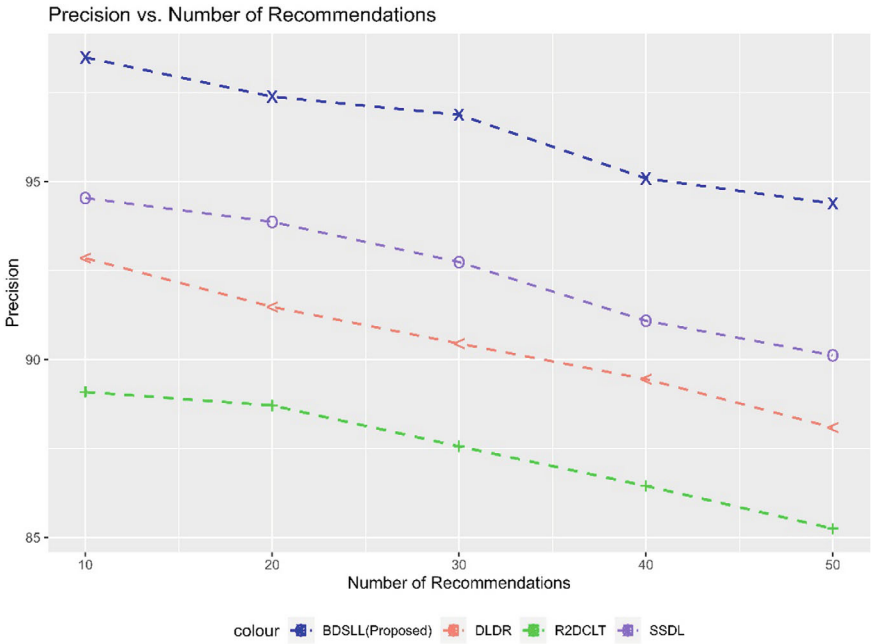


Fig. 2 Precision versus number of recommendations chart

maintain a highly relevant set of instances upon which user queries are processed. The BDSLL framework uses strong learning paradigms and knowledge-addition techniques to improve accessibility. Since the BDSLL utilizes all of these technologies, it is able to produce some of the best performance metrics in comparison with similar models and frameworks.

References

1. Yang Z, Xu W, Chen R (2021) A deep learning-based multi-turn conversation modeling for diagnostic Q&A document recommendation. Inf Process Manage 58(3):102485
2. Tan S, Zhang T, Zhao S, Zhang Y (2023) Self-supervised scientific document recommendation based on contrastive learning. Scientometrics 128(9):5027–5049
3. Ramya RS, Singh G, Sejal SN, Venugopal KR, Iyengar SS, Patnaik LM (2021) R 2 DCLT: retrieving relevant documents using cosine similarity and LDA in text mining. Int J Inf Commun Technol 19(4):391–422
4. Tommasel A, Pablos Sarabia R, Assent I (2023) Re2Dan: retrieval of medical documents for e-health in Danish. In: Proceedings of the 17th ACM conference on recommender systems, pp 1208–1211
5. Vijay M, Deepak G, Santhanavijayan A (2023) FSOL: financial document recommendation using hybridized semantics oriented learning. In: International conference on electronic governance with emerging technologies. Springer Nature Switzerland, Cham, pp 125–139

6. Roberts GL, van Dam A, Zeleznik R, Meyrowitz N A User-controlled document recommendation system for knowledge workers
7. Kumari J, Deepak G, Santhanavijayan A (2023) RDS: related document search for economics data using ontologies and hybrid semantics. In: International conference on data analytics and insights. Springer Nature Singapore, Singapore, pp 691–702
8. Wegner P, Balabin H, Ay AC, Bauermeister S, Killin L, Gallacher J, ... Salimi Y (2023) Semantic harmonization of Alzheimer's disease datasets using AD-Mapper. bioRxiv, 2023–10
9. Pudasaini S, Shakya S (2023) Question answering on biomedical research papers using transfer learning on BERT-base models. In: 2023 7th international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC). IEEE, pp 496–501
10. Reese T (2011) Automated metadata harvesting: low-barrier MARC record generation from OAI-PMH repository stores using MarcEdit. Libr Resour Tech Serv 53(2):121–134

A Review of Digital Twin and the Metaverse from the Standpoint of Smart Factories

Florin Zamfir and Emil Pricop

Abstract This paper aims to identify the impact of emerging technologies such as digital twins and metaverse within smart factory departments. The central departments related to both production and the operation of the entire factory are considered. More specifically, we want to classify technologies according to the needs of these departments and identify those functionalities that would aim to optimize production and workflow. Digital twins and Metaverse, in general, can be used in several distinct situations, such as monitoring and prediction, simulation, replication, training, management, and social networking. The paper also describes how enterprise resource planning, manufacturing execution systems, and product lifecycle management systems help generate the information needed to implement metaverse solutions in a smart factory.

Keywords Digital twins · Metaverse · Smart factory · Virtual reality · Industry 5.0 · Cyber-physical systems

1 Introduction

Regarding the various stages of the evolution of the industrial process, emphasis is placed on the transition from Industry 4.0 to Industry 5.0. For Industry 4.0, it was desired to integrate cyber-physical systems, Internet of things (IoT), cloud computing, big data, additive manufacturing, artificial intelligence, and cybersecurity. Industry 5.0 takes elements from the previous stage and aims at collaboration between robots and humans. Industry 5.0 brings to the forefront those technologies that can achieve this desideratum: artificial intelligence, virtual and augmented reality, collaborative robots, and blockchain.

Digital twin emphasizes the visual representation of physical objects but does not provide enough realism regarding the physical interaction between the user and the

F. Zamfir · E. Pricop (✉)
Petroleum-Gas University of Ploiesti, Ploiesti, RO, Romania
e-mail: emil.pricop@upg-ploiesti.ro

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_12

simulated process. The lack of realism is justified from several perspectives in reference [1]: the level of detail of parts or components is limited; not all senses (auditory, visual, olfactory, and tactile) can be faithfully rendered; the synchronization between DT and the physical process cannot always be achieved in real time.

In addition, other limitations can be related to the information flow at a factory level, for example, the integration with other departments that do not belong to the process but are essential, such as supply chain, human resources, finance, legal, and compliance.

These aspects reveal that it is not enough to use the digital twin technology to consider a factory part of the new industrial revolution. Adding other components of the metaverse can make the transition from Industry 4.0 to Industry 5.0. The metaverse can achieve the critical goal of Industry 5.0, the human-centric approach. It involves the use of collaborative robots (cobots) to augment human capabilities [2], virtual reality (VR), and augmented reality (AR) to enhance learning and skill development [3], along with wearable devices and exoskeletons to empower workers [4].

2 Smart Factories and Emerging Technologies

Implementing a smart factory in the metaverse can be useful from several points of view, whether we are discussing Simulations and digital twins, cloud computing, building information modeling (BIM), geographic information system (GIS), blockchain, digital assets, automation, predictive maintenance, IoT, VR, and AR technologies.

Unlike a classic simulation, a metaverse implementation allows a high degree of interaction between the user and the process. Such an implementation must comply with the basic rules specific to the metaverse and the conditions and restrictions imposed by a dynamic work environment with many fields of activity.

Table 1 briefly presents various departments in the smart factory and the needs of each department in terms of emerging technologies.

2.1 Production, Engineering, Research, and Development

Including the metaverse in the manufacturing process has the effect of changing the paradigm to wisdom manufacturing [5]. The classical approach of a manufacturing process is based on physical equipment (field equipment—transducers and actuators; control equipment—PLC, controllers; centralized control—SCADA).

Obtaining a cyber-physical system (CPS) requires a communication layer located over the physical layer, which can be implemented with the Internet of things (IoT). At this stage, we can talk about the digital twins, and the CPS level is governed by the Internet of services (IoS) or Internet of contents and knowledge (IoCK). Metaverse

Table 1 Specific requirements of each department concerning the metaverse and DT

Departments	Digital twins	Metaverse
Production	Real-time monitoring Performance analysis Process optimization	Virtual factory design Digital twin integration Remote operation and maintenance
Maintenance	Predictive maintenance Performance monitoring Lifecycle management of assets	Remote site inspections Remote equipment monitoring Remote assistance and troubleshooting
Logistics	Optimization of supply chain Real-time inventory tracking Resource allocation	Supply chain visualization Virtual warehouse management Remote fleet management
Quality control	Real-Time quality monitoring Predictive quality analytics Automated quality control Workflows	Virtual quality inspections Training for quality inspectors Quality dashboard Virtual audits and certifications
Engineering	Product design and development Performance monitoring Energy efficiency optimization	Immersive design Virtual prototyping and simulation Virtual product demonstrations
Human resources	Employee performance monitor Skill development and training Career path simulation	Collaborative meetings Employee engagement Performance reviews
Finance department	Financial asset representation Risk simulation Budgeting and forecasting Audit and compliance Fraud detection and security	Virtual financial markets Blockchain and cryptocurrency Transactions Data visualization, virtual conferences Real-time portfolio monitoring
Research and development	Sensor integration for data Collection Materials testing and optimization Remote experimentation and monitoring Simulation of manufacturing processes	Collaborative design and planning Virtual laboratories and Experimentation Virtual prototyping and design Virtual reality simulations
Sales and marketing	Real-time customer behaviour analysis Customer engagement forecasting Market expansion simulation product placement simulations	Virtual product showrooms Interactive product launches Immersive advertising spaces Virtual try-ons and expos

(continued)

Table 1 (continued)

Departments	Digital twins	Metaverse
Legal and compliance	Contract management Litigation risk assessment Legal research and case management Regulatory compliance	Virtual courtroom simulations Compliance training Data privacy audits Immersive ethics training
Information technology & cybersecurity	Cybersecurity threat detection Disaster recovery planning IT service management Cloud resource management Software development & testing	Remote IT collaboration Simulated IT environments Cybersecurity awareness programs Digital asset management
Management and administration	Facility management Space planning and allocation Workforce management Workflow automation and Optimization	Virtual production monitoring Project management and Coordination Virtual onboarding Virtual administrative meetings and committees
Health and safety	Predictive safety analytics Emergency response simulations Occupational health monitoring Compliance monitoring and reporting	Remote site inspections Virtual safety training programs Risk assessment, hazard identification Virtual incident reconstruction Virtual safety audits and inspections
Training	Simulated equipment training Emergency response training Soft skills development Performance assessment and feedback	Collaborative meetings and training Virtual career fairs Interactive role-playing scenarios Virtual mentorship programs

involves generating a social network by bringing humans to the forefront, and at this level, we can speak of the Internet of people (IoP) [5].

Digital twins can be used to simulate the production process to optimize workflow. Using a DT, engineers can identify problems that occur (bottlenecks), make optimizations (improve efficiency), or minimize defects in manufactured products.

With technological evolution and the expansion of the use of DT in smart factories, the possibility arose for designers and engineers to work with this technology before the factory was built. In this way, engineers can use a DT of an existing factory and customize it for the new working conditions of the new premises [6].

Metaverse facilitates development by immersing users in a 3D environment that provides many visualization tools. These environments are based on methods that can test the success rate of the new metaverse created in that field, such as exploratory

factor analysis (examination of clusters of interrelated variables) and structural equation modeling (analyzing complex relationships between indicators and constructs) [7].

R&D teams can have many benefits using the metaverse:

- Use of virtual laboratories to conduct experiments and simulations.
- Collaboration with other teams in multidisciplinary projects.
- Prototyping for new products.
- Participation in virtual conferences and seminars.

2.2 Maintenance, Quality Control, Health, and Safety

The transition to smart factories creates an additional problem for the maintenance department. Engineers and technicians need to get used to new technologies; in some situations, it is simpler to use diagnostic interfaces, digital twins, AI-based assistance, and extended reality. An engineer can perform equipment maintenance using DT because, and in this way, he would have access to analysis modules that are based on AI models, or he can use an AR headset to display additional information or even generate a hologram of an AI-generated expert to guide him step by step in accomplishing maintenance tasks [8].

Similar to the approach to the maintenance department, the metaverse can bring substantial improvements within the quality control department. Digital twins allows quality inspectors to interact with virtual replicas to identify various defects, fix errors in working procedures, or identify opportunities to optimize processes. Moreover, using the metaverse, inspectors can access a centralized and collaborative dashboard to analyze quality metrics and track key performance indicators (KPIs).

Quality control can be automated, especially if a specialized IoT infrastructure is used at the process level, such as the one described in the paper [9]: an artificial intelligence of things (AIoT) network. The authors describe an implementation for quality control of ink cartridge manufacturing lines requiring machine learning algorithms for anomaly detection.

An essential aspect of the production process is human operators' safety, especially in the context of human–robot collaboration. In such a situation, collaborative robots must be able to modify tasks depending on specific events, such as the appearance of an operator in an area that may endanger his safety. The paper [2] proposes an operator detection system that is based on several working scenarios:

- **Safe**: There is no operator in the robot's work area; therefore, the robot will continue to work normally.
- **Potential**: an operator enters the robot's work area, but the areas associated with the two entities do not intersect, and because of this, the robot will change its working speed so that the operator can avoid an accident.
- **Dangerous**: An operator enters the robot's work area, and the areas associated with the two entities intersect, which causes the robot to stop to avoid an accident.

Another way of implementing occupational safety can be realized through prevention measures, which are materialized by training staff working in critical environments using an interactive virtual environment. Training can be done through interactive games or by going through procedures in a virtual environment that does not endanger the employee's life [3].

2.3 Logistics, Finance, Sales, and Marketing

As far as the logistics department is concerned, using the metaverse can bring many benefits: optimizing the purchase flow, monitoring inventory, optimizing delivery schedules, and minimizing costs related to transport [10]. Another important aspect is that the information regarding the supply chain is complex and belongs to multiple information systems, so adding this information to the metaverse can have the effect of making the distribution chain transparent [11].

Metaverse can be used in finance to obtain various advantages, such as creating business models, predicting customer behavior and market trends identification, and creating immersive data visualizations. Financial assets can be stored securely with the help of blockchain and non-fungible token (NFT) technologies [12].

Regarding the finance department in a smart factory, other aspects may be worth considering: virtual representations of financial reports, digital twins of cash flows or budget scenarios, and virtual audit trails for financial transactions.

It can be said that one of the strengths of the metaverse is marketing. Metaverse provides many tools through which users can have access in a simple or even complex form to the products they are about to purchase. The best-known examples are virtual advertising (immersive advertising spaces), virtual commerce and try-ons (customers can visualize how products look), virtual influencers (promote products and engage with their virtual followers), social engagement (virtual rewards and perks for customers), data analytics (customer preferences and behaviors), data personalization (personalized virtual experiences), and cross-platform marketing (digital platforms integration) [13].

2.4 Information Technology and Cybersecurity

The IT department can make full use of the resources offered by emerging technologies, whether it is DT that can be used in real-time monitoring of the network, servers, virtual machines, or metaverse where virtualization can be achieved in terms of helpdesk, training, cybersecurity scenario, digital assets, or even virtualization of the entire IT infrastructure.

A concrete case of using the metaverse within a smart factory is presented in the paper [14]. The authors demonstrate the usefulness of emerging technologies in stopping attacks on a smart grid.

To have all the benefits of the metaverse and digital twins, the entire hardware infrastructure must be known in real time by software implementations. As a result, this information must reach the metaverse with the help of IoT networks. This can also have a negative effect, given that these networks can be a critical point regarding data security. To mitigate this effect, it is necessary to use artificial intelligence, blockchain technologies, and other security and preventive mechanisms against cyber-attacks. The proposed solution consists of implementing a blockchain-based authentication scheme and an intrusion detection system configured with the help of deep learning algorithms [14].

A further step is presented in the paper [15], where the authors consider that DT can also become a cybersecurity risk. Cyber-attack identification is done with the help of the cyber kill chain method. This method involves identifying malicious activities using eight steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, actions on objective, and monetization [16]. Attacks on CPS and DT are different; for example, CPS attacks are about compromising devices, modifying data, and taking control. In contrast, as far as DT is concerned, attacks have the following objectives: identifying system behavior, desynchronizing services, and using acquired knowledge to attack the physical system [15].

2.5 Human Resources, Legal and Compliance, Training, Management, and Administration

Metaverse can be used in HR to manage its employees and facilitate access to Crowdworks platforms. An integrated system can allow the management of company employees, for example, associating workers in various compartments, conducting training sessions, running scenarios on the interaction between employees, and avoiding conflict [17].

In reference [18], the authors present a case study on implementing a smart contract that allows users to enter data, use reports, and act based on predictive analysis decisions. User data are linked to existing administrative spaces and environmental information.

3 Metaverse Architecture of Smart Factory

In an intelligent factory, the employees must evolve, too, and new jobs must consider specific working conditions required by emerging technologies. Within a smart factory, the following types of users who can get in touch with the metaverse infrastructure can be distinguished: Subject matter experts (SMEs), data scientists, software

developers, system administrators, engineers and operators, managers, and stakeholders. Table 2 presents the main roles related to the metaverse and digital twins concepts associated with the users listed above.

The activities of the departments described in the previous section of this paper can be implemented at the informational level by a complex information system containing four components: enterprise resource planning (ERP), product lifecycle management (PLM), manufacturing execution system (MES), and knowledge base system (KBS) [19]. The knowledge base system is a separate entity containing information other systems can use. This is an up-to-date database with information from all factory processes: production, planning, financial, human resources, etc.

ERPs are software platforms that integrate a company's business processes: financial, human resources, supply chain, asset management, inventory, marketing, and customer relationship management. PLM is the tool that tracks the workflow associated with a product from product design, creation, and disposal. The main components of PLM are project management, parts management, change management, requirements management, and quality management. MES is a platform for real-time monitoring and control of the production process. The critical components of the ESM can be listed as follows: resource allocation and status, production planning, data collection and acquisition, quality management, maintenance management, and work order management [19].

Table 2 Role of users in the context of metaverse and DT

User	Roles
Subject matter experts	Provide insights, validate the data sources, define running requirements, monitor the performance, and define the modeling algorithms
Data Scientists	Analyze and preprocess collected data, apply statistical techniques and data mining methods, select modeling techniques, validate and evaluate predictive models, develop diagnostic models, and create visualizations and dashboards
Software developers	Data integration, system architecture design, application development, design user interfaces, write unit, and integration tests
System administrators	Infrastructure setup, deployment and installation, performance tuning, security management, and backup and disaster recovery
Engineers and operators	Monitor the real-time data, operational parameters adjustment, schedule maintenance activities, observe key performance indicators, use fault detection algorithms, and diagnose equipment malfunctions
Managers	Allocate financial and human resources, align metaverse projects with overall business goals, develop budgets and resource plans, and identify and mitigate risks associated with metaverse projects
Stakeholders	Set the strategic direction for metaverse and DT projects, ensure that DT address real-world needs, and collaborate with other organizations to co-develop metaverse solutions

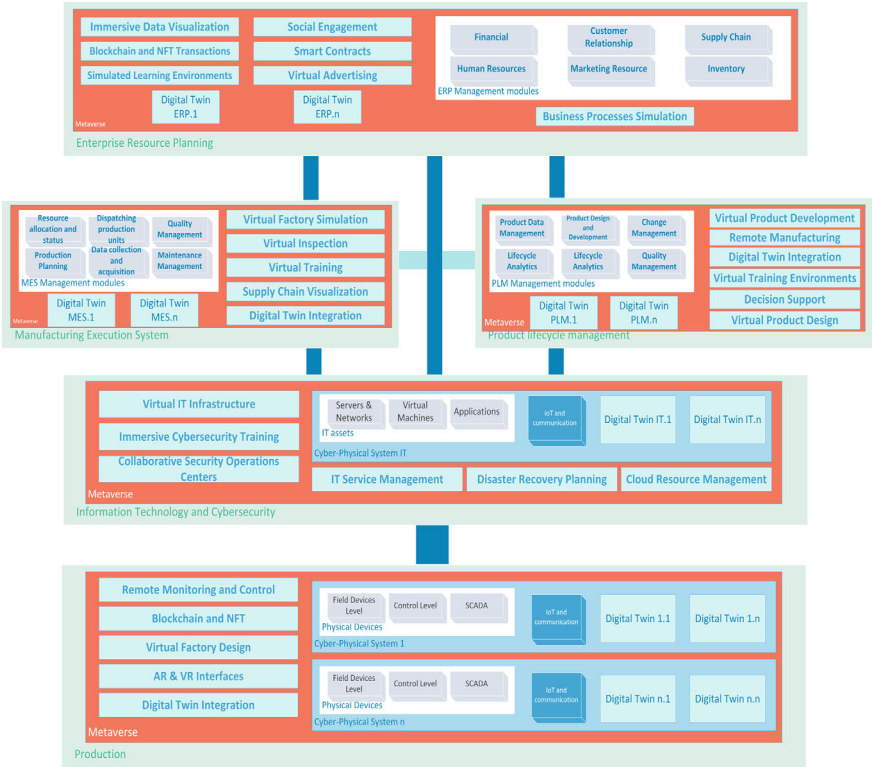


Fig. 1 Architecture of systems that interact with the metaverse

Given that the information received from DT can be included in KBS by integrating MES, ESP, and PLM with KBS, the metaverse functionalities described in Fig. 1 can be achieved.

4 Conclusion

Emerging technologies like digital twins and the metaverse can bring multiple benefits to a company. This paper shows that these technologies can benefit other departments, not just those in the production chain. The factory must once again become a space where humans are the point of interest, and all these technologies must facilitate the transition to Industry 5.0. To achieve this, all possibilities of human interconnection must be considered, in other words, bringing them to the virtual center of the factory. Traditional systems such as ERP, MES, and PLM contain components that help achieve this goal; moreover, in recent years, they have been

updated with modules based on artificial intelligence. But to achieve complete integration, we must also consider the technologies underlying the metaverse: virtual and augmented reality, blockchain, NFTs, and smart contracts.

Acknowledgements/Funding This work was supported by a grant from the Petroleum-Gas University of Ploiesti, Romania, project number 11060/08.06.2023, IME-EMERGENSEC, within the Internal Grants for Scientific Research.

References

1. Starly B, Koprov P, Bharadwaj A, Batchelder T, Breitenbach B (2023) Unreal factories: next generation of digital twins of machines and factories in the industrial metaverse. Manuf Lett 37. https://doi.org/10.1016/j.mfglet.2023.07.021
2. Wang S, Zhang J, Wang P, Law J, Calinescu R, Mihaylova L (2024) A deep learning-enhanced Digital Twin framework for improving safety and reliability in human–robot collaborative manufacturing. Rob Comput Integr Manuf 85:102608. https://doi.org/10.1016/j.rcim.2023.102608
3. Kwegyir-Afful E, Kantola J (2021) Simulation-based safety training for plant maintenance in virtual reality. In: Cassenti DN, Scataglini S, Rajulu SL, Wright JL (eds) Advances in simulation and digital human modeling. Springer International Publishing, Cham, pp 167–173
4. Ralfs L, Hoffmann N, Glitsch U, Heinrich K, Johns J, Weidner R (2023) Insights into evaluating and using industrial exoskeletons: summary report, guideline, and lessons learned from the interdisciplinary project "Exo@Work"'. Int J Ind Ergon 97:103494. https://doi.org/10.1016/j.ergon.2023.103494
5. Yao X, Ma N, Zhang J, Wang K, Yang E, Faccio M (2024) Enhancing wisdom manufacturing as industrial metaverse for industry and society 5.0. J Intell Manuf 35(1):235–255. https://doi.org/10.1007/s10845-022-02027-7
6. Li S et al (2023) Modeling of digital twin workshop in planning via a graph neural network: the case of an ocean engineering manufacturing intelligent workshop. Appl Sci 13(18). https://doi.org/10.3390/app131810134
7. Waqar A et al (2023) Analyzing the success of adopting metaverse in construction industry: structural equation modelling. J Eng 2023:8824795. https://doi.org/10.1155/2023/8824795
8. Bordegoni M, Ferrise F (2023) Exploring the intersection of metaverse, digital twins, and artificial intelligence in training and maintenance. J Comput Inf Sci Eng 23(060806). https://doi.org/10.1115/1.4062455
9. Le D, Yang J, Zhou S, Ho D, Tan R (2023) Design, deployment, and evaluation of an industrial AIoT system for quality control at HP factories. ACM Trans Sens Netw 20. https://doi.org/10.1145/3618300
10. Trivedi S, Negi S (2023) The metaverse in supply chain management: application and benefits. Int J Adv Virtual Real 1(1):36–43
11. Abouzid I, Saidi R (2023) Digital twin implementation approach in supply chain processes. Sci Afr 21:e01821. https://doi.org/10.1016/j.sciaf.2023.e01821
12. Yathiraju N, Dash B (2023) Big Data and metaverse revolutionizing the futuristic fintech industry. AIRCCs Int J Comput Sci Inf Technol 1–13
13. Nagarajan G, Moorthy VS, Mohamed AK, Mohideen AS, Ishaq MM, Lakshmi MR (2023) The role of the metaverse in digital marketing. JETT 14(5):51–59
14. Kumar P, Kumar R, Aljuhani A, Javeed D, Jolfaei D, Islam AKMN (2023) Digital twin-driven SDN for smart grid: a deep learning integrated blockchain for cybersecurity. Sol Energy 263:111921. https://doi.org/10.1016/j.solener.2023.111921

15. Suhail S, Iqbal M, Hussain R, Jurdak R (2023) ENIGMA: an explainable digital twin security solution for cyber–physical systems. Comput Ind 151:103961. https://doi.org/10.1016/j.compind.2023.103961
16. Kour R, Karim R, Dersin P (2024) Game theory and cyber kill chain: a strategic approach to cybersecurity. In: Kumar U, Karim R, Galar D, Kour R (eds) International congress and workshop on industrial AI and maintenance. Springer Nature Switzerland, Cham, pp 451–463
17. Pivnička M, Hrušecká D, Hrbáčková L (2022) Introduction of a new flexible human resources planning system based on digital twin approach: a case study. Serbian J Manag 17(2):361–373
18. Tavakoli P, Yitmen I, Sadri H, Taheri A (2023) Blockchain-based digital twin data provenance for predictive asset management in building facilities. Smart Sustain Built Environ 13. https://doi.org/10.1108/SASBE-07-2023-0169
19. Avvaru VS, Bruno G, Chiabert P, Traini E (2020) Integration of PLM, MES and ERP systems to optimize the engineering, production and business. In: Presented at the IFIP advances in information and communication technology, pp 70–82. https://doi.org/10.1007/978-3-030-62807-9_7

A Comprehensive Patient Triage Algorithm Incorporating ChatGPT API for Symptom-Based Healthcare Decision-Making

Cosmina-Mihaela Roșca, Răzvan-Alexandru Bold, and Alexandru-Eduard Gerea

Abstract In health care, it is crucial to organize patient care and resources based on the urgency level so that patients whose lives are in danger are treated as a priority. This study introduces a new algorithm, created in C# using new technologies such as ChatGPT, to improve the sorting and prioritizing of patients based on the National Urgency Level. The algorithm aims to demonstrate the possibility of real-time reporting of the occupancy level of all hospitals. This way, patients can head to the nearest and most available hospital based on both proximity and urgency level. To identify the urgency level, the algorithm utilizes a widely used external service, ChatGPT. Additionally, the algorithm calculates the average waiting time based on the number of patients, their urgency level, and the number of on-duty doctors. To achieve these goals, the algorithm's implementation has been carried out using modern technologies, such as C# programming language, ranging from the artificial intelligence service ChatGPT to entity framework for managing data in the Microsoft database. The article addresses two major issues, namely text-based classification into a specific category based on common characteristics, such as a patient's symptoms concerning a specific urgency level. The second issue is represented by the proposed calculation formula for real-time estimation of waiting time, based on the identified urgency level.

Keywords ChatGPT · Healthcare · Algorithm · Artificial intelligence

C.-M. Roșca (✉) · R.-A. Bold · A.-E. Gerea
Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, 39 Bucharest Avenue, 100680 Ploieşti, Romania
e-mail: cosmina.rosca@upg-ploiesti.ro

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_13

1 Introduction and Literature Survey

This paper presents a study on modernizing the public medical system to minimize the waiting time in the emergency room. In this regard, the patient triage process can be initiated based on symptoms entered by the user even before reaching the hospital. Achieving such an objective requires the use of an artificial intelligence service, such as ChatGPT. This service is a conversational artificial intelligence (AI) model developed by OpenAI. It is based on the generative pre-trained transformer (GPT) architecture and is designed for natural language understanding and generation in a chat or conversation format. ChatGPT has been trained using multiple inputs and is capable of generating coherent and contextually relevant responses to user input in various conversational scenarios.

The paper [1] explores the potential applications of ChatGPT/GPT-4 in intensive care medicine (ICM) to address challenges posed by the COVID-19 pandemic. It focuses on knowledge augmentation, device management, clinical decision-making support, early warning systems, and the establishment of an ICU database. The study includes insights from an online survey highlighting ChatGPT/GPT-4's role in real-time monitoring and diagnosis of ICU patients.

Paper [2] investigates the integration of large language models (LLMs), exemplified by ChatGPT, into human–robot collaboration. It introduces RoboGPT, a system facilitating natural language communication between human operators and a robot arm. The study's human-subject experiment shows increased trust in the collaboration process with ChatGPT, providing valuable insights into trustworthy human–robot collaboration systems. The same subject is covered in medical tasks and their potential in medical education and public health. The paper [3] emphasizes challenges such as data limitations and the need for responsible development.

Machine learning models in patient risk stratification, emphasizing clinician-initiated versus non-clinician-initiated data, are analyzed in this paper [4]. Deep learning models using clinician-initiated data show performance comparable to full electronic medical record (EMR)-based benchmarks. The study raises concerns about model predictions relying on clinical behavior rather than patient physiology, emphasizing the need for transparency in healthcare risk stratification models.

The possibility of using machine learning (ML) in disease diagnosis through bibliometric analysis is presented in the paper [5]. It delves into trends, algorithms, disease types, data types, and applications, highlighting ML's significance in overcoming diagnostic challenges, especially in resource-limited settings. The review concludes with insights into future trends and opportunities in machine learning-based disease diagnosis. Applications in report summarization, knowledge extraction, personalized health advice, medical education, and research are presented in the paper [6]. It discusses associated risks and underscores the importance of clinical validation and integration with existing healthcare systems. Researchers explore GPT-4's potential in improving diagnostic accuracy for older patients. GPT-4 analyzes medical histories and achieves higher accuracy than clinicians and a diagnostic decision support system. The study suggests GPT-4's promise in aiding clinical diagnoses

but emphasizes the need for clinician interpretation within the clinical context [7]. GPT-4 demonstrates superior accuracy in estimating probabilities after negative test results but varies in performance after positive test results. The study suggests the potential for AI models like GPT-4 to enhance diagnostic reasoning, particularly in specific scenarios [8].

The impact of big data and machine learning in health care is discussed in multiple AI applications ranging from diagnostics to treatment, aiding administrative workflows, clinical documentation, and image analysis [9–11]. The ChatGPT-4, a large multimodal model [11], achieves human-level performance on professional benchmarks, enhancing factuality and behavioral adherence. Med-PaLM 2 and ChatDoctor, other language models, excel in medical question answering, showcasing progress in aligning AI with medical knowledge [12–15].

Paper [13] introduces ChatDoctor, a medical chat model fine-tuned on LLaMA, demonstrating improved accuracy in understanding patient inquiries. The study emphasizes the potential of such advancements in enhancing medical communication.

Ongoing advancements in deep learning [16] hold promise for addressing challenges in biology and medicine, whereas the paper [17] discusses ChatGPT's potential impact on clinical medicine, research, and education, acknowledging limitations and advocating for responsible use. Concurrently, the study [18] introduces a data quality feedback tool for electronic health records (EHR) to foster awareness and improvement, and the paper [19] focuses on addressing overcrowding in emergency departments, proposing a novel multi-agent system approach, and highlighting the significance of improving patient care through ED overcrowding solutions.

Papers [20–25] explore medical urgency alignment, fever triage algorithms, CDSS evolution, ChatGPT ethics, AI in patient queries, and generative AI's medical applications, highlighting their significance and potential impact on healthcare practices and technology evolution.

The upcoming section will introduce a new algorithm that integrates communication with the ChatGPT API for identifying a patient's urgency level. Additionally, the algorithm will incorporate a calculation formula for determining the average waiting time for a patient. The results section illustrates various scenarios for assessing the algorithm's efficiency and its potential for widespread use. This section is followed by a set of conclusions and future directions for development.

2 Patient Triage Algorithm Using Artificial Intelligence

The primary objective of this paper is to streamline the emergency healthcare system, aiming to minimize the estimated waiting time and align it as closely as possible with the actual waiting time. The anticipated benefits include:

- Patients with a very high level of urgency will be prioritized for immediate attention;

- Patients whose lives are not in immediate danger can wait at home or near the hospital until their turn, preventing overcrowding in the emergency waiting area.

In Romania, the waiting time in emergency rooms is notably high, with patient triage being conducted by a nurse. Patients run the risk of contracting additional illnesses while waiting to complete the intake process for medical consultation. Moreover, the average waiting time in emergency rooms is steadily increasing, primarily due to individuals seeking non-urgent care at emergency departments [26].

For this purpose, the algorithm proposed in this paper employs the existing protocol in Romania, which includes five levels of emergency in-patient reception. These emergency levels are determined based on the patient's symptoms, depending on how life-threatening they are.

Currently, there are five urgency levels in the National Urgency Level [27]:

- Urgency level 1: cardiac arrest, massive bleeding;
- Urgency level 2: cardiac-related chest pain, asthma attack;
- Urgency level 3: abdominal pain, high fever with cough;
- Urgency level 4: simple laceration;
- Urgency level 5: rash, prescription refill.

A second major objective concerns exploring the feasibility of employing the ChatGPT service for generating alert levels. Given that ChatGPT is a natural language processing and text generation system, the potential use of the ChatGPT API serves as a case study for this article and is concurrently a trending topic.

The underlying concept is grounded in two significant advantages of utilizing the ChatGPT API for urgency-level classification:

- The capability to access an extensive medical database, which can expand the previously limited set of symptoms to a diverse range of symptoms with equivalent urgency levels.
- The ability to process and accurately associate user-inputted text with the appropriate terminology for urgency levels. For instance, a user may incorrectly input "hedache" instead of "headache". Another issue that this AI service can promptly address involves articulated or plural forms of words, complexities that would be challenging to integrate at the raw algorithmic level. For example, if the database contains the word "hemorrhage", and a user inputs "hemorrhages", the word would be challenging to associate with existing symptomatology.

Hence, the ChatGPT service's capacity to discern word meanings and associate them with other terms in a database constitutes a favorable advantage for exploring the possibility of integration into a software product capable of calculating urgency levels and real-time average waiting times.

The algorithm is integrated into a Web application with a user interface, utilizing the C# language and the ASP.NET framework. It facilitates patient registration by prompting them to input their symptoms through a Web interface. Subsequently, the algorithm processes this information, consulting its knowledge base in conjunction

with ChatGPT API to assess the urgency level associated with each patient's condition. Furthermore, once patients are categorized based on urgency levels, the algorithm continuously reassesses and updates the waiting times based on the real-time status of patients in the waiting area and the number of on-call medical personnel. This dynamic adjustment helps optimize the allocation of healthcare resources and ensures that patients with more pressing needs receive prompt attention.

The proposed real-time waiting time calculation formula is presented in (1). The equation takes into account the patient's urgency level, the number of previous patients and their priority levels, as well as the number of on-duty doctors. The patient intake process prioritizes cases that endanger patients' lives. Thus, a patient with an urgency level of 3, initially having a waiting time of 115 min, may have an updated waiting time of 200 min if other cases requiring priority intake arise.

$$\text{Patient Waiting Time} = \text{Global Waiting Time} + \text{Urgency Level Time} + \text{Offset} \quad (1)$$

Global waiting time represents the sum of the waiting durations for patients who are ahead of the current patient in terms of urgency. Urgency level time represents the time associated with the urgency level of the current patient. For a level 1 urgency, this time is 0; for level 2, it is 15 min, and so forth. The offset represents the delay time based on the on-call doctors. If the number of patients divided by the number of doctors exceeds the number of doctors, then the offset has a value of 10 (minutes).

To achieve the full operational efficacy of the algorithm, a stable connection to the Internet is needed. This can be a disadvantage, but by using this method, we are ensuring that every symptomatology can be covered using the ChatGPT API's large database. For a better understanding, the logical scheme of the patient triage and waiting time calculation is depicted in Fig. 1.

The algorithm works in an infinite loop once it starts. This is done because the program that contains this algorithm needs to run consistently, showing the waiting times for each patient in the emergency room itself. If there are any cases where the program needs to be closed, it can be made safely through closing the program window, without any data loss.

After a patient types the symptomatology in the registration phase, the application initiates the consultation process by querying the knowledge base. This step involves parsing and analyzing the user-input symptoms against the pre-existing repository of medical conditions and associated symptoms. In instances where the knowledge base lacks information about specific symptoms, the application seamlessly integrates the ChatGPT API to solicit insights. Upon receiving the dictated urgency level from ChatGPT, the application dynamically incorporates this newfound information into the knowledge base. This iterative process ensures a continual enhancement of the system's understanding and adaptability to a diverse array of medical scenarios.

Patients in the emergency room require immediate attention, and every second of the waiting process can be life-threatening. The algorithm was designed to give the fastest treatment possible. Once a patient is registered, the algorithm automatically

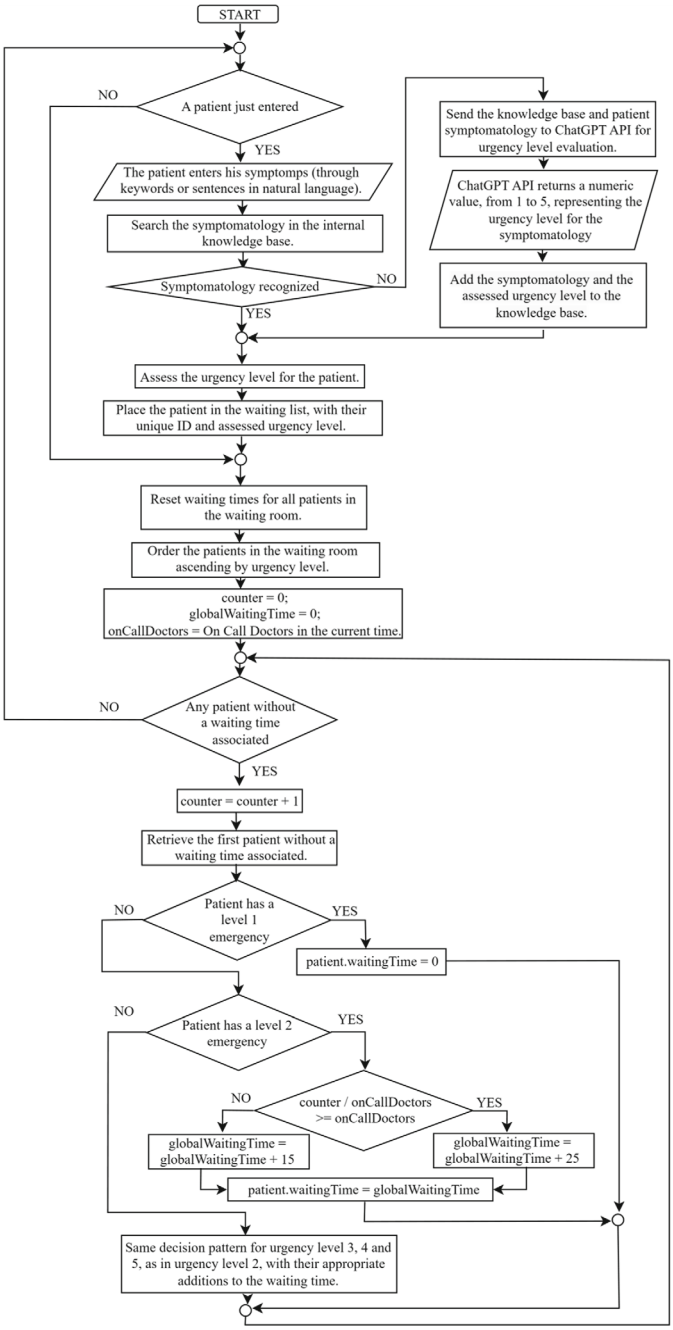


Fig. 1 Patient triage and waiting time calculation

computes their approximate waiting time for consultation by ordering the patients ascending by their urgency level and then calculating each patient's time using a global variable, as can be seen in Fig. 1. Furthermore, consideration has been given to instances where the number of doctors is significantly fewer than the patients in line, aiming to ensure that the waiting time closely mirrors reality.

The doctors in the emergency room need access to results provided by the algorithm to take the patients in need. For this reason, the system where the algorithm is contained needs an additional module that provides this service. The doctor will have a personal computer that accesses a Web application where the unique ID and symptomatology of the current patient will be displayed. The patient will go to the doctor's office when prompted by the waiting room display, indicating it is their turn. After the consultation, the doctor will write in a dedicated section the diagnosis of the patient, which is stored for a later record.

To have a real impact, the algorithm needs to be integrated into a platform that is easily accessible to anyone. For this, the development of the urgent reception application for this algorithm, which is responsible for patient triage and wait time allocation, was carefully orchestrated with ASP.NET.

The application flexibility ensured accessibility not only within the internal computing infrastructure of the urgent care unit but also across conventional computers and mobile devices. Furthermore, the chosen technology stack boasts inherent security measures vital for protecting sensitive patient data and ensuring compliance with stringent healthcare data protection standards. Additionally, the scalability of this solution accommodates future expansions and technological advancements seamlessly.

Figure 2 illustrates the comprehensive triage process diagram, starting from patient registration to their intake. Summarizing, the steps are as follows:

- The patient accesses the triage application via a mobile phone, laptop, or a physical kiosk in the hospital;
- The patient inputs all symptoms into the application's graphical interface;

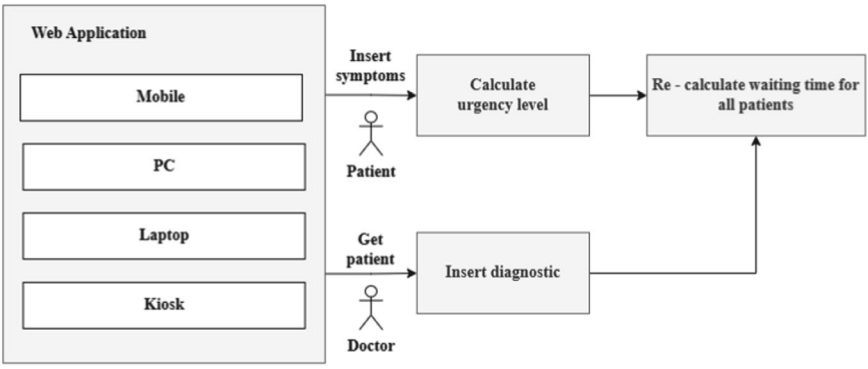


Fig. 2 Triage process diagram

- The application captures the user-entered text and, with the assistance of the ChatGPT service, identifies an urgency level;
- The application calculates the patient's waiting time and recalculates the waiting times for all patients;
- The application generates a unique patient identification code. If the patient is in the hospital's waiting room, a ticket containing the unique code is printed using a receipt printer;
- The waiting list, containing unique patient codes and the waiting time for each patient, is displayed in the application, and the hospital's waiting area is shown on a monitor;
- Doctors have their section within the application, allowing them to be aware of the number of on-duty doctors constantly;
- The application displays a button for doctors to take the first patient from the list. Upon pressing the intake button, the waiting times for patients are recalculated.

The triage system provides immediate action and exempts medical personnel from organizational tasks, so they can be focused on treating patients. It can provide a major advantage, but the use of ChatGPT API in this scenario should be approached with caution because it can cause life-threatening situations. Tests need to be made to make sure that the API assesses urgency levels correctly. The next chapter will discuss this aspect.

3 Results and Discussions

In instances where the knowledge base lacks the urgency level for the presented patient symptoms, the recourse is made to the utilization of the ChatGPT API. By using the API, the algorithm can associate urgency levels for any patient, no matter the articulation used in the symptomatology expression. This is a real advantage and makes the triage algorithm accessible for any person, without being constrained by certain writing rules. For the program to work in the real world, there is a need to test that it assesses urgency levels correctly, because it is used in life-threatening situations, and every little mistake is detrimental.

To make sure that ChatGPT API does the judging correctly, the test implies prompt assessment of an urgency level, from one to five, where the patient types his symptoms similar to those presented in Table 1, where their correct urgency level is known.

For assessing the urgency level, the following question will be asked to ChatGPT: "I have the following symptomatology: '<here we insert the symptomatology expression>'. Rate these symptoms on a scale from 1 to 5 based on severity using the emergency severity index, with 1 being the highest urgency level". Subsequently, it becomes imperative to extract solely the urgency number from the response, exemplified in this instance as "1", employing a regular expression.

For showcasing the results, the system that was developed for testing the algorithm is used. For urgency level 1, the following symptoms were typed in a textbox: "Chest

Table 1 Symptomatology for each urgency level

Urgency level	Example of an expression illustrating the symptomatology
Level 1	"Chest pain and trouble breathing!"
Level 2	"On-and-off chest pain feels like pressure"
Level 3	"Sharp abdominal pain"
Level 4	"Cut my hand, bleeding but not too much"
Level 5	"Developed rash, need a prescription refill"

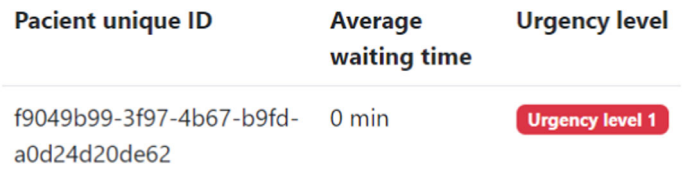


Fig. 3 Patient with urgency level 1 in the triage application

pain and trouble breathing!" which would suggest a cardiac arrest, one of the urgency level 1 cases. After the symptomatology is typed, the patient presses the registration button. After the button is pressed, the symptomatology is sent to ChatGPT and then processed. In this case, the response from ChatGPT was: "Chest pain: This symptom is often considered high urgency, potentially indicating a life-threatening condition. It could be assigned a rating of 1". In this case, the response is correctly classified.

Figure 3 presents an example of how the patient would look in the triage application. After the urgency level is assessed, the patient sees in the waiting room list that he is an urgency level 1 and should enter a consult as fast as possible.

For the phrase resembling urgency level 2 in Table 1, the following response is returned from ChatGPT API: "In a hypothetical situation, based on the severity of the described symptoms, a healthcare professional might assign a high ESI level, such as level 1". This response is incorrect because on and off chest pain was categorized as an urgency level 1 when it should have been an urgency level 2.

The assessment provided by the API for the symptomatology corresponding to urgency level 3 in Table 1 exhibited a degree of ambiguity rather than precision. The returned response suggested an indecision, stating: "sharp abdominal pain: This could indicate a serious underlying issue. Depending on the severity and other factors, it might be categorized as either level 2 or 3".

For the symptomatology in urgency level 4, the API returned: "I would rate this as a Level 4 on the ESI scale, indicating a low level of urgency", which is correct.

For the last urgency level, the API returned a generic response: "Based on the information you provided, a rash alone may not necessarily indicate a high level of urgency, especially if it is not associated with other severe symptoms". Due to the absence of a specific numerical value in its response, the algorithm faced challenges in utilizing this information for precise classification.

ChatGPT API successfully derived two out of the five urgency levels. In the majority of incorrect instances, it tended to allocate a level higher than what was deemed appropriate. This observation raises concerns about its current accuracy. Nevertheless, it is essential to acknowledge that over time, refinement and improvement can be achieved, particularly by seeking the input of medical professionals before the assignment of urgency levels. This collaborative approach may serve to address and enhance the accuracy of the algorithm in the context of the urgency-level association.

Figure 4 presents a screenshot of the graphical interface displaying the waiting time. In this figure, it can be observed that the patient's name is not shown to avoid violating confidentiality. Instead, a unique code is displayed, which is not the personal numeric code. In this figure, the average waiting time for each patient, expressed in minutes, can also be observed. It should be noted that this waiting time may increase if other urgent cases appear in the emergency room, requiring priority intake.

The algorithm demonstrates that it is possible to use a nationwide application that allows the display of average waiting times and effectively manages crowded situations. This approach aims to minimize contact between patients, reducing the risk of contamination with other illnesses for a patient in the waiting room.

The results of this study cannot be directly compared to previous studies due to the innovative nature of the proposed algorithm. The authors are not aware of any other existing algorithm with the same objectives as the proposed one.

As a future research direction, the authors aim to calculate the accuracy of the ChatGPT service for classifying a large volume of inputs, on the order of thousands. Another proposed development will involve creating a proprietary custom-made classification model to conduct a comparative analysis with the ChatGPT service.

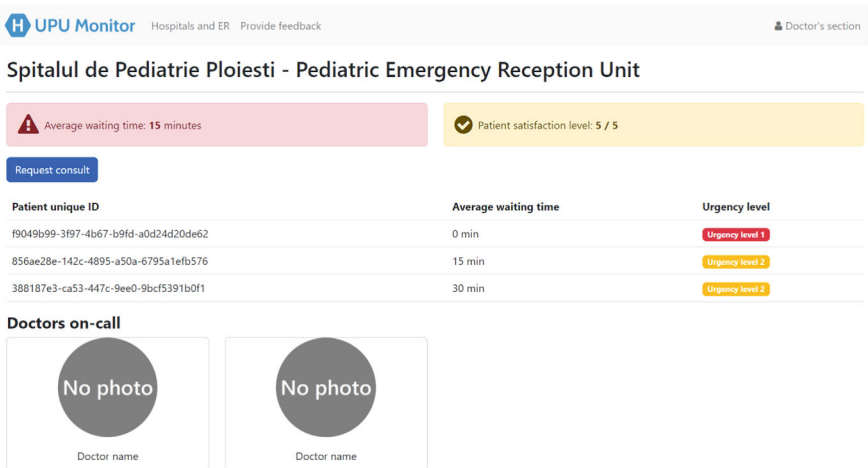


Fig. 4 Screenshot of the graphic interface in the triage application

4 Conclusions

This contribution presented a new algorithm for the emergency healthcare system by computing the average waiting times. Anticipated benefits include prioritizing high-urgency cases, allowing non-life-threatening patients to wait outside the hospital, and reducing emergency room overcrowding. The research explores ChatGPT's potential for classifying urgency levels. The algorithm, integrated into a C# ASP.NET Web application, registers patients' symptoms, categorizes urgency using ChatGPT, and dynamically updates wait times based on real-time conditions. The proposed real-time waiting time formula adapts to changing scenarios, ensuring optimized resource allocation and prompt attention to critical cases. While Internet dependency is a consideration, it allows comprehensive symptom coverage through ChatGPT's extensive database. Integrating the ChatGPT API into the medical triage algorithm represents a noteworthy advancement in in-patient classification process. This study highlights the potential and benefits of artificial intelligence technology in the medical domain, opening new avenues for enhancing national emergency reception services.

References

1. Lu Y, Wu H, Qi S, Cheng K (2023) Artificial intelligence in intensive care medicine: toward a ChatGPT/GPT-4 way? Ann Biomed Eng 51:1898–1903. https://doi.org/10.1007/s10439-023-03234-w
2. Ye Y, You H, Du J (2023) Improved trust in human-robot collaboration with ChatGPT. IEEE Access 11:55748–55754. https://doi.org/10.1109/ACCESS.2023.3282111
3. Qiu J, Li L, Sun J, Peng J, Shi P, Zhang R, Dong Y, Lam K, Lo FP, Xiao B, Yuan W, Wang N, Xu D, Lo B (2023) Large AI models in health informatics: applications, challenges, and the future. IEEE J Biomed Health Inform 27(12):6074–6087. https://doi.org/10.1109/JBHI.2023.3316750
4. Beaulieu-Jones BK, Yuan W, Brat GA, Beam AL, Weber G, Ruffin M, Kohane IS (2021) Machine learning for patient risk stratification: standing on, or looking over, the shoulders of clinicians?. npj Digital Med 4. https://doi.org/10.1038/s41746-021-00426-3
5. Ahsan MM, Luna SA, Siddique Z (2022) Machine-learning-based disease diagnosis: a comprehensive review. Healthcare 10. https://doi.org/10.3390/healthcare10030541
6. Ali H, Qadir J, Alam T, Househ M, Shah Z (2023) ChatGPT and large language models in healthcare: opportunities and risks. In: 2023 IEEE international conference on artificial intelligence, blockchain, and internet of things (AIBThings). IEEE, Mount Pleasant, pp 1–4. USA. https://doi.org/10.1109/AIBThings58340.2023.10291020
7. Shea YF, Lee CMY, Ip WCT, Luk DWA, Wong SSW (2023) Use of GPT-4 to analyze medical records of patients with extensive investigations and delayed diagnosis. JAMA Netw Open 6(8). https://doi.org/10.1001/jamanetworkopen.2023.25000
8. Rodman A, Buckley TA, Manrai AK, Morgan DJ (2023) Artificial intelligence vs clinician performance in estimating probabilities of diagnoses before and after testing. JAMA Netw Open 6(12). https://doi.org/10.1001/jamanetworkopen.2023.47075
9. Lin D, Lin H (2019) Translating artificial intelligence into clinical practice. Ann Transl Med 8(11). https://doi.org/10.21037/atm.2019.11.110
10. Bohr A, Memarzadeh K (2020) The rise of artificial intelligence in healthcare applications artificial intelligence. In: Bohr A, Memarzadeh K (eds.) Healthcare. Academic Press, London, pp 25–60. https://doi.org/10.1016/B978-0-12-818438-7.00002-2

11. OpenAI: Gpt-4 technical report (2023). https://arxiv.org/html/2303.08774v4, last accessed 2024/01/25
12. Singhal K, Tu T, Gottweis J, Sayres R, Wulczyn E, Hou L, Clark K, Pfohl S, Cole-Lewis H, Neal N, Schaekermann M, Wang A, Amin M, Lachgar S, Mansfield P, Prakash S, Green B, Dominowska E, Aguera y Arcas B, Tomasev N, Liu Y, Wong R, Semturs C, Mahdavi SS, Barral J, Webster D, Corrado GS, Matias Y, Azizi S, Karthikesalingam A, Natarajan V (2023) Towards expert-level medical question answering with large language models. arXiv:2305.09617v1 [CL]. https://doi.org/10.48550/arXiv.2305.09617
13. Li Y, Li Z, Zhang K, Dan R, Jiang S, Zhang Y (2023) ChatDoctor a medical chat model fine-tuned on a large language model meta-AI (LLaMA) using medical domain knowledge. Cureus 15(6). https://doi.org/10.7759/cureus.40895
14. Cascella M, Montomoli J, Bellini V, Bignami E (2023) Evaluating the feasibility of ChatGPT in healthcare: an analysis of multiple clinical and research scenarios. J Med Syst 47(33). https://doi.org/10.1007/s10916-023-01925-4
15. Sallam M (2023) ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. Healthcare 11(6). https://doi.org/10.3390/healthcare11060887
16. Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, Ferrero E, Agapow PM, Zietz M, Hoffman MM, Xie W, Rosen GL, Lengerich BJ, Israeli J, Lanchantin J, Woloszynek S, Carpenter AE, Shrikumar A, Xu J, Cofer EM, Lavender CA, Turaga SC, Alexandari AM, Lu Z, Harris DJ, DeCaprio D, Qi Y, Kundaje A, Peng Y, Wiley LK, Segler, MHS, Boca SM, Swamidass SJ, Huang A, Gitter A, Greene CS (2018) Opportunities and obstacles for deep learning in biology and medicine. J Roy Soc Interface 15(141). https://doi.org/10.1098/rsif.2017.0387
17. van der Bij S, Khan N, Ten Veen P, de Bakker DH, Verheij RA (2017) Improving the quality of EHR recording in primary care: a data quality feedback tool. J Am Med Inform Assoc 24(1):81–87. https://doi.org/10.1093/jamia/ocw054
18. Dragan I, Matesoane S, Petrica A, Lungeanu D (2017) A look into emergency department overcrowding: ideas and overview. In: 2017 E-health and bioengineering conference (EHB). IEEE, Sinaia, Romania, pp 113–116. https://doi.org/10.1109/EHB.2017.7995374
19. Giannouchos TV, Ukert B, Wright B (2024) Concordance in medical urgency classification of discharge diagnoses and reasons for visit. JAMA Netw Open 7(1). https://doi.org/10.1001/jamanetworkopen.2023.50522
20. Mondragón N, Istrate D, Wegrzyn-Wolska K, García JC, Sanchez J (2013) Patient classification algorithm at urgency care area of a hospital based on the triage system. J Med Inform Technol 22
21. Sutton RT, Pincock D, Baumgart DC, Fedorak RN, Kroeker KI (2020) An overview of clinical decision support systems: benefits, risks, and strategies for success. npj Digit Med 3. https://doi.org/10.1038/s41746-020-0221-y
22. Wang C, Liu S, Yang H, Guo J, Wu Y, Liu J (2023) Ethical considerations of using ChatGPT in health care. J Med Int Res 25. https://doi.org/10.2196/48009
23. Ayers JW, Poliak A, Dredze M, Leas EC, Zhu Z, Kelley JB, Faix DJ, Goodman AM, Longhurst CA, Hogarth M, Smith DM (2023) Comparing physician and artificial intelligence chatbot responses to patient questions posted to a public social media forum. JAMA Intern Med 183(6):589–596. https://doi.org/10.1001/jamainternmed.2023.1838
24. Zhang P, Kamel Boulos MN (2023) Generative AI in medicine and healthcare: promises opportunities and challenges. Future Int 15(9):286. https://doi.org/10.3390/fi15090286
25. Xue VW, Lei P, Cho WC (2023) The potential impact of ChatGPT in clinical and translational medicine. Clinic Transl Med 13(3). https://doi.org/10.1002/ctm2.1216
26. Timp Online. https://timponline.ro/de-ce-se-asteapta-in-continuare-mult-la-upu-cine-sunt-nemultumitii/. last accessed: 2024/01/26
27. Gilboy N, Tanabe P, Travers DA (2005) The emergency severity index version 4: changes to ESI level 1 and pediatric fever criteria. J Emerg Nurs 31(4):357–362. https://doi.org/10.1016/j.jen.2005.05.011

Enhancing Segmentation Accuracy of Diabetic Retinal Images Using Transform Techniques and deepLabV3+ with Ensemble Classification

S. Mohana Saranya, T. Pranesh, R. Rajasurya, A. H. Sabareesh, Dinesh Komarasamy, and S. Mohanapriya

Abstract This work provides a thorough examination of developments in medical image analysis, with an emphasis on image preprocessing methods to improve raw images in preparation for further tasks including object identification, extraction of features, and classification The study delves into image augmentation, employing techniques like zoom, shear, rotation, and flips, including horizontal and vertical flips, and introduces grid distortion for dataset augmentation. Beyond preprocessing, it explores image transformation operations, encompassing resizing, pixel value normalization, and histogram equalization, coupled with geometric transformations to enhance dataset robustness. The investigation delves into image compression techniques, specifically analyzing the Haar wavelet transform and discrete cosine transform (DCT) for both lossless and lossy compression. Subsequently, this study explores segmentation models (U-Net, DeepLabV3+, and Iterative U-Net) and classification models (Random Forest, EfficientNet B0, and an Ensemble model). The methodology outlines the research approach, including dataset acquisition, image preprocessing, segmentation, and classification, with a detailed performance analysis. The results highlight the superior accuracy of the Iterative U-Net and Ensemble

S. M. Saranya (✉) · T. Pranesh · R. Rajasurya · A. H. Sabareesh · D. Komarasamy · S. Mohanapriya
Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu 638060, India
e-mail: mohanasaranya.cse@kongu.edu

T. Pranesh
e-mail: pranesht.20cse@kongu.edu

R. Rajasurya
e-mail: rajasuryar.20cse@kongu.edu

A. H. Sabareesh
e-mail: sabareeshah.20cse@kongu.edu

D. Komarasamy
e-mail: dinesh.cse@kongu.edu

S. Mohanapriya
e-mail: mohanapriyas.cse@kongu.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_14

model, reaching 92.27% on DRIVE and EyePACS dataset. This study concludes by emphasizing the crucial role of medical image segmentation, suggesting future avenues for exploration, and serving as a foundation for continuous innovation in medical image analysis and diagnosis.

Keywords Medical image segmentation · DRIVE · EyePACS · Haar · DCT · U-Net · DeepLabV3+ · Iterative U-Net · Random forest · EfficientNet B0 · Ensemble

1 Introduction

Significant advancements are being made in the field of medical image analysis, particularly in the vital area of raw picture preprocessing. For photos to be ready for later tasks like feature extraction, object detection, and categorization, they must be improved. This study delves into the intricate world of image preprocessing, particularly focusing on image augmentation techniques to artificially expand datasets, a pivotal step when dealing with limited samples. Techniques, like zoom, shear, rotation, and various preprocessing, and functions, including horizontal and vertical flips, contribute to the augmentation process. Additionally, grid distortion, employing image warping based on a grid structure, plays a crucial role in enhancing dataset diversity and model robustness. Beyond image preprocessing, it explores image transformation operations aimed at improving image quality and feature extraction. These operations encompass resizing, normalization of pixel values, and techniques like histogram equalization to enhance contrast. Geometric transformations, including rotation, flipping, and cropping, contribute to dataset augmentation, ultimately improving model robustness. The integration of image transformation optimizes input data, addressing issues, such as noise and distortions, leading to enhanced performance in downstream algorithms, particularly in tasks like object recognition and classification. The study then delves into two powerful image compression techniques: the Haar wavelet transform and the DCT. The Haar wavelet algorithm proves effective for both lossless and lossy compression, manipulating image matrices to create sparse or nearly sparse matrices, leading to reduced file sizes. The DCT, on the other hand, facilitates the conversion of signals or images from spatial to frequency domains, with applications in compression and signal reconstruction.

Moving forward, the study explores three significant segmentation models: U-Net, DeepLabV3+, and Iterative U-Net. U-Net, designed for semantic segmentation, employs a unique contraction and expansion pathway. DeepLabV3+ stands as a cutting-edge model for semantic image segmentation, featuring encoding and decoding phases to assign semantic labels to each pixel. The Iterative U-Net, a modified version of U-Net, focuses on refining segmentation results through iterative applications, demonstrating advantages in handling complex cases and enhancing network robustness. Three models are demonstrated in the classification section: an Ensemble model that incorporates EfficientNet B0 and B1, Random Forest, and

EfficientNet B0. Random Forest is a well-known Ensemble learning technique that can be utilized for tasks combining both regression and classification. Its effectiveness in these tasks has been well established. EfficientNet B0, designed for image classification, balances efficiency and performance. The Ensemble model leverages two EfficientNet variants using the averaging method, showcasing the power of combining multiple models for improved accuracy.

The methodology section outlines the research approach, starting with dataset acquisition, image preprocessing using Haar wavelet transform and DCT, segmentation using U-Net, DeepLabV3+, and Iterative U-Net, and classification employing Random Forest, EfficientNet B0, and the Ensemble model. The study concludes by emphasizing the critical role of medical image segmentation in facilitating detailed anatomical examinations. It highlights the potential for future exploration, suggesting avenues, such as integrating additional convolutional neural network (CNN) models and combining different transformation techniques. The journey outlined in this study serves as a foundation, encouraging continuous exploration and innovation to advance the landscape of medical image analysis and diagnosis.

2 Related Work

Priyatno [1] presents a novel method for diabetic retinopathy classification, combining HAAR wavelet and backpropagation neural networks. Utilizing 612 images, the study achieves a peak accuracy of 56.25%, yet recognizes limitations in HAAR wavelet's efficacy in identifying retinopathy features due to potential information loss during decomposition. The conclusion emphasizes the imperative for future research to refine preprocessing techniques for enhanced feature extraction. Sambyal et al. [2] introduce an innovative method for detecting diabetic retinopathy by effectively identifying microaneurysms and exudates using a modified U-Net architecture with a ResNet34 encoder. The proposed approach demonstrates exceptional performance on datasets like IDRiD and e-ophtha, surpassing existing methods. Emphasizing the significance of automated segmentation, the study highlights features like transfer learning, advanced upsampling techniques, and cross-connections, contributing to a robust and highly efficient lesion segmentation method. Pratheeba et al. [3] present a unique SVM-based automated technique for the detection of diabetic retinopathy. Marin et al. propose a feature-based and supervised classification method for identifying exudates in retinal images. Dheeba and Singh utilize fuzzy support vector machines to detect hard exudates, while other studies explore various techniques, such as CNN, support vector machine, radial basis function, deep neural network, back propagation neural network, and KNN classifier, for diabetic retinopathy detection and achieving an impressive accuracy of 99.89%. Sarobin et al. [4] presented a comprehensive approach to detect diabetic retinopathy by employing image enhancement techniques and integrating CNNs with transfer learning, specifically DenseNet and ResNet. Their comparative analysis underscores the superior performance of the hybrid CNN with DenseNet, reaching a remarkable 96.22%

accuracy. This demonstrates the effectiveness of advanced learning techniques in accurately predicting diabetic retinopathy stages, marking a significant advancement in the field. Madduri et al. [5] research addresses diabetic retinopathy using advanced deep learning techniques, focusing on segmentation and classification for DR and retinopathy of prematurity. Their notable contribution includes a DCGAN-based data augmentation method, significantly improving classifier performance and achieving outstanding $F1$-score, sensitivity, specificity, accuracy, precision, and ROP classification outcomes. The proposed model stands out as a promising solution for the automated diagnosis of retinal diseases, surpassing conventional methods. David [6] introduces an innovative method for early diabetic retinopathy detection. The study utilizes advanced techniques, such as k-means clustering, clustering ANN algorithm, and Gaussian filtering for optic cup and disk segmentation. The proposed system, implemented with Spartan 3 FPGA hardware and MATLAB programming, demonstrates accurate cup-to-disk ratio (CDR) computation, offering a cost-effective screening solution for Diabetic Retinopathy in diverse healthcare settings. Sikder et al. [7] introduce an innovative method for precise diabetic retinopathy classification. Employing advanced Ensemble learning and feature-rich datasets, the approach emphasizes overcoming dataset challenges and utilizing the XGBoost algorithm. The study underscores the competitive performance of the proposed technique, particularly in feature selection using Genetic Algorithms, concluding with a comprehensive evaluation and future research directions. Deepa et al. [8] introduce a three-step system using InceptionV3 and Xception CNN models to enhance diabetic retinopathy (DR) grading accuracy. Utilizing patch-based and Ensemble methods, beyond state-of-the-art techniques, Multi-stage Patch-based Deep CNN (MPDCNN) achieves an astonishing 96.2% accuracy. The study suggests future research directions, including refining Proliferative DR image classification and exploring advanced neural network architectures. Mondal et al. [9] demonstrate a deep learning method for DR detection using an Ensemble approach, combining modified DenseNet101 and ResNeXt models. Evaluated on APTOS19 and DIARETDB1 datasets, the approach exhibits improved accuracy, particularly in addressing class imbalance. Utilizing CLAHE, GAN-based data augmentation, and thorough comparisons with existing methods, the research provides valuable insights for efficient DR detection and classification. Nair et al. [10] research addresses the challenge of classifying DR severity levels using advanced image processing techniques and K-Means clustering. They applied methods like DCT, Walsh–Hadamard transforms, and the Discrete Sine Transform (DST) on 89 fundus images from the DIARETDB1 database, achieving successful normal image classification and identifying severity level concentrations. The study contributes valuable insights to early DR detection and categorization, with implications for automated diagnosis and treatment advancements.

3 Working Methodology

The methodology outlined for this research commences with the acquisition of a dataset [11, 12] containing diverse medical images, potentially encompassing instances of diabetic retinopathy. To optimize computational efficiency and preserve essential image features, a two-fold image compression strategy is implemented, utilizing the Haar Wavelet Transform and Discrete Cosine Transform. The subsequent extraction of the high-frequency band from the compressed images is crucial, as it retains pivotal details and edges vital for accurate segmentation.

Following the preprocessing steps, segmentation is applied specifically to the high-frequency band. This segmentation process aims to partition the medical images into meaningful segments, focusing on regions associated with diabetic retinopathy. The adoption of three distinct convolutional neural network architectures—namely U-Net, DeepLabV3+, and Iterative U-Net—proves essential for image segmentation. These architectures are chosen for their demonstrated effectiveness in capturing spatial dependencies, ensuring robust and accurate segmentation results.

For the simulation of models, keras and tensorflow have been used for the output of the segmentation process reveals highlighted features, particularly emphasizing essential nerves within images related to diabetic retinopathy. A subsequent evaluation step involves comparing the segmented images produced by the different CNN architectures, with a focus on accuracy assessment. This comparative analysis seeks to elucidate the varying capabilities of the models in capturing the desired structures within the medical images, and EyePACS process flow is shown in Fig. 1.

In Fig. 2, the final stage of the methodology, the segmented images undergo a classification process. Leveraging techniques, such as machine learning, deep learning, and Ensemble learning, this classification step assigns specific labels or categories to segmented regions. This classification not only enhances our understanding of distinct features within the medical images but also facilitates the identification of crucial patterns associated with diabetic retinopathy. The comprehensive approach outlined in this methodology highlights the amalgamation of sophisticated image processing techniques and deep learning methodologies for accurate medical image analysis.

4 Image Transformation

It encompasses a series of operations designed to improve image quality and extract relevant features. These operations involve resizing images to standardized dimensions, normalizing pixel values for consistent scales, and employing techniques, such as histogram equalization to enhance contrast.

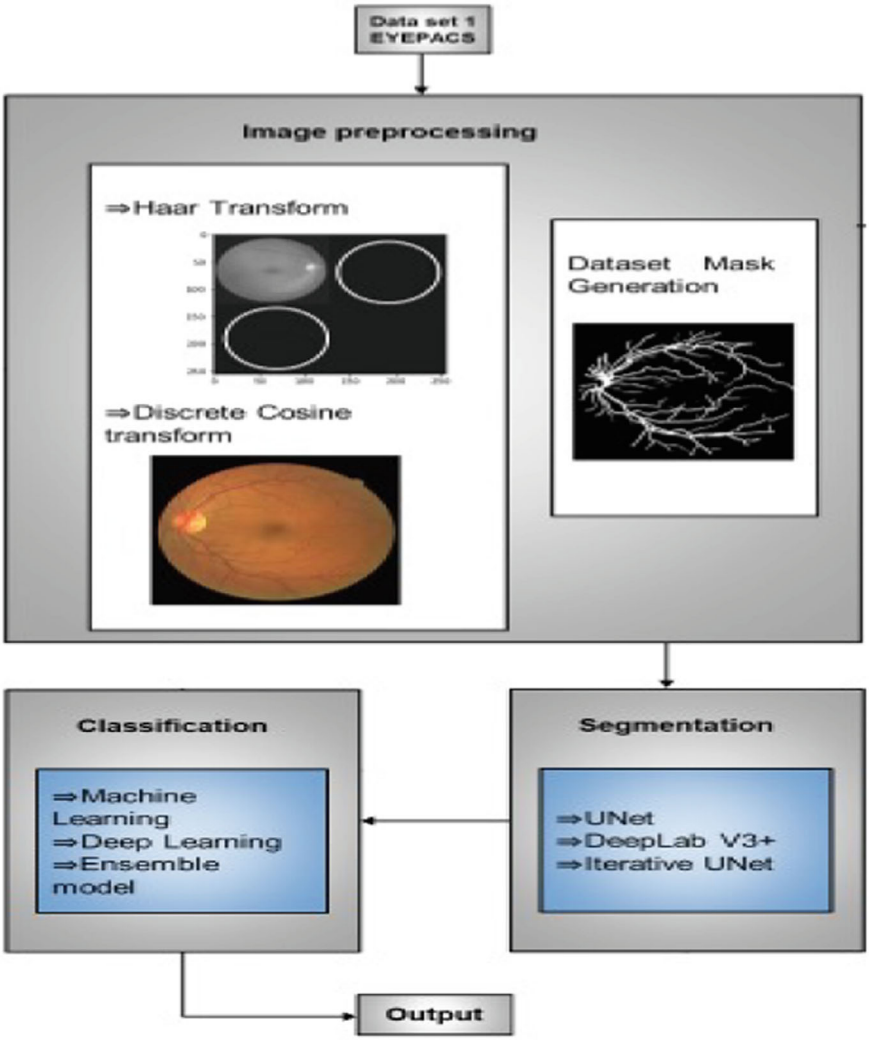


Fig. 1 EyePACS process flow

4.1 Haar Transform

The Haar wavelet algorithm is an effective method for compressing images, both with and without loss of data. This algorithm works by manipulating the image matrix through a process of averaging and differentiation. The outcome is a sparse or almost sparse matrix, meaning that a large number of entries are zero. Sparse matrices offer efficient storage, leading to reduced file sizes. The formula for the one-dimensional Haar transform is,

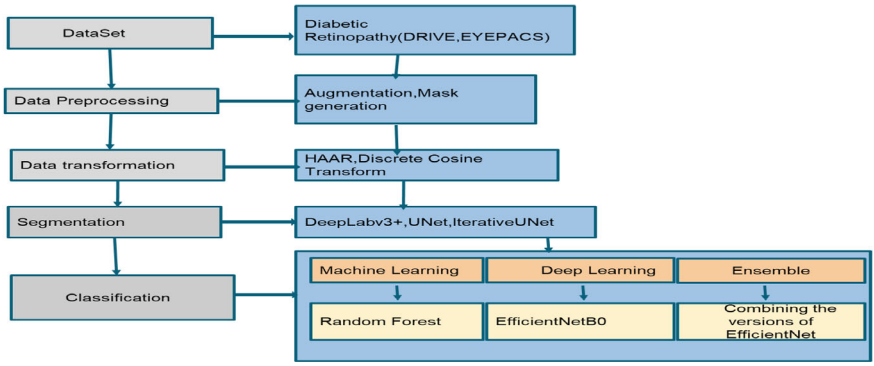


Fig. 2 Methodology

$$H = \left(\frac{1}{\sqrt{v}}\right) * [1\ 1\ 1\ \ldots\ 1;\ 1\ -1\ 0\ \ldots\ 0;\ 0\ 0\ 1\ -1\ \ldots\ 0; \ldots;\ 0\ 0\ \ldots\ 1\ -1] \quad (1)$$

where v is the length of the signal or image, and the matrix represents the Haar matrix.

4.2 Dct

The DCT is a widely used mathematical technique that helps to convert a signal or image from its spatial domain to the frequency domain. This transformation is achieved by representing the signal as the total of different frequency cosine functions. The DCT formula for a one-dimensional signal, x, of length R, is as follows:

$$X_S = 2 * \sum \left(x_q * \cos\left(\left(\frac{\pi}{R}\right) * \left(q + \frac{1}{2}\right) * S \right) \right) \quad (2)$$

for $q = 0 - R - 1$ and $S = 0 - R - 1$. Here, X_S is the S-th DCT coefficient, and cos() represents the cosine function. The inverse DCT formula, which is used to reconstruct the signal from its DCT coefficients, is as follows:

$$x_q = \left(\frac{2}{R}\right) * \sum \left(X_S * \cos\left(\left(\frac{\pi}{R}\right) * \left(q + \frac{1}{2}\right) * S \right) \right) \quad (3)$$

for $S = 0 - N - 1$ and q $= 0 - N - 1$.

5 Segmentation Models

Segmentation models are algorithms used in computer vision to partition an image into meaningful regions.

5.1 U-Net Architecture

The U-Net architecture is specifically designed for semantic segmentation and is characterized by contraction and expansion pathways. The pathway model follows a standard convolutional network architecture. It entails applying two 3×3 successive convolutions (without padding). A rectified linear unit (ReLU) is applied following each convolution. For downsampling, a 2×2 max-pooling procedure with a stride of 2 is applied.

5.2 DeepLabV3+ Architecture

Applying a semantic label, such as "person," "dog," or "cat," to every pixel in an input image is a modern deep learning model for semantic image segmentation. A convolutional neural network (CNN) is employed in the encoding stage to extract pertinent information from the input image. On the other hand, using the data gathered from the encoding phase, the decoding phase reconstructs the output to the correct dimensions.

5.3 Iterative U-Net Architecture

A modified version of the U-Net architecture, known as the Iterative U-Net, was created especially for semantic segmentation in medical images, particularly in low contrast situations, indistinct boundaries, or other complexities posing challenges to segmentation. Rooted in the U-Net framework, a prevalent CNN for image segmentation, the Iterative U-Net adopts a novel approach to refining segmentation results through iterative application of the U-Net architecture on the input image and the current segmentation outcome.

6 Result and Discussion

In the thorough examination of segmentation performance, we meticulously compared the outcomes generated by the U-Net, DeepLabV3+, and Iterative U-Net architectures. This evaluation, detailed in Table 1, includes cases with and without image preprocessing. The tables meticulously lay out the performance metrics, shedding light on the nuances of each architecture's ability to segment medical images. Employing a diverse set of methodologies, we delved into machine learning with Random Forest, deep learning with EfficientNet B0, and a sophisticated Ensemble model combining EfficientNet B0 and EfficientNet B1. The culmination of these efforts is encapsulated in the insightful Fig. 3.

Upon thorough analysis, a compelling inference emerged: the Iterative U-Net architecture, coupled with the Ensemble model for classification, demonstrated unparalleled performance. In Table 2, the dynamic duo achieved an impressive accuracy rate of 92.27%. This finding underscores the effectiveness of leveraging advanced architectures and Ensemble approaches, pointing toward a promising avenue for accurate and robust medical image analysis. The significance of these results lies not only in the achievement of high accuracy but also in the informed selection of the most potent methodologies for segmentation and classification tasks in the realm of medical imaging.

Table 1 Comparing performance of segmentation models for EyePACS dataset

	Without preprocessing		With Haar		With DCT	
Algorithm	Dice coefficient	IOU	Dice coefficient	IOU	Dice coefficient	IOU
U-Net	0.2531	0.2336	0.5295	0.6859	0.5110	0.6643
DeeplabV3+	0.4886	0.3232	0.6714	0.6549	0.6279	0.5041
Iterative U-Net	0.2796	0.4376	0.7196	0.7896	0.5376	0.7469

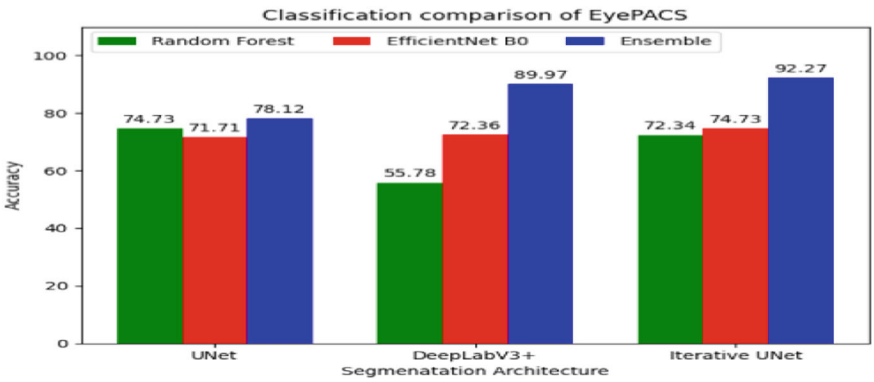


Fig. 3 Accuracy Chart for classification of EyePACS dataset

Table 2 Accuracy in Classification of EyePACS dataset

	Classification Models		
Algorithm	Random forest	Efficient B0	Ensemble
U-Net	74.73	71.71	78.12
DeepLabV3+	55.78	72.36	89.97
Iterative U-Net	72.34	74.73	92.27

In Fig. 4, visual representation of the segmented images is obtained through the application of our segmentation architectures on the EyePACS dataset. This dataset, encompassing a diverse array of eye-related medical images, serves as a real-world testbed for evaluating the efficacy of our segmentation methodologies. Each segmented image in the figure vividly captures the intricate details and delineates specific regions of interest, showcasing the segmentation prowess of our chosen architectures—U-Net, DeepLabV3+, and Iterative U-Net. Figures 5 and 6 are demonstrated, for better understanding of results.

The segmented images provide a comprehensive visual insight into how each architecture interprets and partitions the complex structures within the EyePACS dataset. Not only do these visual representations highlight the distinctive features of

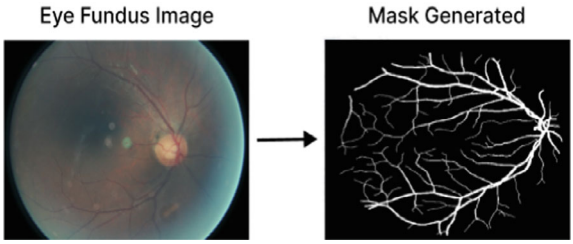


Fig. 4 Segmented output image

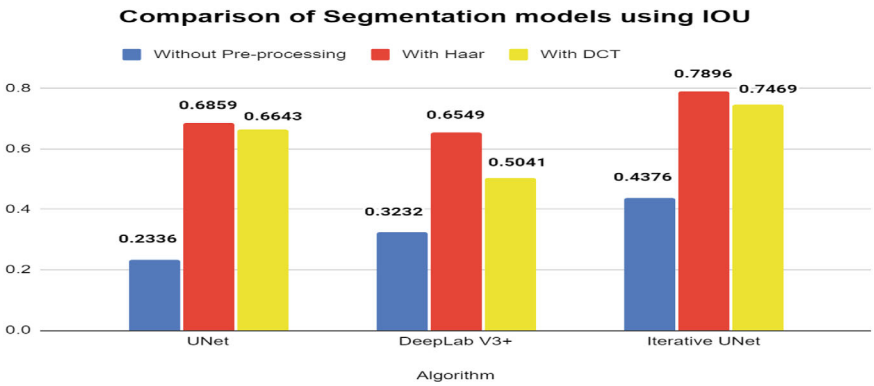


Fig. 5 Comparing segmentation models using IOU for EyePACS dataset

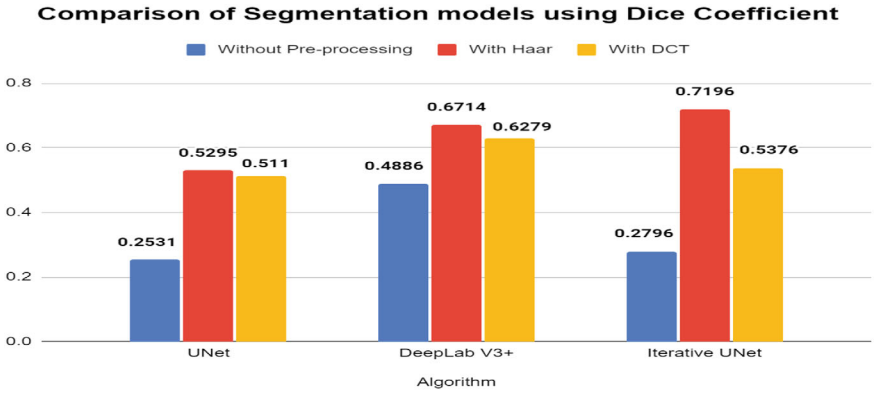


Fig. 6 Comparing segmentation models using Dice Coefficient for EyePACS dataset

diabetic retinopathy and other eye-related conditions, but they also underscore the precision and accuracy achieved by our segmentation methodologies. This visual exploration is integral in complementing the quantitative metrics presented earlier, offering a tangible glimpse into the segmentation capabilities of each architecture. The segmented images serve as a valuable resource for clinicians and researchers, aiding in the qualitative assessment of the segmentation outcomes and providing a deeper understanding of the architectural nuances in capturing relevant anatomical details within the medical images.

The combination of quantitative metrics and visual representations paints a comprehensive picture of the segmentation performance, reinforcing the robustness and applicability of our chosen methodologies in the challenging domain of medical image analysis.

7 Conclusion and Future Work

In conclusion, the pivotal role of medical image segmentation lies in its capacity to isolate and highlight pertinent anatomical regions, facilitating a meticulous examination of crucial data. This study underscores the predominant utilization of CNNs in the domain of image segmentation, with a particular emphasis on their integration with various architectures. Notably, the introduction of image preprocessing through Haar wavelet transform emerges as a key contributor, significantly reducing runtime and enhancing accuracy compared to conventional CNN models.

The present investigation delves into the comparative analysis of U-Net and DeepLabV3+, shedding light on their respective accuracies. However, the trajectory of this research extends beyond its current scope, envisioning several avenues

for future exploration. One promising avenue involves the integration of additional CNN models, augmenting the comparative framework and providing a more comprehensive understanding of their strengths and weaknesses.

Furthermore, the quest for heightened accuracy propels the exploration of advanced transformation techniques. A prospective direction involves combining the output of the Haar wavelet transform with the DCT for a synergistic accuracy enhancement. This nuanced approach anticipates a leap forward in the refinement of segmentation outcomes, capitalizing on the synergy between different transformation methodologies.

References

1. Priyatno AM (2019) The application of Haar wavelet and backpropagation for diabetic retinopathy classification based on eye retina image. Inf Technol 2(1)
2. Sambyal N, Saini P, Syal R, Gupta V (2020) Modified U-Net architecture for semantic segmentation of diabetic retinopathy images. Biocybernetics and Biomedical Engineering 40(3):1094–1109
3. Pratheeba C, Nirmal Singh N (2019) A novel approach for detection of hard exudates using random forest classifier. J Med Syst 43(7):180
4. Sarobin MVR, Panjanathan R, Jasmine SG, Anbarasi LJ (2022) Diabetic retinopathy classification using CNN and hybrid deep convolutional neural networks. Symmetry 14:1932
5. Madduri VK, Rao BS (2023) An approach to classify and segment diabetic retinopathy and retinopathy of prematurity. Trait Signal 40(3):939
6. David D (2020) Retinal image classification system for diagnosis of diabetic retinopathy using morphological edge detection and feature extraction techniques. Artech J Eff Res Eng Technol 1:28–33
7. Sikder N, Masud M, Bairagi AK, Arif ASM, Nahid AA, Alhumyani HA (2021) Severity classification of diabetic retinopathy using an ensemble learning algorithm through analyzing retinal images. Symmetry 13(4):670
8. Deepa V, Sathish Kumar C, Cherian T (2022) Ensemble of multi-stage deep convolutional neural networks for automated grading of diabetic retinopathy using image patches. J King Saud Univ-Comput Inf Sci 34(8):6255–6265
9. Mondal SS, Mandal N, Singh KK, Singh A, Izonin I (2022) Edldr: an ensemble deep learning technique for detection and classification of diabetic retinopathy. Diagnostics 13(1):124
10. Nair M, Mishra DS (2019) Categorization of diabetic retinopathy severity levels of transformed images using clustering approach. Int J Comput Sci Eng 7(1):642–648
11. DRIVE Dataset. https://www.kaggle.com/datasets/andrewmvd/drive-digital-retinal-images-for-vessel-extraction
12. EyePACS Dataset. https://www.kaggle.com/datasets/ascanipek/eyepacs-aptos-messidor-diabetic-retinopathy

Predicting Ocean Currents: A Deep Learning Approach

Ameesha Anil and M. Dhanya

Abstract The current study explores deep learning techniques to investigate ocean current velocity and predictability. We utilize the long short-term memory (LSTM) deep learning network to analyze ocean current data. Here, a time series of ocean currents data at one location obtained from the Indian Space Research Organisation's (ISRO) Meteorological and Oceanographic Satellite Data Archival Centre (MOSDAC) is used. We demonstrate that the LSTM can predict the current speed (V) as well as the two velocity elements, zonal velocity (u) and meridional velocity (v). The performances of three LSTM network variants, viz., vanilla LSTM, stacked LSTM, and bidirectional LSTM, are compared. The evaluation metrics used here are (i) root mean squared error (RMSE), (ii) mean absolute error (MAE), and (iii) mean of prediction (MoP). We also investigate how the number of epochs affects the predictions. Our findings have numerous relevant applications, including weather prediction and predicting tidal energy variance. They have a high potential for use in various practical problems in physical oceanography.

Keywords Deep learning · Long short-term memory (LSTM) · Meridional velocity · Ocean currents · Zonal velocity

1 Introduction

Ocean currents are major driving factors of the earth's climate [1]. The horizontal movement of seawater, named ocean currents, occurs due to wind and is influenced by gravity, water density, and the Coriolis effect [2–4]. The movement of heat, fluctuations in biodiversity, and the earth's climate system are all caused by this

A. Anil · M. Dhanya (✉)
Amrita Vishwa Vidyapeetham, Amritapuri, India
e-mail: dhanyam@am.amrita.edu

A. Anil
e-mail: ameeshaanil@am.students.amrita.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_15

abiotic system. Ocean currents move clockwise in the Northern Hemisphere and anticlockwise in the Southern hemisphere due to the Coriolis effect [5].

Out of the five oceanic regions in the world, the Indian Ocean is the third largest one [6]. Due to the monsoon circulation, the Indian Ocean's surface currents experience semiannual and annual variations. Indian Ocean gyre, a large system of rotating ocean currents, reverses its direction from clockwise in summer to anticlockwise in winter [7–9]. Studies such as [10] provide a comprehensive understanding of ocean currents in the Indian Ocean.

Traditional ocean current prediction methods, such as those based on numerical simulation, have space for improvement to capture the complex dynamics of ocean currents adequately [11]. To overcome these difficulties and improve the precision of predictions of ocean currents, deep learning techniques [12] present intriguing options. Among the various deep learning techniques, the particularly long short-term memory (LSTM) networks hold much promise. Recurrent neural networks (RNNs), such as LSTM networks [13], excel in modeling sequential data by accurately capturing long-term dependencies [14]. They are ideal for modeling ocean currents because they can selectively forget and retain knowledge over long periods [15].

Several researchers have introduced the LSTM architecture and discussed how it differs from conventional RNNs in that it can avoid the vanishing gradient issue. It serves as a starting point for further study on layered LSTM models [16]. Multiple LSTM layers are stacked in a deep stacked LSTM architecture for demand forecasting to capture hierarchical patterns in the time series data. Comparing the findings to conventional LSTM models, the results show better forecasting accuracy [17, 18]. There have been works that introduced the idea of residual learning in time series forecasting; however, they were not specifically geared toward stacked LSTM models. Nonetheless, to enhance data flow and model performance, the concept of residual connections between stacked LSTM layers can be developed from this [19]. Researchers have also looked into stacked residual LSTM networks for multi-horizon time series forecasting. The experiments contrast various stacking techniques and show the advantages of stacking LSTM models for predicting multiple future time steps [20]. A stacked LSTM model with periodicity detection to increase multi-step forecasting accuracy has also been developed, focusing on time series data with periodic patterns [21]. In light of this, stacked LSTM models have become a potent method for time series forecasting, allowing for capturing hierarchical patterns and complicated temporal connections in the data. It has also been investigated how stacked LSTM models differ from conventional LSTM models and other forecasting strategies. Additionally, stacked LSTM models have been expanded to address various problems, including multi-step forecasting, periodic time series, and forecasting perishable product demand.

The intricate connections and patterns in time series data are frequently tricky for conventional forecasting techniques to capture. Bidirectional long short-term memory (Bi-LSTM) networks have become popular because they can simulate forward and backward temporal relationships. Two LSTM layers make up the bidirectional LSTM architecture, one of which processes the input sequence forward and the

other processes it backward. As a result, the network can collect information about the past and the future, leading to more precise forecasts. Bidirectional LSTM was first introduced by Graves (2013) [22], who also showed how good it was for voice recognition tasks. Since then, time series forecasting has primarily embraced the LSTM's bidirectional character [23]. Researchers have also suggested modifications to boost the bidirectional LSTM architecture's predicting performance further. For instance, a time series forecasting attention method for the bidirectional LSTM network was introduced. The attention mechanism aided the model's emphasis on pertinent time steps, making the forecasts easier to understand [24]. As a result, the use of bidirectional LSTM networks for time series forecasting has yielded encouraging results in several fields.

We would be examining the speeds of ocean surface currents. They can be categorized as periodic or mean currents from the perspective of time series analysis. Occasional currents are those that cycle through shifting speed and direction changes at regular intervals. However, mean ocean circulations contain mean regions that fluctuate little or not at all through time and place. The two components of surface current velocities are u and v, where u is the velocity of the surface current toward the east and v is the velocity toward the north [13]. u and v are known as zonal velocity and meridional velocity, respectively.

Here, we utilize deep learning techniques in order to examine and forecast the velocities of the ocean surface currents. To be more precise, we use global ocean current data received from the Meteorological and Oceanographic Satellite Data Archival Centre (MOSDAC) [25] and analyze it at a latitude of 11.25° N and longitude of 71° E from 2016 to 2022. We apply various LSTM deep learning network variants, such as vanilla LSTM, stacked LSTM, and bidirectional LSTM (Bi-LSTM), to the current speeds and the velocity components. Using root mean squared error, mean absolute error, and mean of prediction, we examine and quantify the performance of each of the three models. We also investigate how the number of epochs impacts the forecasts.

2 Methodology

2.1 Long Short-Term Memory

LSTM is an example of a recurrent neural network (RNN) that can learn long-term relationships, especially in sequence prediction tasks. Hochreiter and Schmidhuber [26] formulated it to eliminate the "vanishing gradient" problem in the general RNN. Four neural networks and several memory-building pieces known as cells make up the chain structure of the LSTM. The cells are responsible for information retention, and the gates control memory manipulation. Input and output gates manage the data added to and erased from the memory cell. Furthermore, the output cell regulates the data sent from the memory to the output. This enables the LSTM networks to learn

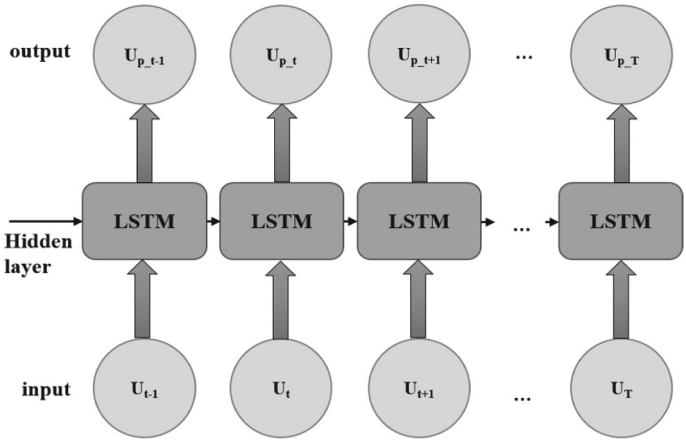


Fig. 1 Vanilla LSTM

long-term dependencies by retaining or discarding selected information as it passes through the network [27].

In this work, we will be implementing and comparing the predictions done by three LSTM network variants,

(i) Vanilla LSTM: A vanilla LSTM has one hidden layer of LSTM unit and one output layer. It works well for modeling sequential data with short-term dependencies. Complex long-term dependencies could be challenging to capture in the data. Figure 1 shows a simple schematic of our vanilla LSTM architecture.
(ii) Stacked LSTM: Figure 2 shows a simple schematic of our stacked LSTM architecture. Here, multiple LSTM layers are stacked one on top of another. Each layer in the stack processes the input sequence before being passed on to the layer next to it. Due to its more intricate design, stacked LSTM can capture more intricate temporal correlations in the data. If the model is too large for the dataset, it may be subject to overfitting and need more processing resources.
(iii) Bidirectional LSTM: By processing the input sequence both forward and backward, the Bi-LSTM improves upon the conventional LSTM. Here, two LSTM layers are used, with one handling the input processing in the forward direction and the other in the backward direction. Bi-LSTM can be helpful for jobs that call for a thorough knowledge of the input sequence because it gathers information from both past and future context. It is frequently used for operations like named entity recognition, translations, and speech recognition. Figure 3 shows a simple schematic of our Bi-LSTM architecture.

We made use of data on the world's ocean currents from 2016 to 2022 obtained from MOSDAC [25]. We examined the data files, read them, and extracted the ocean current velocities in the Indian Ocean at latitude 11.25° N and longitude 71° E and made a data frame in Python. This location is selected as it is in the Arabian Sea and

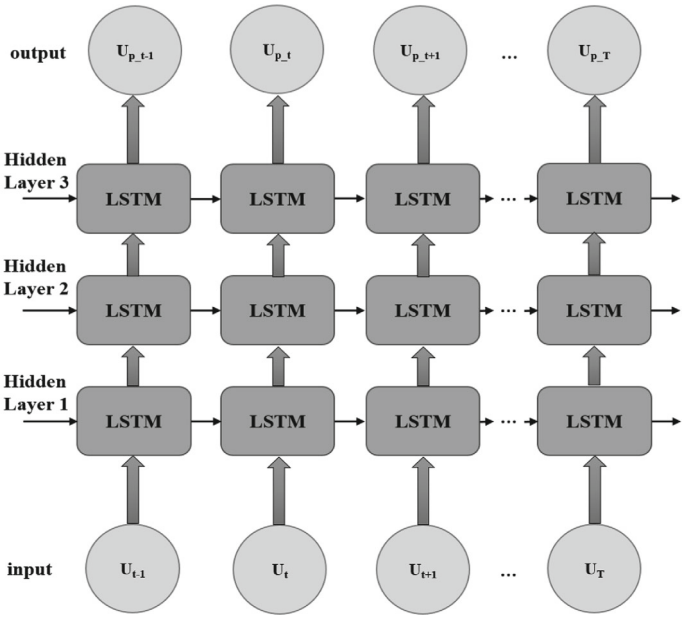


Fig. 2 Stacked LSTM

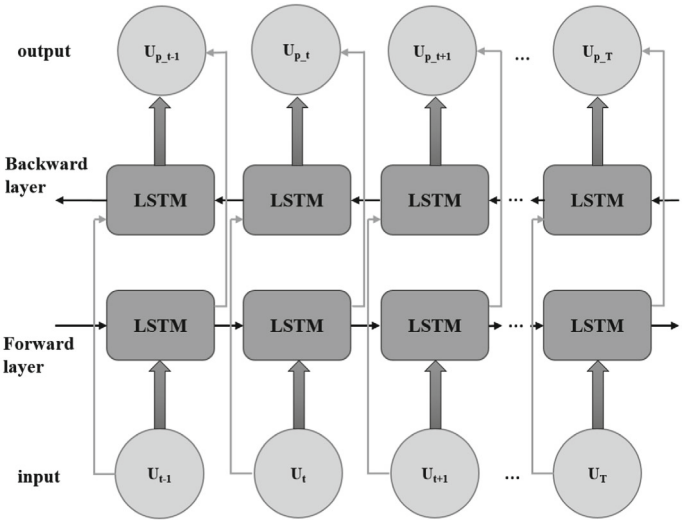


Fig. 3 Bidirectional LSTM

is not very close to the coast, so coastal influences are minimal. A time series dataset is created once the data frame has been constructed.

The time series dataset is then scaled, normalized, and split into training and testing samples. The training test has data from 2016 to 2020, and the testing set is from 2021 to 2022. A vanilla LSTM, stacked LSTM, and Bi-LSTM models are built. The developed models are then fitted with the datasets. The metrics root mean squared error, mean absolute error, and mean of prediction are used to assess the predictions.

3 Results and Analysis

We have analyzed the ocean current speeds, zonal velocity, and meridional velocity. The three models are compared by evaluating their performances using several evaluation metrics. The effect of the number of epochs on the performance of the models is also evaluated.

A plot of the ocean currents' speed (V), zonal velocity (u), and meridional velocity (v) from January 1, 2016, to December 31, 2022, is created as shown in Fig. 4a, b, c, respectively.

The speed of surface currents at the study location is provided in Fig. 4a. It indicates that the surface current speed undergoes interannual as well as intra-seasonal oscillations. We can observe the time periods where the ocean surface currents are the least and most intense. The current speeds are observed to be the lowest in April, November, January, June, July, December, and October of each of the years 2016, 2017, 2018, 2019, 2020, 2021, and 2022, with June 2019 recording the lowest current speed ever during the 7-year period. Meanwhile, current speeds are the highest in July 2016, December 2017, April 2018, December 2019, February 2020, February 2021, and May 2022 for each year, with the most intense currents being in March 2020.

3.1 Effect of Epochs on Model Performance

The effect of varying the number of epochs on the model performance is analyzed. This is done on the current speed data, and root mean squared error is chosen as the evaluation metric to study its effect. The observations are plotted graphically and are shown in Fig. 5.

As the number of epochs rose, we can observe that the performance increased. However, it is found that the model began to overfit as the number of epochs exceeded 50 considerably. Accordingly, at 50, we stopped.

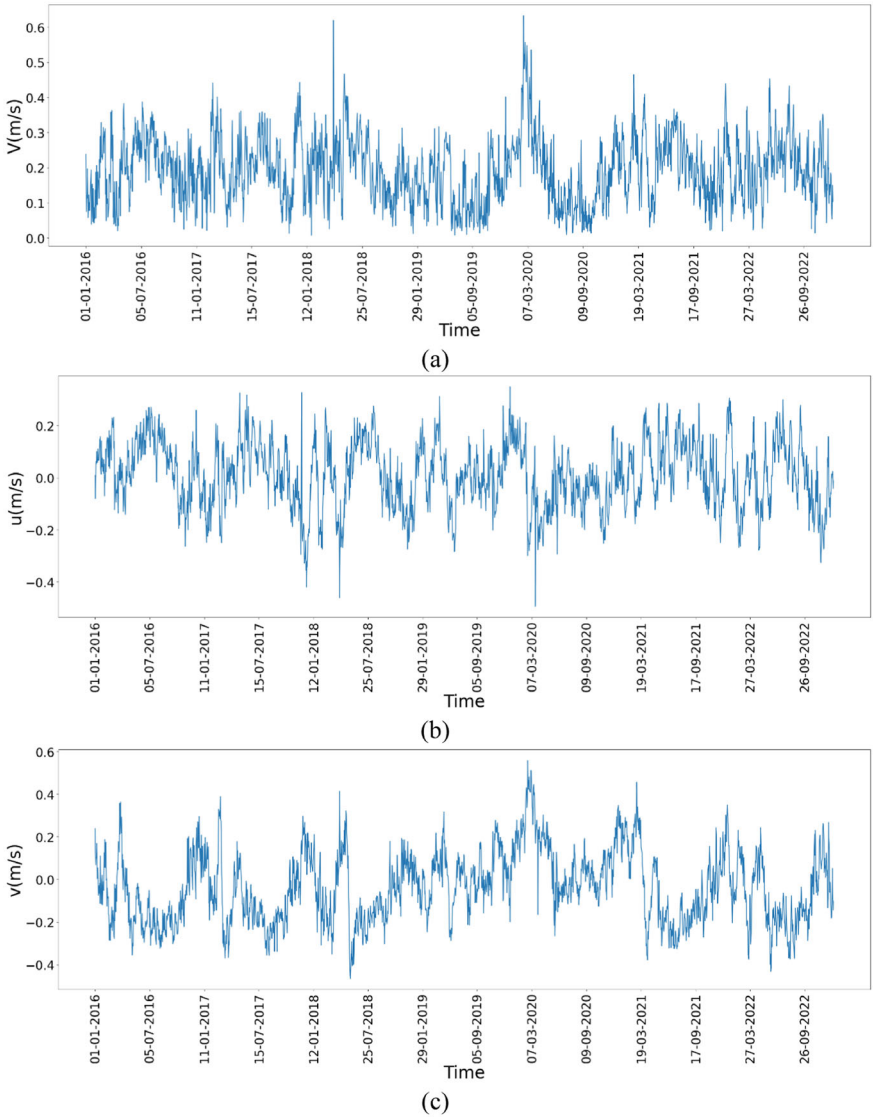


Fig. 4 Plot of **a** current speed, **b** zonal velocity, and **c** meridional velocity

3.2 Forecast of Current Speed

Vanilla LSTM, stacked LSTM, and bidirectional LSTM is implemented on the currents' speed to obtain the following predictions as shown in Fig. 6.

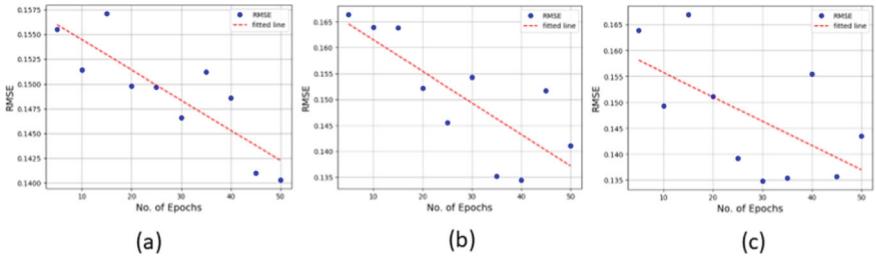


Fig. 5 Effect of number of epochs on performance of **a** vanilla LSTM, **b** stacked LSTM, and **c** bidirectional LSTM

As demonstrated in Fig. 6a, the vanilla LSTM captures the current speeds' underlying pattern fairly well. It does not, however, accurately depict the peaks and magnitudes. Figure 6b shows that, in comparison with the first vanilla LSTM run, the stacked LSTM's performance in capturing the underlying pattern of the current speeds is noticeably worse. Its capacity to capture peaks and magnitude, however, is almost as equal to that of vanilla LSTM. We are able to deduce the bidirectional LSTM's performance from Fig. 6c. We may observe that, in comparison with vanilla LSTM and stacked LSTM, its capacity to capture the underlying pattern of current speeds, as well as zonal and meridional velocities, was noticeably less. On the contrary, its performance in capturing peaks and magnitude is comparable to the first two LSTM networks.

Our predictions were validated by implementing an eye test and predicting 30 days of future data. The plot, including the future predicted data, is shown in Fig. 7.

We can determine whether our estimates are still accurate in the near future by looking at the eye test plot. Future projections of current speeds for vanilla LSTM and stacked LSTM are similar, as illustrated in Fig. 7. But they are significantly poorer for bidirectional LSTM. This aligns with the evaluation metrics that are gathered.

By following the same process again, predictions for the velocity elements, zonal and meridional velocities using the three LSTM networks are also achieved, but the resulting graphs are not shown here, for brevity.

The three LSTM models considered in this study are evaluated using quantities, such as root mean squared error, mean absolute error, and the mean of prediction. The performance evaluations of the three LSTM models for the current speed, zonal velocity, and meridional velocity are recorded in Table 1.

The vanilla LSTM model has the lowest error and highest average prediction value overall among the three models, as demonstrated by the RMSE, MAE, and MoP observations and assessment metrics. This shows that the vanilla LSTM model is the most successful with accurate forecasts for this specific task and dataset. All three models did a similar job of capturing the magnitude peaks; the meridional velocity estimates being the best. In general, vanilla LSTM performs better for real-time applications or circumstances with resource limitations since it strikes a balance between computational efficiency and performance.

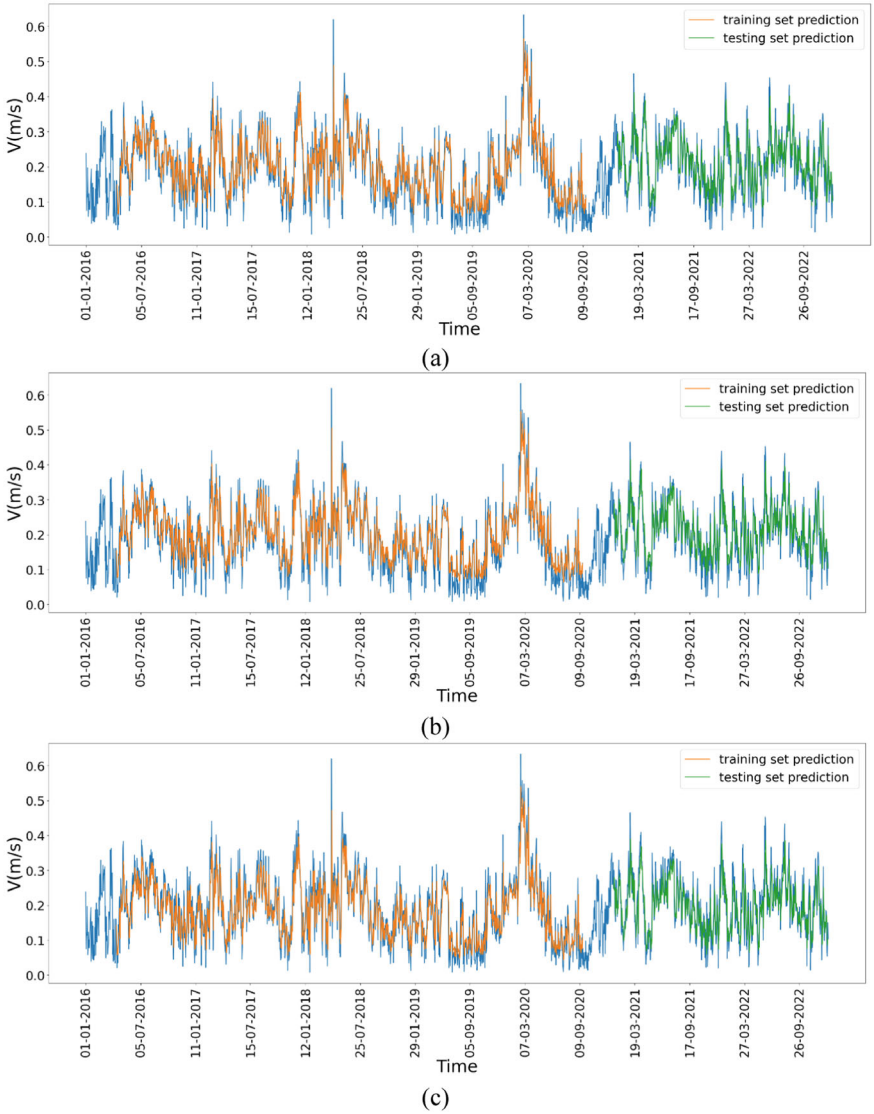


Fig. 6 Current speed predictions by **a** vanilla LSTM, **b** stacked LSTM, and **c** bidirectional LSTM

4 Conclusions

In this work, we examined the predictability of ocean current speed, zonal velocity, and meridional velocity using deep learning. To examine the prediction performance of the long short-term memory deep learning network, we focused on its performance on the ocean current speed time series. The dataset utilized in this study is created

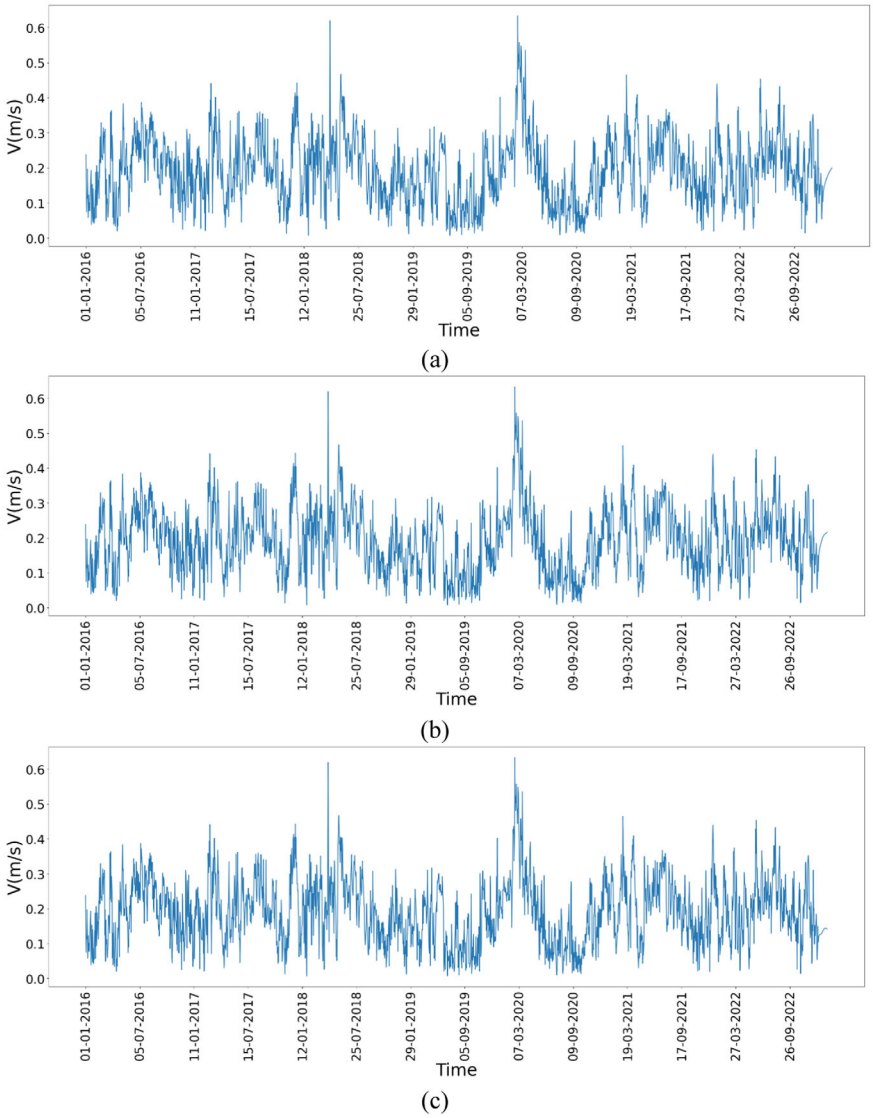


Fig. 7 Eye test plot for **a** vanilla LSTM, **b** stacked LSTM, and **c** bidirectional LSTM

using worldwide ocean current data obtained by MOSDAC between January 1, 2016, and December 31, 2022. We evaluated the data and obtained the current speeds at 11.25° N and 71° E, which is in the Indian Ocean, more precisely in the Arabian Sea. For this work, we compared the performance of three different LSTM variants: vanilla LSTM, stacked LSTM, and bidirectional LSTM. As evaluation metrics for the LSTM models, we employed root mean squared error, mean absolute error, and

Table 1 Evaluation metrics of three LSTM networks

	Training set			Testing set		
	V	*u*	*v*	*V*	*u*	*v*
Vanilla LSTM						
RMSE	0.1381	0.2299	0.1718	0.1403	0.2365	0.1629
MAE	0.1108	0.2133	0.1513	0.1147	0.2199	0.1407
MoP	0.1993	0.3812	0.2711	0.2124	0.3919	0.2569
Stacked LSTM						
RMSE	0.1392	0.2448	0.1611	0.1411	0.2518	0.1523
MAE	0.1117	0.2288	0.1394	0.1155	0.2359	0.1293
MoP	0.1992	0.3654	0.2837	0.2119	0.3758	0.2695
Bi-LSTM						
RMSE	0.1409	0.2432	0.1669	0.1435	0.2505	0.1584
MAE	0.1205	0.2269	0.1463	0.1260	0.2344	0.1361
MoP	0.1858	0.3675	0.2764	0.1983	0.3774	0.2619

the mean of prediction. We also tested our model by running an eye test to determine whether our forecasts looked good in the future. Further, we looked at how the number of epochs affects the forecasts. It is discovered that the models begin to overfit after 50, so we stopped there. From the analysis performed, it can be inferred that among the three LSTM networks investigated, the most basic network, i.e., the vanilla LSTM network, performed the best in capturing the data's underlying pattern. The evaluation metrics obtained also corroborated this. The capacity of the three LSTM models to capture magnitude peaks, on the other hand, remained comparable. Finally, based on the evaluation criteria and overall performance, we can conclude that vanilla LSTM outperformed stacked LSTM and bidirectional LSTM in this study. Additionally, the vanilla LSTM model is also a better option for many real-world applications since it provides an overall balance between computational efficiency and performance. The present work indicates the usefulness of deep learning methods, such as LSTM in forecasting ocean current observations. The current study, however, is limited since it is based on a single location's measurement of ocean currents and does not consider the entire ocean current pattern in the Indian Ocean region.

References

1. Reid PC et al. (2009) Chapter 1 impacts of the oceans on climate change. Adv Mar Biol 56:1–150
2. Tomczak M, Godfrey JS (2003) Regional oceanography: an introduction, 2nd edn. Chapter 1–3, pp 1–38
3. Williams JC, Restrepo JM (1999) The wave-driven ocean circulation. J Phys Ocean 29:2523–2540

4. Matano RP, Palma ED (2008) On the upwelling of downwelling currents. J Phys Ocean 38(11): 2482–2500
5. Stevens ANP (2012) Factors affecting global climate. Nat Educ Knowl 3(10):18
6. Mukhopadhyay R, Loveson VJ, Iyer SD, Sudarsan PK (2021) Blue economy of the Indian ocean: resource economics, strategic vision, and ethical governance
7. Centurioni LR et al. (2017) Northern Arabian Sea circulation-autonomous research (NASCar). A research initiative based on autonomous sensors. Oceanography 30(2):74–87
8. Schott FA, McCreary JP (2001) The monsoon circulation of the Indian Ocean. Prog Oceanogr 51(1):1–123
9. Trott CB, Subrahmanyam B, Murty VSN (2017) Variability of the Somali Current and eddies during the southwest monsoon regimes. Dyn Atmos Ocean 79:43–55
10. Shenoi SSC, Saji PK, Almeida AM (1999) Near-surface circulation and kinetic energy in the tropical Indian Ocean derived from Lagrangian drifters. J Mar Res 57:885–907
11. Wang R (2021) Physics guided deep learning for dynamical systems: a survey. Comput Res Repos
12. Anilkumar A, Yadukrishnan V, Nimal Madhu M, Hareesh V, Premjith B (2023) Deep learning-based time series forecasting for CO_2 emission. In: Vasant P et al. Intelligent computing and optimization. ICO 2023. Lecture notes in networks and systems, vol 729. Springer, Cham
13. Sreedhar R, Varshney A, Dhanya M (2022) Sugarcane crop classification using time series analysis of optical and SAR sentinel images: a deep learning approach. Remote Sens Lett 13(8):812–821
14. Lipton ZC (2015) A critical review of recurrent neural networks for sequence learning. Comput Res Repos
15. Bayindir C (2023) Predicting the ocean currents using deep learning. TWMS J App Eng Math 13(1):373–385
16. Gers FA, Schmidhuber J, Cummins F (1999) Continual prediction using LSTM with forget gates. In: Marinaro M, Tagliaferri R (eds) Neural nets WIRN vietri-99, perspectives in neural computing, Springer, pp 133–138
17. Sagheer A, Kotb M (2019) Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate timeseries forecasting problem. Sci Rep
18. Sowmya C, Kumar AG, Kumar SS (2021) Stacked LSTM recurrent neural network: a deep learning approach for short term wind speed forecasting. In: International conference on intelligent technologies (CONIT), pp 1–7
19. Shen Z, Zhang Y, Lu J, Xu J, Xiao G (2019) A novel time series forecasting model with deep learning. Neurocomputing
20. Putz D, Gumhalter M, Auer H (2021) A novel approach to multi-horizon wind power forecasting based on deep neural architecture. Renew Energy 178:494–505
21. Suradhaniwar S, Kar S, Durbha SS, Jagarlapudi A (2021) Time series forecasting of univariate agrometeorological data: a comparative performance evaluation via one-step and multi-step ahead forecasting strategies. Sensors 21(7)
22. Graves A, Jaitly N, Mohamed AR (2013) Hybrid speech recognition with Deep Bidirectional LSTM. In: IEEE workshop on automatic speech recognition and understanding, pp 273–278
23. Bhattacharjee B, Kumar R, Senthilkumar A (2022) Unidirectional and bidirectional LSTM models for edge weight predictions in dynamic cross-market equity networks. Int Rev Financ Anal 84
24. He YL, Chen L, Gao Y, Ma JH, Xu Y, Zhu QX (2022) Novel double-layer bidirectional LSTM network with improved attention mechanism for predicting energy consumption. ISA Trans 127:350–360
25. MOSDAC (http://www.mosdac.gov.in), Sikhakolli R, Sharma R, Kumar R, Gohil BS, Sarkar A, Prasad KVSR, Basu S (2013) Improved determination of Indian Ocean surface currents using satellite data. Rem Sens Lett 4:335–343
26. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
27. Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput 31(7):1235–1270

Geotagging in Livestock Farming: AI-Based Technologies, Applications, and Challenges

Shuchi Verma, Neelotpal Sharma, and Gaurav Trivedi

Abstract The geotagging technology has emerged as a transformational tool for livestock management, especially in India, where agricultural technology is changing rapidly. The usage of geotagging technology that has been developed recently may bring changes in all agricultural activities, in livestock management, and in animal health monitoring system. The animal's size, weight, and growth rate are the significant factors in determining a country's economy in India. The applications of artificial intelligence (AI) merged with geotagging technology could be considered for discovering the disease surveillance, illegal trade in live animals, and control over livestock movement and overgrazing of pasture. The article discusses the use of geotagging in different animals like beef, pig, sheep, and buffalo industries in India through which cattle, pigs, sheep, and buffaloes can be tracked physically. The aim of this paper is to provide an overview of the potential uses, challenges, and advantages of combining geotagging with geographic information systems (GIS) for livestock management. In addition, this study focuses on the geotagging techniques operated through satellites, for instance, to sensor networks, machine learning and GPS, RFID, Bluetooth, and Wi-Fi items. With wide-range technologies, such as animal behavior sensors, health monitors, and environment data gathering ones like these can help the farmers to make accurate decisions for the productivity and manage resources in a better way. Nevertheless, many obstacles must be solved before geotagging could be very common. It is therefore clear that GPS-based information can facilitate hassle-free management of grazing lands and improve efficiency in livestock production across India as well as in the rest of the world.

Keywords Geotagging technology · Livestock management · Agricultural landscape · GIS integration · Challenges in geotagging

S. Verma
Ramjas College, Delhi University, Delhi, India

N. Sharma (✉) · G. Trivedi
IIT Guwahati, Guwahati, India
e-mail: neelotpal2021@rnd.iitg.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_16

1 Introduction

The majority of livestock production is conducted without supervision, which poses challenges in maintaining proper animal welfare standards. Furthermore, the regions where animals graze typically have severe orography, limited communications infrastructure, and challenging to access. For these reasons, it is imperative to provide a low-power livestock localization and monitoring system. This system is not only necessary for environmentally friendly farming but also, because of its controlled supervision, for the preservation of native breeds and meats. The application of geotagging in agriculture field has been documented in North America, New Zealand, Europe, South Africa, and Asia; therefore, its applicability is not restricted to India. On the other hand, worldwide livestock producers may reap many benefits by keeping an eye on wild animal migrations for conservation purposes and by managing highland pastures more effectively. Many technologies have gained popularity one being geotagging which is availed on geographical information systems (GIS) and livestock management. Geotagging is the process of geocoding the data of every individual by way of pinpointing their exact geographical coordinates, fast-track monitoring while providing traceability, and making the data readily available enabling data-driven decision-making. In India, where agriculture and livestock are the most important sectors of the economy, the application of geotagging technology will be a revolutionary approach. Sensors and tags are fitted on the animals' ears can be GPS-activated and can either send automatic text messages or transmit location information to a centralized database [1]. This new type of annotation can lead to a real transformation of the whole farm production process as growers and ranchers apply it successfully in managing their herds. The use of geotagging in the context of agriculture is what can help the farmers to restock in the best possible way, which is one of the most important gains of geotagging for farmers. Herders do this by keeping track of where their animals are and how they are moving physically and even seeing whether there is enough food and water from an overhead satellite. This can increase the longevity and reproducibility of the herd [1]. So, not just because of identification, we may also use geotagging to track and locate the cattle in the area where the disease epidemic is spread. Farmers will quickly find out which animals are sick and isolate them so that the disease will not spread to the remaining herd due to monitoring the animals' movements. It can increase the durability of the herd and reduce the financial losses occasioned as a result of communicable infections.

Additionally, geotagging cattle can greatly lower the number of cattle stolen or lost. The "lifting livestock" issue is virtually across the globe and the traditional methods in finding the missing animals are quickly turning out to be time-consuming and inefficient. By enabling the approach of traceability using geotags, farmers will be able to track down animals more quickly and use the appropriate measures to take them back home.

Nevertheless, expecting livestock producers to make the investment in geotagging technology for livestock management means considerable financial, material, and technological tenaciousness has to be spent. The farmers and ranchers need to be

equipped with the technology and complete with the technical support. To ensure that the data acquired is accurate and useable, strong data management systems are required.

The aim of this paper is to provide an overview of the potential uses, challenges, and advantages of combining geotagging with geographic information systems (GIS) for livestock management. Also, this study focuses on the geotagging techniques which run from satellites, for instance, to sensor networks, machine learning and GPS, RFID, BLE (Bluetooth), and Wi-Fi items. With wide-range technologies, such as animal behavior sensors, health monitors, and environment data gathering ones like these can help the farmers to make accurate decisions for the productivity and manage resources in a better way.

2 The Scenario of Geotagging of Livestock in India

With increasing the global livestock population mostly in developing countries like India, where in a span of 100 years there is an increase of livestock population from 1.5 billion to 8 billion [2], as shown in Fig. 1. This arises many hurdles of which monitoring, tracking infectious diseases, and livestock theft are the major issues faced by the livestock breeders. Integrating the geotagging of livestock has been one of the recent main attractions in India where it is useful in improving livestock management practices, monitoring, production, and also in supporting numerous policy and program activities.

In the Indian livestock sector, geotagging technology is a step forward that opens up a wide range of data to understand the movement pattern, behavior, and health of animals and allows for tracking disease outbreaks and to access trade through trade.

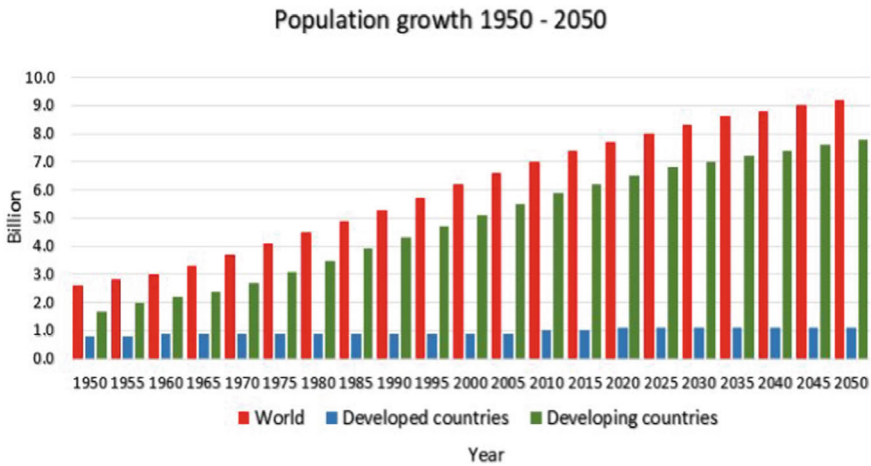


Fig. 1 Global livestock productivity [2]

Regarding the production system, geotagging of cattle has been successfully used in both dairy and beef systems, as well as in different types of husbandries of cattle, sheep, and other hoofed animals. This may involve, the tagging of individual animals/groups of animals with GPS tracking facilitates like collars, ear tags, and/or leg bands, which provide location-based data and can be linked to a centralized database or mapping system for analysis and visualization. The government of India, research institutes, and other stakeholders have been unanimous in facilitating geotagging in agriculture to achieve efficient livestock management methods, have high animal welfare, boost the production of positive economics, and ensure the quality and safety of livestock products. Besides, it has also been used in state programs and rural community initiatives in the fields of animal health, breed development, insurance, market integration, etc. However, the geotagging of livestock in India, like any other technology, has drawbacks, including the requirement for sufficient infrastructure, technical expertise, data administration, and privacy concerns. Although, it is seen as a promising tool that can help in introducing more efficient and helpful farming methods for the livestock sector across the country, where farmers will be able to make profits in livestock systems as well.

3 Adaptability of Various Animals for Geotagging

Geotagging is not specific to India only; it is a common approach employed in the agricultural engineering field where it is used in the rearing of livestock and managing animal husbandry internationally. In North America, geotagging is one of the best-known and frequently used tools, mainly offered to producers of beef and dairy. Examples from these locations include Texas and Montana where tracking of grazing patterns and movements has been revolutionized through GPS and GIS technology, and now optimizing herd grazing management strategies have become a part of beef production. Location-based tracking can now be applied to cows in dairy production, to track and make an account of their feeding and milking behaviors, especially in Wisconsin and California. In Europe, the use of geotags for cattle can be seen in some European countries, among the United Kingdom, France, and Germany. United Kingdom targets this by completing their highland region with GPS and GIS technology in tracking the movements of the sheep. Hence for better management and to prevent overgrazing. An example of geotagging is its utilization in France when it comes to bovine tuberculosis and brucellosis surveillance. Geotagging is a location-based surveillance system that is highly efficient. Geotagging is utilized for cattle-based systems in Australia and New Zealand. The GPS and GIS technology are widely used in Australia to track the movement of livestock in massive grazing areas where farmers often keep their herds, monitor grazing habits by the livestock, and work out the best herd management practices. Geotagging has been used in dairy and beef sectors in New Zealand to monitor individual animals' movements within their systems and also be used to manage traceability and certification programs that need export markets [3]. Kenyan pastoralists employed GPS and GIS technologies

Fig. 2 Geotagging of livestock in Uttar Pradesh. *Source* ANI News UP

to track their movements and those of their animals to guide grazing patterns that aid the reduction of range disputes as well as improve sales of goods and services. Geotagging has been used in South Africa to gather elephant and rhinoceros tracking information jointly with their natural behavior objectives. The main cause of this is to help maintain population stability by avoiding poaching incidences of these wild species [4]. Similarly, some countries in Asia are using geographic positioning system (GPS) for cattle rearing. In India, like for example, to monitor cattle and buffalo's movement in large open grazing lands, to check their health, and to help implement disease control programs, remote monitoring techniques such as GPS and GIS technologies have been used. Yak herders are following the technology of geotagging in China which is helping them to track around the large-scale landscapes over the altitude to provide the effectiveness of grazing management and avoid excessive overgrazing.

3.1 Geotagging of Cows in Uttar Pradesh

In 2018, the government of Uttar Pradesh launched an association known as geotagging of over 5 million cattle that were throughout the entire state (Fig. 2). The project used a particular technology, i.e., geotagging to keep track of the cow's locations, and state of health and reduce livestock theft. Cows acquired the ability to be geotagged by attaching RFID tags to their ears [5].

3.2 Geotagging of Pigs in Kerala

In 2020, this initiative, spearheaded by the Kerala Veterinary and Animal Sciences University, geotagged the pigs of a downward state. The project geolocation system and immunization were used to identify the location of the pigs. Pigs were geotagged to monitor their movements by creating a GPS circuit around their necks [6].

3.3 Geotagging of Sheep and Goats in Telangana

The Telangana government decided to start a new program in 2019 to geotag 4 million sheep and goats by using the technological medium present in the state. The project utilizes the GPS tracking system to observe the animals' movement, to keep up with their health, and to detect stock theft. They have tagged Tibetan sheep and goats with radio frequency identification (RFID) tags.

3.4 Geotagging of Buffalo in Haryana

The Haryana government started a program to geotag over 2 million buffaloes in the year 2017. The ordeal aimed at the employ of geotagging technology to track the movements of buffalos and thereby reduce the level of pilferage in the livestock industry. The buffaloes were geotagged by inserting RFID tags on their ears. The RFID tags carried a special code to transmit the signal.

4 Exploring Geotagging Strategies for Livestock Monitoring and Management

Livestock geotagging is the digitizing of the movements, health, and behavioral changes of animals through technological instruments that offer a significant amount of data for this purpose. There are various techniques and methods which are involved in the geotagging livestock sector, and all of them possess their own set of benefits and downsides. The system architecture design [7] is shown in Fig. 3.

GPS technology deploys satellites that serve as a triangulation tool to spot the area where the tagged animal is. All animals are fixed with GPS tracking devices, and the geolocation is offered in real time or at a scheduled interval. GPS tags provide very precise tracking within a wide scale range and are therefore excellent for tracking livestock, such as cattle and sheep, which are grazing freely [8]. The RFID tags can be read electronically, and then, the data they have is sent back wirelessly through the tags using radio frequency readers. An RFID tag can be implanted or can be glued onto the animal's ear or collar, and they are relatively inexpensive. However, their monitoring zone is far less than that of GPS and, hence, the most suited in monitoring livestock issues where the area is confined, such as in feedlots and barns. Bluetooth and Wi-Fi tags communicate through short-range radio, and they are usually used to check the animals' movement in a confined area (e.g., a barn or a pasture). These transponders can be linked with the fixing of receivers all over the farm and a better response on the animals' movements and activities is achieved. Bluetooth beacons and Wi-Fi tags can be used well for indoor and short-range outdoor applications, although the data coverage requires additional support infrastructure sometimes [9].

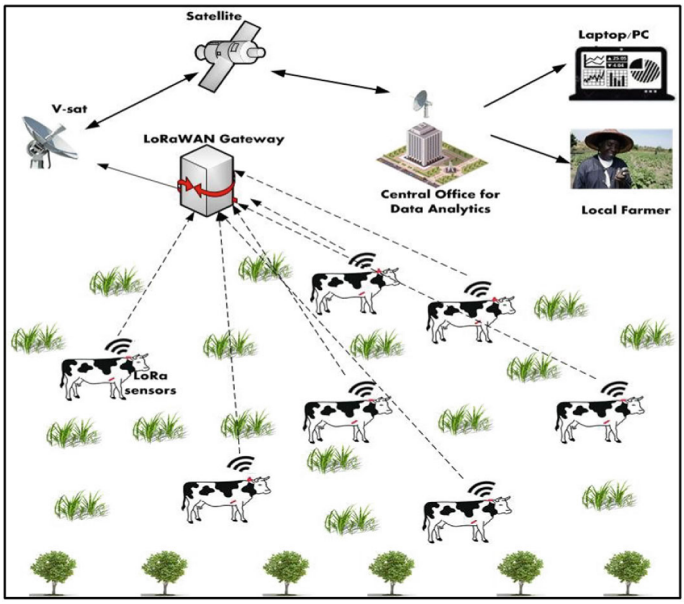


Fig. 3 System architecture design [7]

Satellite-based tracking systems utilize satellite communication networks which can be installed on mobile animals in areas that are far remote or inaccessible where usual cellular networks are not available. GPS technology-based networks work with satellite data, which can track and check everything over a big area. Both scientists and farmers find them helpful for watching over things in rural areas. Geotagging systems get better with sensor networks, which can easily monitor the health of animals. Other parameters like temperature, heartbeat, eating habits, and the weather come from tags. The tags go on collars. From these, farmers can see when an animal feels sick or stressed. Sensor networks also make farming more productive [10].

Farmers can use geotagging data to make smarter choices. The livestock breeders can do this by combining it with advanced analytical tools. Different geotagging methods have changed the way farmers care for their animals, for the better. Farmers can use their resources best. The animals are taken care of, and the work continues. Farming can be complicated. But with these new technologies and data methods, farmers can do well in the changing world of animal farming.

Recently, researchers have automated and monitored livestock recognition and identification for various animals, such as bears, sheep, giant pandas, and cattle, using new digital approaches based on computer vision (CV), deep learning (DL), machine learning (ML), and artificial intelligence (AI) [11]. With the use of artificial intelligence (AI) and machine learning (ML) algorithms, the geotagged data can be analyzed. The AI and ML shows the breeders various patterns, trends, and strange things in how animals act and feel. Breeders use this to predict when there might be

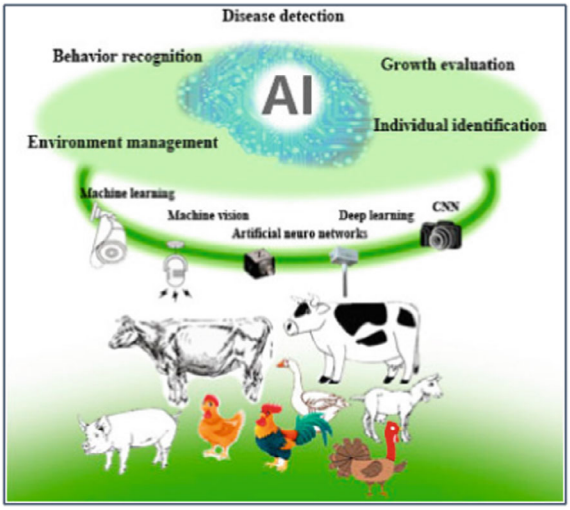


Fig. 4 AI application in animal farming [14]

a sickness. They manage their fields better and work more efficiently. By combining geotagged data, history, and weather, farmers can plan for the future. They lower risks and do the best for their animals. The various applications of AI technology in animal farming are shown in Fig. 4 [12–14]. Farmers and producers are very interested in animal identification and recognition for traceability reasons. Microchips, ear tags, tattoos, and radio frequency identification (RFID) collars are a few of the traditional methods used to identify livestock. Unfortunately, because of animal movement, direct contact, and bites, the majority of these identifying techniques are prone to being misplaced and destroyed. Moreover, electronic devices can be rather expensive for big herds of animals and can be physically traded or compromised for fraudulent purposes.

5 Navigating Challenges in the Implementation of Geotagging Livestock in India

The government has found several challenges, from technical issues to regulatory problems and many more because the tracking of animals in India is tough and time-consuming. Technical expertise is necessary to instruct farmers and animal owners in the proper use of equipment and software for geotagging. Many livestock owners may lack educational and technological knowledge making geotagging systems difficult to establish and maintain. Geotagging of livestock requires the acquisition and monitoring of sensitive information about the owners of the livestock and their animals. To prevent misuse or unauthorized access to sensitive data, it must be kept safe and confidential [1, 15]. In regulatory structure, the adoption of geotagging for livestock requires a strong legislative structure to regulate its use and ensure compliance

with international standards. In a country as wide and diverse as India, establishing and implementing such standards can be difficult. Factors like cost, reliability, and cultural preferences also could cause cattle owners not to adopt new technologies as quickly as desired. To ensure widespread success, it will be critical to convince cattle owners of the benefits of geotagging and to address their concerns successfully. Because geotagging is not widely used yet, integrating the technology with existing livestock management systems and government databases is a challenge and priority identified in the study. For the least duplication of efforts and the fewest data conflicts, different systems need to be compatible and needed to work in a coordinated manner. To overcome these challenges, government agencies, technology providers, livestock owners, and other stakeholders must collaborate to develop tailored solutions that are practical, cost-effective, and culturally appropriate [16].

6 Conclusion

This study at how geotagging technology is changing animal farming, especially in India. Using geotagging with geographical information systems (GIS) in animal farming has many benefits. Benefits like better production, disease surveillance, controlling grazing, and stopping cattle theft. This new way is updating the old methods. Not just in India, but worldwide, geotagging helps manage grazing, tracks wild animal migrations, and stops diseases. The study looks at many geotagging tools. Tools like satellites, artificial intelligence, GPS, RFID, Bluetooth, Wi-Fi tags, etc. But, using geotagging is not easy. There are problems like lack of infrastructure, cost issues, needing technical knowledge, and worries about data privacy. So, for geotagging to work well, everyone involved needs to help. In summary, geotagging proves to be a powerful instrument for improving livestock management techniques around the world, highlighting the significance of technology breakthroughs and cooperative ventures for the growth of sustainable agriculture.

References

1. Aharwal B, Roy B, Lakhani G, Yadav A, Baghel R, Saini K (2019) Livestock traceability: an overview. Int J Livestock Res (0):1. https://doi.org/10.5455/ijlr.20190616062444
2. FAO (2018) Global livestock environmental assessment model (GLEAM) availability in Italy. In: Food and agriculture organization of the United Nations, Rome. http://www.fao.org/gleam/results/en/. Accessed 12 Apr 2018
3. Bowling MB et al (2008) Review: identification and traceability of cattle in selected countries outside of North America. Prof Anim Sci 4:287–294. https://doi.org/10.15232/s1080-7446(15)30858-5
4. Daum T, Ravichandran T, Kariuki J, Chagunda M, Birner R (2022) Connected cows and cyber chickens? Stocktaking and case studies of digital livestock tools in Kenya and India. Agric Syst 103353. https://doi.org/10.1016/j.agsy.2021.103353

5. "Geo-tagging of cattle: another innovative step by Yogi government. National Herald, Aug 20, 2019. https://www.nationalheraldindia.com/india/geo-tagging-of-cattle-another-innovative-step-by-yogi-government. Accessed 03 Mar 2024
6. Digitisation drive: Kerala govt to geotag 10 lakh livestock farmer houses|India News—The Indian Express. The Indian Express, Feb 18 2018. https://indianexpress.com/article/india/kerala-govt-to-geotag-10-lakh-livestock-farmer-houses-5068679/. Accessed 03 Mar 2024
7. Abdullahi US, Nyabam M, Orisekeh K, Umar S, Sani B, David E, Umoru AA (2019) Exploiting IoT and LoRaWAN technologies for effective livestock monitoring in Nigeria. Arid Zone J Eng Technol Environ 15:146–159
8. Braunreiter C, Rothmund M, Steinberger G, Auernhammer H (2007) Potentials of GPS-collar application in pasture farming. In: Precision livestock farming '07, Brill|Wageningen Academic, pp 87–93
9. Alanezi MA, Shahriar MS, Hasan MB, Ahmed S, Sha'aban YA, Bouchekara HREH (2022) Livestock management with unmanned aerial vehicles: a review. IEEE Access 45001–45028. https://doi.org/10.1109/access.2022.3168295
10. Asmare B (2022) A review of sensor technologies applicable for domestic livestock production and health management. Adv Agric 1–6. https://doi.org/10.1155/2022/1599190
11. Dac HH, Gonzalez Viejo C, Lipovetzky N, Tongson E, Dunshea FR, Fuentes S (2022) Livestock identification using deep learning for traceability. Sensors 21:8256. https://doi.org/10.3390/s22218256
12. Ilyas QM, Ahmad M (2020) Smart farming: an enhanced pursuit of sustainable remote livestock tracking and geofencing using IoT and GPRS. Wirel Commun Mobile Comput 1–12. https://doi.org/10.1155/2020/6660733
13. Koompairojn S, Puitrakul C, Bangkok T, Riyagoon N, Ruengittinun S (2017) Smart tag tracking for livestock farming. In: 2017 10th international conference on ubi-media computing and workshops (Ubi-Media). https://doi.org/10.1109/umedia.2017.8074146
14. Bao J, Xie Q (2022) Artificial intelligence in animal farming: a systematic literature review. J Clean Prod 331:129956
15. Aquilani C, Confessore A, Bozzi R, Sirtori F, Pugliese C (2022) Review: precision livestock farming technologies in pasture-based livestock systems. Animal (1):100429. https://doi.org/10.1016/j.animal.2021.100429
16. Trotter M (2018) Precision livestock farming and pasture management systems. In: Precision agriculture for sustainability. Burleigh Dodds Science Publishing, pp 421–459

A Simulation-Based Approach to Evaluate the Effects of Cyberattacks Targeting Transducers in Control Systems

Marian Popescu and Emil Pricop

Abstract Industrial control systems (ICS) are critical in correctly and efficiently operating various industrial facilities, including petrochemical plants, refineries, wastewater plants, and production and manufacturing facilities. As critical components of the operational technology system of smart factories, ICS are connected to IT networks or the Internet and are often targeted by cyberattacks that exploit different vulnerabilities and breaches. Also, the attacks might come from inside the company if the attacker gets physical access to the control infrastructure. This paper focuses on analyzing the potential impact of cyberattacks on the field equipment level of hierarchical industrial control systems. A traditional automatic level control system for water accumulation is simulated, along with various attack scenarios. The simulation results demonstrate that cyberattacks on ICS can cause temporary service interruptions or incorrect system operation and physical damage to industrial infrastructure components, increasing their wear in a short period.

Keywords Industrial control system cybersecurity · Level control system · Cyberattack simulation · Cyberattack impact · Cyberattack effects

1 Introduction

Industrial control systems (ICS) are nowadays key components of all critical infrastructures. Moreover, ICSs are vital for the correct and optimal functioning of most modern factories operating under the Industry 4.0 concept. The complexity of control systems varies, ranging from local control loops to SCADA systems and integrated production facilities. The most important and common aspect is that the correct operation of the system, whether simple or complex, is based on reliable data transfer and processing. Given this, the ICS has become a target for various types of attacks

M. Popescu · E. Pricop (✉)
Automatic Control, Computers and Electronics Department, Petroleum-Gas University of Ploiesti, Ploiesti, Romania
e-mail: pricop@upg-ploiesti.ro

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_17

and threats, which can affect their functioning, lead to economic losses, and harm human lives.

For a long time, the main focus in the industrial field was on designing and implementing high-performance systems that can monitor, control, and optimize technical processes. Modern computing-based systems were integrated with pre-existing physical systems to facilitate data exchange between enterprise systems and users. Additionally, these systems offered remote access to data, which was considered a significant achievement. However, this evolution also led to an increase in the vulnerability of industrial systems to cyberattacks. Remote intervention in control systems became possible, posing a threat to the safety and security of the facility, personnel, and surrounding communities. These problems led to the emergence of a novel development area and research topic: the security of automation systems against potential external threats and avoiding any interruptions in the operation of control systems.

2 Related Work

Since industrial control systems are complex structures comprising IT and industrial elements, the threats can be broadly categorized into IT-related and industrial-specific.

The first category includes well-known IT security threats, which have been extensively studied and often countered. Poorly secured systems are vulnerable to interruptions caused by computer attacks, viruses, and improper operations, leading to significant economic losses and even accidents with human life at stake. The attacks are similar to those on the Internet and classical computer networks.

The security threats specific to industrial processes are continuously evolving and have become a significant concern for many security companies, as presented in reports as [1–3]. The nature of attacks on industrial facilities can be hazardous if they lead to a physical impact. The attackers target SCADA systems and control systems components, such as sensors, trying to modify the values sent to the controllers, thereby disrupting the entire process. Generally, the industrial processes attacked are critical and have a complex automation structure, making them vulnerable to security issues. Understanding the different attack types that could be launched against industrial systems and the methods used to mitigate them is essential. For instance, attacks on the sensors can be prevented by implementing encryption mechanisms to protect the data transmitted from the sensors to the controllers and authentication mechanisms, such as the one proposed in reference [4]. Similarly, next-generation firewalls and intrusion detection systems (IDSs) can be used to prevent unauthorized access to the network and systems. Additionally, properly training employees on security best practices can go a long way in preventing attacks on industrial processes.

Moreover, it is crucial to maintain a balance between security and system performance, ensuring that security measures do not negatively impact the automation system's ability to achieve its objectives. Ultimately, protecting the system's functions should remain a top priority.

As stated, any existing system is likely to be vulnerable to a given type of attack launched from inside or outside the infrastructure, whether it is the IT or the OT sector. This is one of the reasons we need to consider any possible attack scenario to develop proactive security measures. This paper evaluates the potential effects of attacks targeting transducers in simple control systems.

The paper is structured as follows. The second section focuses on control systems design and architecture basics. The third section comprises the simulation of a basic automatic level control system for the water accumulation process. The fourth section presents various scenarios and discussions on the impact of cyberattacks on the simulated system.

3 Industrial Control Systems Architecture

Process control is a complex activity that consists of directing a technical process evolution in time by applying commands to achieve imposed objectives from three main categories as presented in the references [5] and [6]:

- quality objectives;
- efficiency objectives;
- security objectives.

The objectives mentioned above can be attained by using specific systems. In order to evaluate the status of the process, a monitoring (measurement) system is needed. For example, the efficiency objective of a process is managed by an optimization system. It will be an automatic optimization system if its operation does not require human operator intervention.

The security objective is related not only to the cybersecurity of the process but also to the safe operation even in unexpected conditions, so there will be no harm to humans, the environment, and infrastructure. It is attained by using various safety and protection systems.

The quality objective is specific to process control systems. The main scope of an automatic control system is to attain and maintain a reference process state without human intervention. The control systems (ICS) range from simple structures—conventional systems that control only one parameter—to complex or advanced systems capable of controlling the whole process (many parameters).

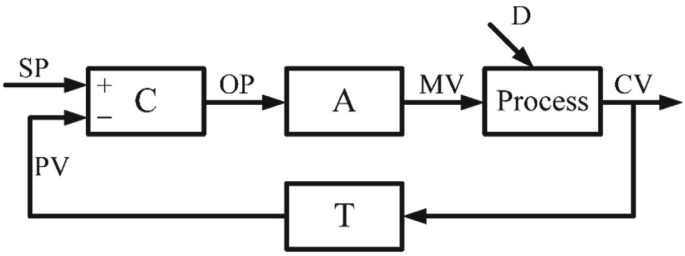


Fig. 1 Block diagram of a control system: C—Controller; A—Actuator; T—Transducer; SP—setpoint; PV—Process variable; OP—Control signal (Controller output); MV—Manipulated variable; CV—Controlled variable; D—Disturbance

The basic control system design, as presented in Fig. 1, comprises the following elements:

- transducer (sensor)—T;
- controller—C;
- actuator—A.

The transducer measures a parameter and usually converts it into an electrical signal, which is transmitted to a controller. The controller compares the parameter's current value with the desired one (called setpoint) and, based on the difference between the two, calculates a control signal and sends it to an actuator. The actuator modifies a manipulated variable so that the controlled variable equals the setpoint.

The actual industrial control systems are complex structures that feature both distributed and hierarchical elements. These systems are interconnected using specific industrial networking protocols, such as Modbus, Profibus, Profinet, Hart, and Ethernet/IP. Often, they are connected in SCADA—supervisory control and data acquisition systems that could be remotely operated. The possible cyberattacks on the systems vary with the system's complexity. Still, in the context of this paper, we will analyze attacks that could happen in elementary structures of control systems.

4 Simulation of a Simple Process Control System

To investigate the possibility of using simulation to evaluate the potential impact of a cyberattack on the functioning of a control system, we considered a simple process and the corresponding process control system: the water accumulation process.

Figure 2 presents the structure of a water tank level control system.

In order to study this process, a simulation was built using the UniSim® Design environment.

The feed stream was configured in the simulator as a mixture of water (90 mol%) and air (10 mol%). Other conditions include:

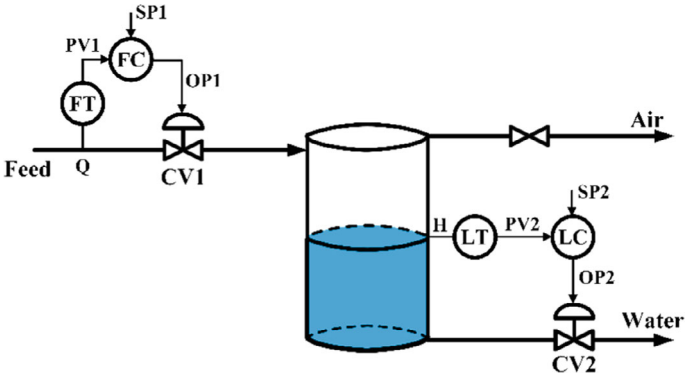


Fig. 2 Water accumulation process: FT—Flowrate transducer; FC—Flowrate controller; LT—Level transducer; LC—Level controller; CV1, CV2—Control valves; Q—Feed flowrate; SP1—Flowrate setpoint; H—Tank liquid level; SP2—Level setpoint; PV1, PV2—Process variables; OP1, OP2—Control signals

- feed flow rate—5 m^3/h;
- pressure—6 bar;
- temperature—25° C.

A flow rate control system is present on the feed stream to maintain the feed flow rate at a desired value.

The water tank has a volume of 4 m^3. The vessel liquid level is maintained using a level control system that modifies the position of the outlet stream control valve (CV2).

The simulation diagram from the UniSim® Design environment is presented in Fig. 3. It corresponds to the system presented in Fig. 2.

In order to have a stable system to work with, first, the flow rate controller was tuned, and to test its performance, the setpoint was modified as presented in Fig. 4.

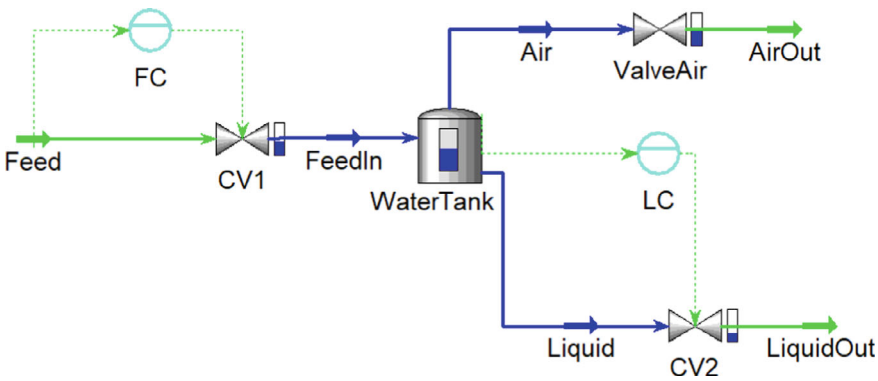


Fig. 3 Simulation diagram from UniSim® design environment

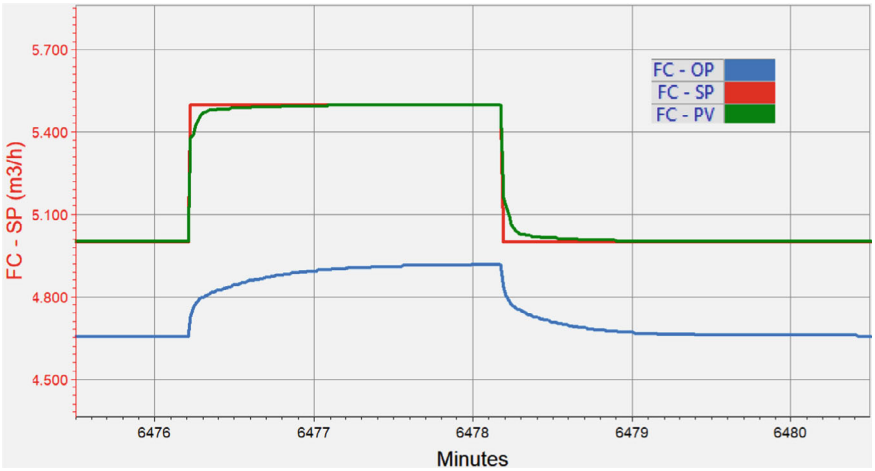


Fig. 4 Time evolution of flowrate at setpoint changes

The next step was to analyze the level control system by tuning the level controller and modifying the setpoint. Figure 5 presents the time evolution of the level at the setpoint changes.

The evolution from Fig. 5 is associated with the regular operation of the level control system.

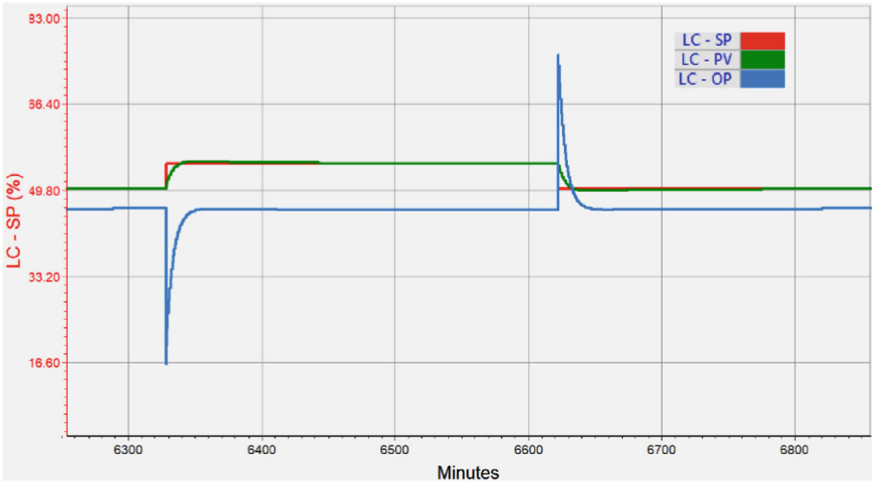


Fig. 5 Time evolution of level at setpoint changes

5 Simulation of Cyberattack on the Process Control System

5.1 Attack Scenario 1—Level Transducer Unavailability/ MITM Attack

Pricop et al. [7] presents the idea that an insider threat is a dangerous type of threat in any industrial control system or any IT system. In the analyzed system, an insider attacker can gain physical access to the control system and make the level transducer unavailable. It could be a physical removal of the transmission line or a DDoS-like attack that will interrupt communication between the transducer and the controller.

The same scenario corresponds to a man-in-the-middle (MITM) attack. Most control systems use standardized communication protocols, such as Modbus/TCP, Profibus, and Profinet, which are not complicated to replicate, especially by using existing open-source libraries, such as [8] or [9]. Using a simple Arduino board or any other microcontroller-based development system, the attacker can create a simple replacement for the transducer and force the device to transmit a fixed value. If the attacker obtains physical access to the infrastructure, the device can physically replace the transducer, and the MITM attack could be in place effortlessly. Its impact could be catastrophic, as presented below.

This scenario can be simulated in the UniSim® Design setup by setting the controller PV input to receive 0. Such an attack has the effect illustrated in Fig. 6.

From Fig. 6, it can be seen that when the signal from the transducer to the controller is most likely interrupted or set to zero (PV = 0%), and considering the setpoint unmodified (50%), the controller tries to compensate and completely closes the control valve (OP = 0%). Because there is no actual feedback, this has the effect

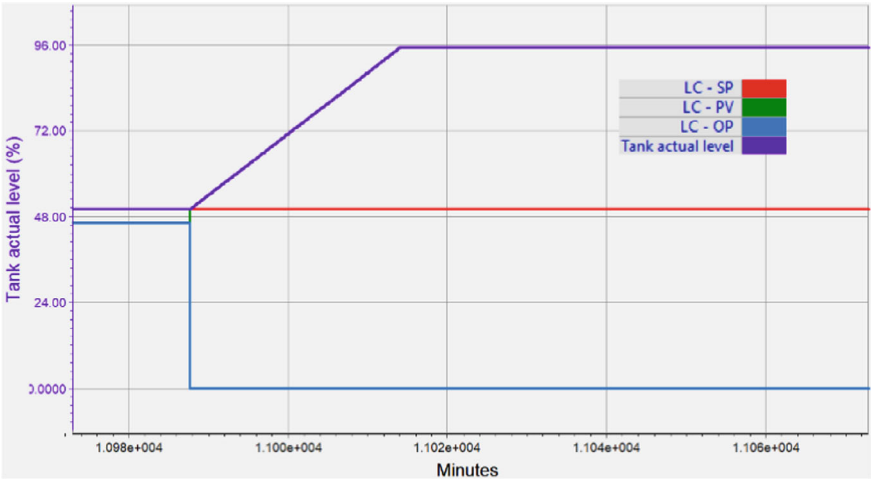


Fig. 6 Impact of transducer signal interruption

of increasing the tank's actual level to the maximum possible value. In the case of the analyzed process, the effect is only filling up the tank and affecting the quality objective. Still, in the case of other processes (e.g., flammable liquid accumulation, gas accumulation, heating of a distillation column products), the result can be an explosion that could harm humans and the environment and could lead to substantial economic losses by affecting all the process control objectives.

5.2 *Attack Scenario 2—Man-In-The-Middle Attack—Spoofing the Level Transducer*

An attacker can easily spoof the level transducer by mainly using open-source software and publicly available information. The perpetrator should first do a reconnaissance attack to correctly identify the communication protocol for connecting the transducer and the controller. In most cases, it will be a well-known, standardized, industrial protocol that, unfortunately, lacks any authentication mechanism for the communicating entities as presented in references [4, 10, 11]. The lack of any source of data authentication will allow the attacker to spoof the sensor and inject a fixed measured PV in the control system.

The simulation of this situation is presented in Fig. 7. Two situations were taken into consideration: the attacker set PV is below and, respectively, above the setpoint.

In the first case, the value provided by the cyberattack as PV is 45%, and because it is lower than the setpoint, the controller progressively closes the control valve. Considering that the PV has a fixed value and there is no actual feedback, the control

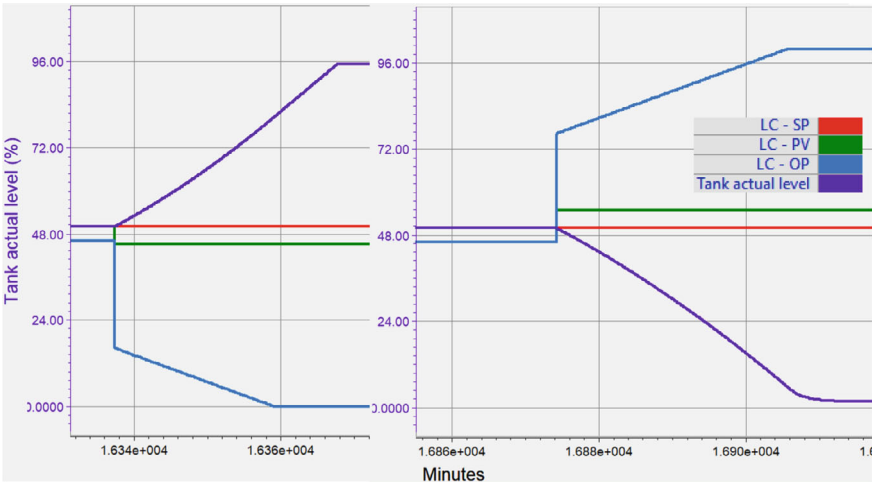


Fig. 7 Effect obtained when cyberattack provides as PV fixed values

valve will be closed completely. The effect will be the increase of the actual level to the maximum value, similar to the one presented in attack scenario 1.

In the second case presented in Fig. 7, the value provided by the spoofed transducer as PV is 55%. Because it is greater than the setpoint, the controller progressively opens the control valve (CV2). With a fixed PV and no actual feedback, the control valve will be fully opened, leading to the decrease of the actual level to the minimum value.

5.3 Attack Scenario 3—DDoS-Like Attack on the Level Transducer

This attack scenario presented in this section relies on the concept of Denial-of-Service/Distributed Denial-of-Service (DoS/DDoS). These types of attacks can cause significant latency in communication by overwhelming a device or a network with traffic or requests, rendering it inaccessible to legitimate users. Basically, a DoS or DDoS attack floods the target device with a high volume of requests or traffic, causing it to slow down or even crash. In this case, legitimate traffic will be delayed, as there will be an increased latency. The following simulation will investigate the potential effects of such delays.

The corresponding simulation scenario involves delaying the signal from the level transducer, as it is the supposed effect of a DDoS-like attack. This is simulated in the UniSim® Design environment using a transfer function module on the PV transmission line, as presented in Fig. 8.

In order to formulate a conclusion, various time delay values should be tested. The effect on the level control system's functioning might be different and vary with the time delay, as presented below.

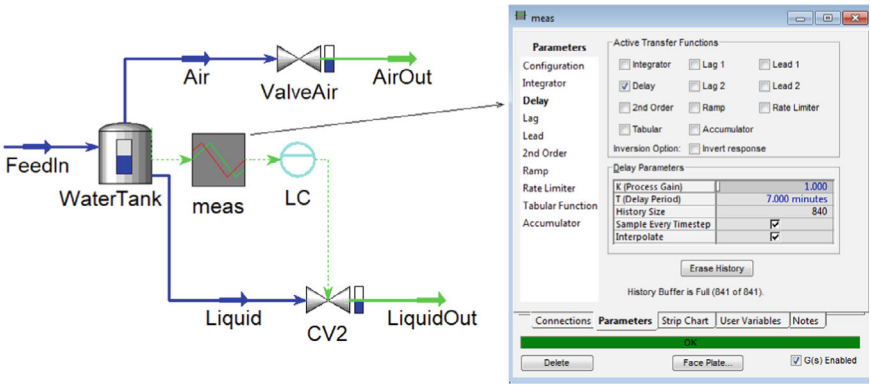


Fig. 8 Use of a transfer function module for signal delay simulation

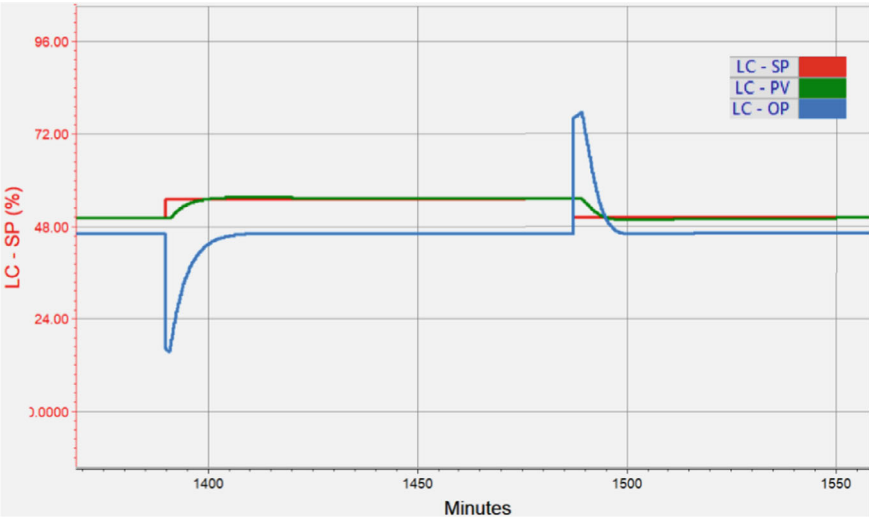


Fig. 9 Effects of modifying time delay (1–2 min) on level transducer signal

When the PV value has small delays (1–2 min), the control system's transient time increases as presented in Fig. 9. This leads to reduced system efficiency, high resource, and energy usage and negatively affects the quality and efficiency objectives. In the case of some processes, an increased transient time will affect the system's safety.

When the time delay is more prolonged (e.g., time delay = 5 min), the system response starts to oscillate as it can be seen from Fig. 10. When the time delay increases even further (time delay = 7 min), the level control system becomes unstable (as illustrated in Fig. 11) with damaging effects on the control valve.

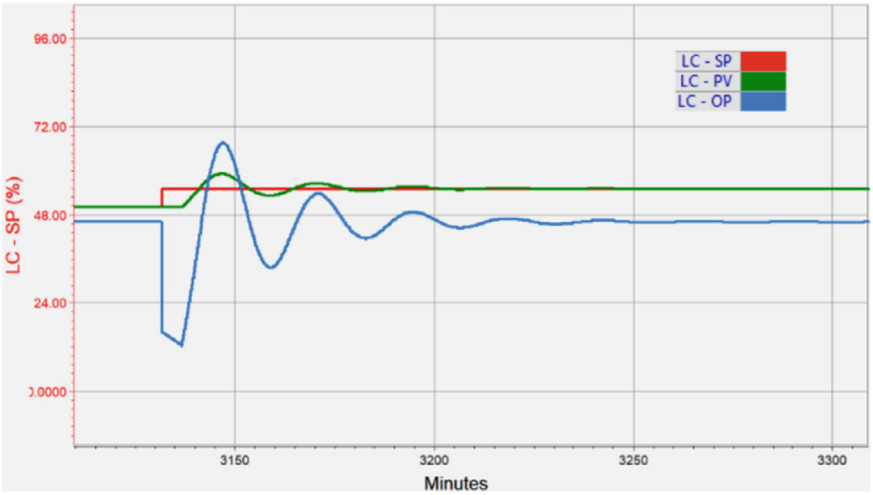


Fig. 10 Effects of modifying time delay (5 min) on level transducer signal

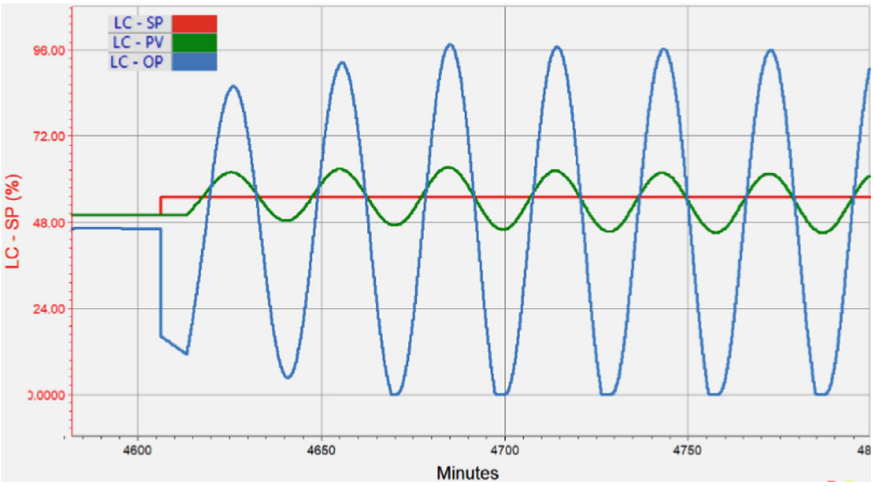


Fig. 11 Effects of modifying time delay (7 min) on level transducer signal

6 Conclusion

Cyberattacks targeting the components of control systems are likely to happen since industrial facilities have become the preferred target of attackers, ranging from script kiddies to state actors.

Modern industrial control systems are sophisticated and incorporate both distributed and hierarchical elements. Interconnected through industrial networking

protocols like Modbus, Profibus, and Profinet, these systems often operate within SCADA setups, allowing remote monitoring and control. The complexity of such systems demands a thorough understanding of potential cyber threats. In order to understand the security of complex systems, it is necessary to experiment with basic control structures. In the paper, a simple level control system for the water accumulation process in a tank was simulated using the UniSim® Design. The simulation was customized in order to evaluate the impact of various cyberattack scenarios targeting the level transducer: device unavailability, man-in-the-middle/transducer spoofing attack, and latency generating (DDoS-like attack).

The simulation results underscore the critical importance of cybersecurity in industrial control systems. Cyberattacks on even elementary structures can have severe consequences, affecting efficiency, safety, and quality objectives. Also, the attack effects can lead to physical damage to the components of the installation. Understanding and mitigating potential threats are essential for ensuring the reliability and security of industrial processes.

Acknowledgements/Funding This work was supported by a grant from the Petroleum-Gas University of Ploiesti, Romania, project number 11060/08.06.2023, IME-EMERGENSEC, within the Internal Grants for Scientific Research.

References

1. Cybersecurity OT (2024) The 2023 year in review report. Dragos Inc., Washington D.C. USA
2. The Crisis of Convergence: OT/ICS Cybersecurity in 2023. TxOne Networks (2024)
3. Goncharov E (2024) ICS and OT threat predictions for 2024. Kaspersky Industrial Control Systems CERT
4. Pricop E, Fattahi J, Paraschiv N, Zamfir F, Ghayoula E (2017) Method for authentication of sensors connected on Modbus TCP. In: 2017 4th international conference on control, decision and information technologies (CoDIT). IEEE, Barcelona, pp 0679–0683. https://doi.org/10.1109/CoDIT.2017.8102673
5. Paraschiv N, Popescu M (2014) Distributed systems for monitoring and control (in Romanian). Editura Universitatii Petrol-Gaze din Ploiesti
6. Paraschiv N (2015) A hierarchical approach to automation functions. Mircea cel Batran Naval Academy Scientific Bulletin. XVIII:223–228
7. Pricop E, Mihalache SF, Fattahi J (2016) Innovative fuzzy approach on analyzing industrial control systems security. In: Pricop E, Stamatescu G (eds) Recent advances in systems safety and security. Springer International Publishing, Cham, pp 223–239. https://doi.org/10.1007/978-3-319-32525-5_13
8. PyModbus—A Python Modbus Stack. https://github.com/pymodbus-dev/pymodbus
9. P-net—Profinet device stack for embedded systems. https://github.com/rtlabs-com/p-net
10. Bîrleanu FG, Anghelescu P, Bizon N, Pricop E (2019) Cyber security objectives and requirements for smart grid. In: Kabalci E, Kabalci Y (eds) Smart grids and their communication systems. Springer Singapore, Singapore, pp 607–634. https://doi.org/10.1007/978-981-13-1768-2_17
11. Hayes G, El-Khatib K (2013) Securing Modbus transactions using hash-based message authentication codes and stream transmission control protocol. In: 2013 Third international conference on communications and information technology (ICCIT). IEEE, Beirut, Lebanon, pp 179–184. https://doi.org/10.1109/ICCITechnology.2013.6579545

Urban Greens—An Initiative Towards Clean Air

Auckloo Pritee, Sanjay Ojha, Anupama Joshi, Himanshi Gupta, Hrithik Pandey, Manisha Bhakuni, and Shikha Bisht

Abstract For years, Delhi, India's capital, has been struggling with a severe air quality crisis. The city's air quality is influenced by a complex interplay of factors, including vehicular emissions, industrial pollution, agricultural residue burning and inappropriate land allotment to tree plantation. The resultant air pollution poses significant health risks, including respiratory illnesses and cardiovascular problems for its residents. So, what is the solution? The obvious solution towards a better air quality is **planting more trees**. This paper supports SDG goals towards sustainable and responsible development of urbanization in an age where we are confronted with increase in pollution. It consists of an expert system based on machine learning which suggests the required tree percentage to make the AIR QUALITY INDEX better or more sustainable practices based on the land availability in the particular districts of Delhi. This proposed system is trained using data collected in our research and analysis. It is not limited to predict only the tree percentage required to plant but also suggest methods other than plantation to beat pollution. Also, it will suggest ways and methods to maintain a healthy tree. This system uses Google Earth API to analyse the districts for the plantation, SWI-PROLOG for building an expert system. An interactive UI is designed using FLASK to make it easy for the user to enter the required input data.

Keywords Air quality index · Google earth API · SWI-PROLOG

A. Pritee
Mauritius Institute of Education, Reduit, Mauritius

S. Ojha (✉)
Centre for Development of Advanced Computing (C-DAC), Noida, India
e-mail: sanjayojha@cdac.in

A. Joshi · H. Gupta · H. Pandey · M. Bhakuni · S. Bisht
GGSIPU, Delhi, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_19

1 Introduction

"Ahem Ahem", "Sorry I had a little cough and a sore throat". These are some very common lines that we hear almost every day in all the metro cities. Especially in our national Capital New Delhi, which is now the most air-polluted capital in the whole world. Air quality is a privilege that no one in the capital can attain. As a matter of fact, the Hon'ble Supreme Court of India deemed air quality in Delhi as harmful as smoking 20 cigarettes in a day. Many people in Delhi who have never smoked a day in their lives are getting lung diseases, and foreign tourists who visit Delhi have difficulty in breathing the minute they land at the IGI airport. So what is the solution? The answer is **more trees**. But Where? When? What types of trees? And what proportion of area should be allotted to growing trees? Growing trees is not a day's work, so how can we tackle the time factor? How the AIR QUALITY INDEX of Delhi can be improved, or a better question, can it be improved?

This paper presents a solution in a more sustainable and smart way. It consists of an expert system based on machine learning which suggests the required tree percentage to make the Air Quality Index (AIR QUALITY INDEX) better, more sustainable practices based on the land availability in the particular districts of Delhi, ways to maintain a healthy tree and methods other than plantation to beat pollution. This proposed system is trained using data collected in our research and analysis.

1.1 Problems Identified

1.1.1 Unequal Distribution of Tress

Taking reference of the maps from Figs. 1, 2 and 3 we can easily conclude that there is unequal distribution of trees among the several districts of Delhi. Only South Delhi and New Delhi region is dominantly covered with trees. Other regions have a low percentage of trees. Subsequently, we can observe the difference in AIR QUALITY INDEX even during peak days. Even from the past 12 years there is no significant change in tree percentage.

The above AIR QUALITY INDEX problem and this tree distribution problem led us to the conclusion that there is a need for a proper system that would suggest how many trees are needed in a particular area.

1.1.2 Causes for Sudden Change in AIR QUALITY INDEX—Winter, Diwali, Humidity, Fires

AIR QUALITY INDEX in Delhi starts worsening in October and winter strikes in November. This is accompanied by stubble burning by farmers of Punjab and Haryana. Also, Smog from the onset of winters coincident with Diwali causes the

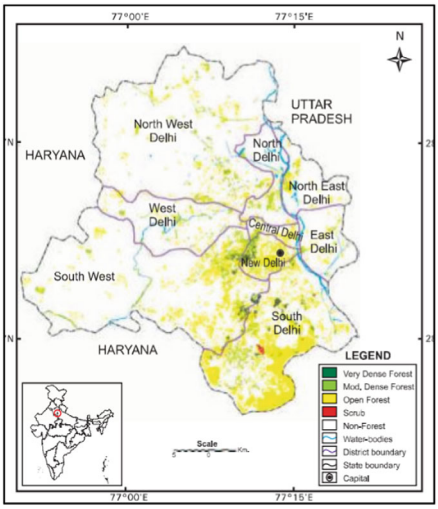


Fig. 1 *Source* Forest Survey of India 2011 Tree Density: 11.88%

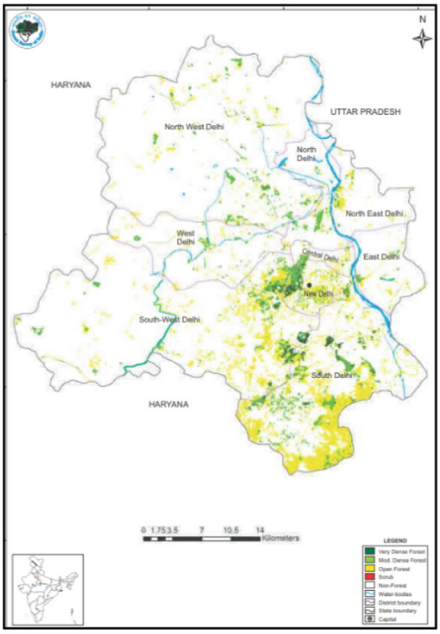


Fig. 2 *Source* Forest Survey of India 2019 Tree Density: 13.18%

peak air pollution in Delhi. As the winter season sets in, dust particles and pollutants in the air become unable to move. Due to stagnant winds, these pollutants get locked in the air and affect weather conditions, resulting in smog.

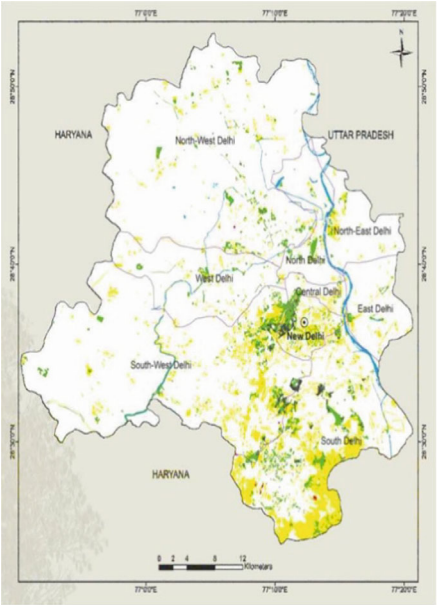


Fig. 3 *Source* Forest Survey of India 2021 Tree Density: 13.157%

1.1.3 No Space for Tree Plantation—Industrial Area

There are some industrial areas in Delhi in which pollution is drastically high like Mundka. These are completely occupied, and there is barely any land available for plantation. In such areas, even available trees are not sufficient or not maintained properly.

1.1.4 Delhi's Deteriorating Air Quality Index (AQI)

From Table 1, we can say that Delhi's AIR QUALITY INDEX has been getting worse back to back especially during November to January due to several reasons like seasonal change to winters, smog during Diwali and fires in Punjab and Haryana. These are the significant reasons for bad AIR QUALITY INDEX at that time.

1.1.5 Strictness by the Government and Abiding by the Rules

Even after having the rules and regulations, they are not strictly followed or abided by the government officials and by citizens as well. There is a need for a strict regulating body that keeps eye on this matter. Countries with enhanced air quality have demonstrated effective rule enforcement across various departments concerning AIR QUALITY INDEX, ensuring strict adherence to regulations. China serves as a notable example of this successful approach.

Table 1 Year-wise Delhi's AQI

Month	2018	2019	2020	2021	2022
January	328	328	286	324	279
February	243	242	241	288	225
March	203	184	128	223	217
April	222	211	110	202	255
May	217	221	144	146	212
June	202	189	123	147	190
July	104	134	84	110	87
August	111	86	64	107	93
September	112	98	116	78	104
October	269	234	266	173	210
November	335	312	328	377	320
December	360	337	332	336	319

All of the above problems led us to creation of this expert system which gives the result according to the current situation and close to the realistic implementation. This system will not be restricted only towards providing the required tree percentage to improve the AIR QUALITY INDEX but also it provides solutions for areas that do not have land available for plantation or already have required trees. Also, it will suggest methods for good maintenance of trees and solutions other than trees.

2 Literature Survey

2.1 Need of the Study

Air pollution is one of the greatest environmental risks to health. Reducing air pollution levels can significantly alleviate health risks, including stroke, heart disease, lung cancer and respiratory ailments like asthma, contributing to the prevention of 6.7 million premature deaths linked to ambient and household air pollution each year.

2.2 How Does AIR QUALITY INDEX Work?

The AIR QUALITY INDEX (AQI) is a numerical measure of air quality, where higher values indicate increased pollution levels and greater health risks.

The AIR QUALITY INDEX (AQI) scale ranges from 0 to 500, with levels categorized as acceptable below 100, unhealthy for sensitive groups above 100, harmful above 150 and hazardous above 300 [1].

Five Major Pollutants

EPA establishes an AIR QUALITY INDEX for five major air pollutants regulated by the Clean Air Act [1]:

- Ground-level ozone,
- Particle pollution (also known as particulate matter, including PM2.5 and PM10),
- Carbon monoxide,
- Sulphur dioxide,
- Nitrogen dioxide.

2.3 Data Collection

We have chosen the cities which have good and moderate Air Quality Index throughout the year and having similar weather to Delhi. The cities consist of similar constraints like industries and crowd but still maintaining good AIR QUALITY INDEX. Collecting data from cities with good air quality and a sufficient number of trees is important to teach a smart system about how to improve air quality.

We have not considered coastal areas and dense forest areas. This is because these places are mostly clean and green due to less industries, high forest cover and less crowded. Considering such data will be unrealistic for training.

2.4 Tools and Technology

Tools and technology used are mentioned in Table 2.

3 Proposed System

Our expert system will take the input of the required area. For the input data, we have a total of 9 districts of Delhi after taking 3 districts (East, Shahdara, North East) together. Detailed working is explained in Table 3. Flowchart for the same is depicted in Fig. 4.

4 Methodology

STAGE 1: Data Collection

Fields in the dataset are as follows:

Table 2 Technology used in stages

Technology	Used in stage
Artificial intelligence	Stage 4
PySWIP	
Prolog	
Machine learning	Stage 3
XGBoost	
Pandas	
Sklearn	
Google earth engine	Stage 2
Google earth engine API	
Google cloud SDk	
ESA WorldCover V200	
To display map	Stage 5
Leaflet	
OpenStreetMap	

City/District: Cities consistently maintaining a good Air Quality Index throughout the year.
State: States to which Cities belong.
AQI: AIR QUALITY INDEX of the identified cities according June 2023.
Tree cover (in percentage): Present Tree percentage according 2021 FSI Survey.
Land (in km²): Land area of that particular City.
Latitude: The measurement of distance north or south of the Equator.
Longitude: The measurement east or west of the Prime Meridian.
Weather (in Celsius): Temperature in Celsius.
Humidity (in percentage): Humidity of the city in June 2023.
Population Density: According to 2011 census from government official website.
Precipitation: Data from Indian Meteorological Centre.
Traffic: Divided the traffic on the scale of 1–5.

An example of dataset is shown in Fig. 5.

STAGE 2: Identifying Occupied Vegetational and Non-Vegetational Urban Land

Successful use of IRS LISS-3.

We are first analysing the input data and scanning the area through satellite using IRS LISS-3. This results in the percentage of empty land, crop land, green cover and built cover. Successful attempt of IRIS image scanning is shown below in Fig. 6.

STAGE 3: Building and Training Algorithm Based on Previous Data to Get Required Percentage of Trees

The desired input area is scanned.

Table 3 Working of expert system

Working	
Input	Required area from drop down menu
Output Then the area will be scanned through our integrated Google earth engine API and fetches 4 observations as output	Tree cover percentage—present green cover (forest cover) built up—already occupied by buildings or roads cropland—low dense green cover like farms bare area—empty land
We have a target AQI from 100 to 150. System will compare the required tree percentage to the present tree percentage and provide three types of outputs based on the conditions	
The area has bad AQI but tree density is enough	System will suggest methods to maintain the trees like artificial rain, water sprinkling on roads and plants and other ways other than trees to improve the air quality. For example: liquid trees adopted by Serbia
The area has bad AQI and land available for plantation	System will provide effective ways of tree plantation like Miyawaki plantation. It will also suggest the best trees according to the major pollutant
The area has bad AQI but no land available for plantation	System will search for plantation land in nearby areas and also it will suggest ways to improve the air quality by implementing certain methods and technologies like hydrogen power plants, traffic management, tree maintenance, liquid trees
If the target area has required tree density	If the target area already has the required tree density then the system will suggest methods to maintain the trees
After training our system we have reached an accuracy of 69% using xgboost model with test size data 20%. Users will be able to interact through UI. The website will show a map of the selected district	

Model is built and trained using XGBoost and train test split from Sklearn. Data is splitted as 80% for training and 20% for testing. Model is trained with the columns aqi, land area, latitude, longitude, weather, humidity, population density, precipitation, and traffic. Model is giving an accuracy of 69%. Pandas is used to make the dataframe. After training, we will get the expected tree percentage for that particular area on passing the coordinates of the desired area through user interface.

Also, we are predicting the number of trees that can be planted in that bare area considering a tree taking 25 m^2.

STAGE 4: The Expert System Designing

Expert system is designed using PySWIP and prolog. PySWIP enables querying SWI-Prolog in python programmes. Prolog is a programming language used for logic-based AI applications. The flowchart showed above is implemented using prolog in python programme by applying conditions in logic statements, conditions to compare current tree density and expected tree density to produce relevant output according

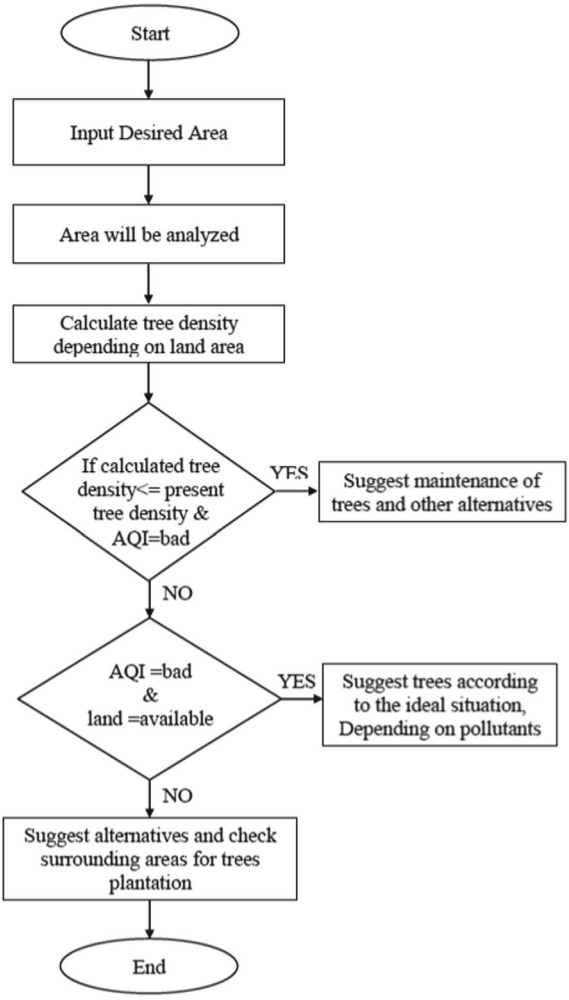


Fig. 4 Flowchart of working model

CITY/ DISTRICT	STATE	AQI	TREECOVER	LAND	LATTITUDE	LONGITUDE	WEATHER	HUMIDITY	POPDENSE	PERCIPITATION	TRAFFIC
ETAWAH	UTTAR PRADESH	50	10.88	2311	26.7767	79.023	40	25	683	-98	3
MIRZAPUR	UTTAR PRADESH	60	18.25	4405	25.1398	82.5804	40	30	561	-43	3
BIKANER	RAJASTHAN	90	0.85	30239	28.0229	73.3119	40	33	78	99	3
SAMBALPUR	ODISHA	80	49.7	6624	21.4669	83.9752	40	41	158	-42	3
MATHURA	UTTAR PRADESH	50	1.71	3340	27.4924	77.6737	39	25	761	-37	3
VARANASI	UTTAR PRADESH	50	1.11	1535	25.3176	82.9739	39	25	2,399	91	3

Fig. 5 Sample dataset used for training

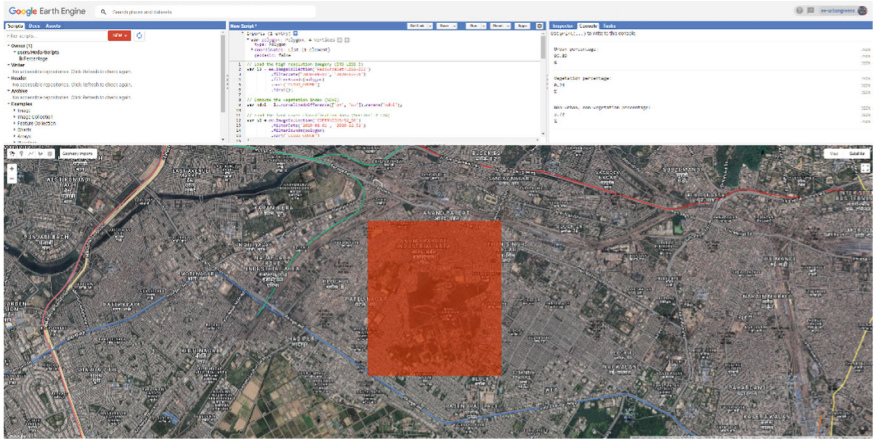


Fig. 6 Area scanning

to present conditions. District-wise solutions have been provided so that the system would return the most optimal solution.

STAGE 5: Integrating the Modules by Providing Ease to the User Through UI

Here is a glimpse of user interface in action as shown in Fig. 7.

4.1 Solutions Given by Expert System

After evaluating the required tree density and other necessary conditions, system will behave as described in Table 4.

4.2 District-Wise Solutions

East Delhi/Shahdara/North East Delhi

Major Hotspot: ISBT, Anand Vihar, Vivek Vihar, Patparganj Industrial Area, Sahibabad, Ghazipur Landfills.

- Use of smog guns on regular basis.
- Smog towers (Already present but not operational).
- Use of chimneys by industries in Patparganj and Sahibabad to bring the level of particulate matter down.
- Water sprinkling.
- Covering loose soil and construction waste with netting.

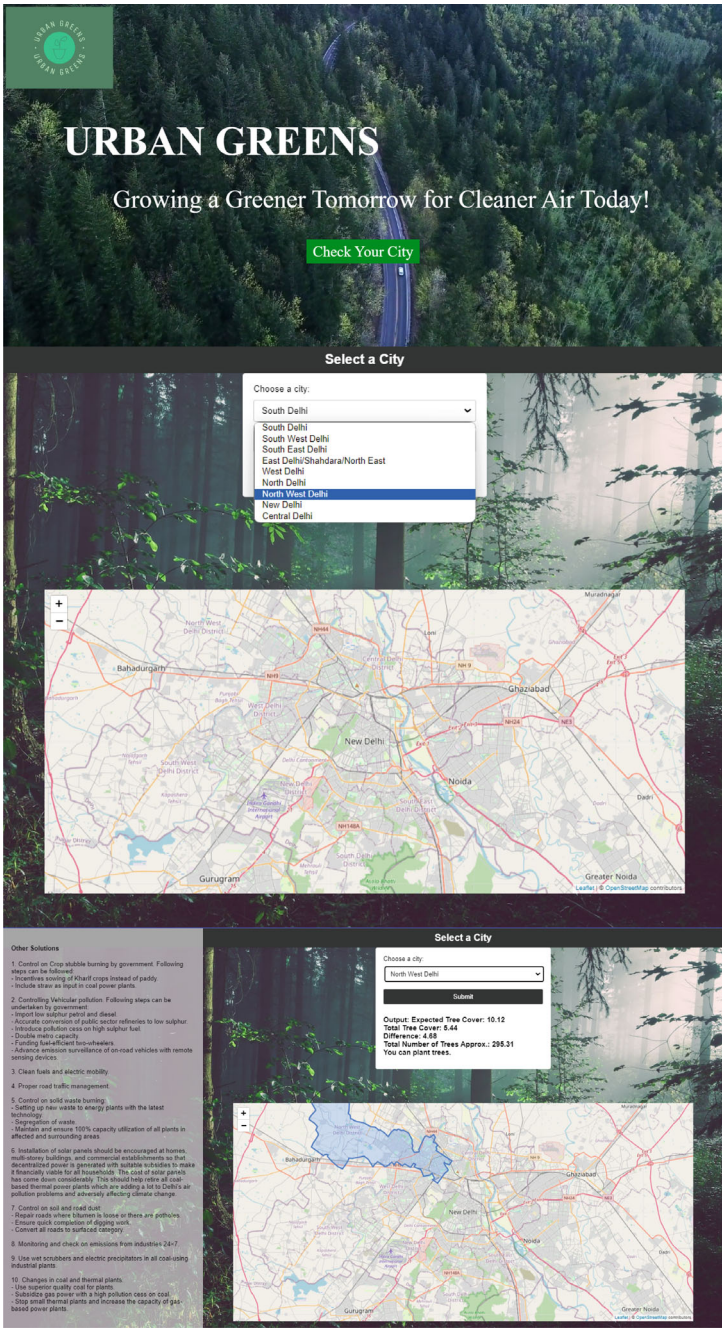


Fig. 7 User interface to interact

Table 4 Solution proposed by algorithm

Solutions
Instead of tree plantation 1. Liquid tree [2] 2. Water sprinkler
If land is available for plantation 1. Trees for suggestion—Peepal, Neem, Banyan, Saptaparni, Jamun, Devdar, Champa 2. Congested farming, Miyawaki farming [3, 4]
Large scale strategies (by government) 1. Hydrogen power plants [5, 6] 2. Artificial rain 3. Smog towers 4. Traffic management policies 5. Monitoring construction sites 6. Prohibition of burning of garbage in open areas 7. Water sprinkling on roads 8. Mandatory allocation of land to green cover in societies 9. Strictness by government

- Cleaning roads near transportation hubs reduces dust pollution.
- Encourage the use of hydroelectric vehicles [7].
- Construction sites must be covered and netted properly.
- The installation of a diesel particulate filter can greatly diminish emissions from diesel vehicles.
- Installing De-Sox-ing and De-NOx-ing systems in Delhi's power plants and refineries reduces SO2 and NOx emissions.
- Waste management system.
- Installing perforated high-density polyethylene pipes and establishing a dedicated surveillance squad are crucial measures to prevent the build-up of highly flammable gas and closely monitor activities at dumping sites.

South West Delhi

Major Hotspot: Dwaraka, RK Puram, Najafgarh, Naraina Vihar.

- Use of smog guns on regular basis.
- Smog towers (Already present but not operational).
- Water sprinkling.
- Traffic management.
- Covering loose soil and construction waste with netting.
- Cleaning roads near transportation hubs reduces dust pollution.
- Encourage the use of hydroelectric vehicles.
- Construction sites must be covered and netted properly.
- The installation of a diesel particulate filter can greatly diminish emissions from diesel vehicles.
- Waste management system.

- Control on crop stubble burning by government.
- Incentives sowing of Kharif crops instead of paddy include straw as input in coal power plants.
- Regular maintenance of boilers, such as dust removal and minimizing excess air, can significantly decrease industrial air pollution.

New Delhi

Major Hotspot: Chanakyapuri, Rashtrapati Bhawan, India Gate.

- The installation of a diesel particulate filter can greatly diminish emissions from diesel vehicles.
- Water sprinkling on roads.
- Use of smog guns on a regular basis.
- Smog towers (Already present but not operational).
- Encourage the use of hydroelectric vehicles.
- Road traffic management.
- Maintenance of trees.
- Control on crop stubble burning by government.
- Incentives sowing of Kharif crops instead of paddy include straw as input in coal power plants.

North West

Major Hotspot: Rohini, Jahangirpuri, Bawan, Wazirpur, Narela, Mukherjee NagarGAR, Model Town.

- Control on crop stubble burning by government.
- Incentives sowing of Kharif crops instead of paddy, include straw as input in coal plants.
- Controlling vehicular pollution.

  Following steps can be undertaken by government:
  Import low sulphur petrol and diesel, accurate conversion of public sector refineries to low sulphur, introduce pollution cess on high sulphur fuel, double metro capacity, funding fuel efficient two wheelers.

Advance emission surveillance of on road vehicles with remote sensing devices.

- Clean fuels and electric mobility.
- Proper road traffic management.
- Control on solid waste burning:

  Setting up new waste to energy plants with the latest technology.
  Segregation of waste.

Maintain and ensure 100% capacity utilization of all plants in affected and surrounding areas.

- Encourage widespread adoption of solar panels in homes, buildings and businesses with subsidies to ensure financial viability, aiming to replace coal-based thermal power plants and mitigate Delhi's air pollution and climate impact.
- Control on soil and road dust:

  Repair roads where bitumen is loose or there are potholes.
  Ensure quick completion of digging work.
  Convert all roads to surfaced category.

- Monitoring and check on emissions from industries 24 × 7.
- Use wet scrubbers and electric precipitators in all coal using industrial plants.
- Changes in coal and thermal plants:
  Use superior quality coal for plants, subsidize gas, stop small thermal plants and increase capacity of gas-based power plants.

Central Delhi

Major Hotspot: Burari, Paharganj, Connaught Place, Chandani Chowk.

- Cleaning roads near transportation hubs reduces dust pollution.
- Use of smog guns on a regular basis.
- Smog towers (Already present but not operational).
- Waste management.
- Promote hydroelectric vehicles alongside increasing electric, hybrid and BS-VI vehicle adoption, and enhance public transportation for comprehensive pollution control efforts.
- Water sprinkling.
- The installation of a diesel particulate filter can greatly diminish emissions from diesel vehicles.
- Repair roads where there are potholes.
- Maintenance of trees.
- Control on solid waste burning:
- Setting up new waste to energy plants with the latest technology.
- Segregation of waste.
- Maintain and ensure 100% capacity utilization of all plants in affected and surrounding areas.
- Control on solid waste burning:

Setting up new waste to energy plants with the latest technology, segregation of waste, maintain and ensure 100% capacity utilization of all plants in affected and surrounding areas.

West Delhi

Major Hotspot: Mundka, Punjabi Bagh, Hari Nagar, Uttam Nagar, Janakpuri, Pashchim Vihar, Mayapuri, Shadipur.

- Cleaning roads near transportation hubs reduces dust pollution.
- Construction sites must be covered.

- Water sprinkling.
- Strict inspection of construction sites.
- Waste management system.
- Use of smog guns on a regular basis.
- Smog towers (Already present but not operational).
- Road traffic management.
- Monitoring emissions and maintenance 24 * 7.
- Repair roads where there are potholes.
- Abatement of air pollution caused by crop residue burning.
- Maintenance of trees.
- Control on crop stubble burning by government.
- Following steps can be followed:
- Incentives sowing of Kharif crops instead of paddy.
- Include straw as input in coal power plants.
- Control on soil and road dust.
- Repair roads where bitumen is loose or there are potholes.
- Ensure quick completion of digging work.
- Convert all roads to surfaced category.

South East Delhi

Major Hotspot: Okhla Phase 2, Defence Colony, Kalkaji, Lajpat Nagar.

- Water sprinkling.
- Use of smog guns on a regular basis.
- Smog towers (Already present but not operational).
- Encourage the use of hydroelectric vehicles.
- The installation of a diesel particulate filter can greatly diminish emissions from diesel vehicles.
- Road traffic management.
- Waste management system.
- Cleaning roads near transportation hubs reduces dust pollution.
- Regular check of use of chimneys by industries.
- Maintenance of trees.
- Control on crop stubble burning by government.

  Following steps can be followed:
  Incentives sowing of Kharif crops instead of paddy.
  Include straw as input in coal power plants.

- Use wet scrubbers and electric precipitators in all coal using industrial plants.
- Control on solid waste burning.
- Setting up new waste to energy plants with the latest technology.
- Segregation of waste.
- Maintain and ensure 100% capacity utilization of all plants in affected and surrounding areas.

5 Conclusion

AIR QUALITY INDEX can be tackled by implementing the solutions given by the expert system in terms of the number of trees and solutions.

Problem of unequal distribution of trees can be optimally solved by planting the trees in which the space is available.

Places where there is no space for plantations in industrial areas, our expert system has suggested ways other than trees like liquid trees, water sprinkling, methods for maintenance of trees, artificial rain and smog towers.

Expert system will suggest methods specific to districts and their problems.

Expert system is suggesting an approximate number of trees that can be planted in the particular bare land.

References

1. Chakravarty A, Siva Sathya S, Sharmiladevi S, An exploratory analysis of Delhi air quality using statistics and machine learning models. In: Conference on interdisciplinary approaches in technology and management for social innovation (IATMSI)
2. Dhar A, Dey S, Sarkar S (2023) Liquid trees: a novel approach for air pollution mitigation. Vigyan Varta 4(11):193–196
3. Manuel C (2020) The Miyawaki method—data and concepts. Urban Forests Company
4. Aarthi R, Sumaiya Begum A, Yaswanthshahi SK, Avinashreddy Y, Sharfuddin SK (2021) Miyawaki forest automation and unauthorized restrictions
5. Devblan I, Yadav A, Green hydrogen from green electricity. In: Advances in science and engineering technology international conferences (ASET), India
6. Sharma S, Kumar A, Jain P, Green hydrogen in India-applications, opportunities and challenges. In: 1st international conference on intelligent computing and research trends (ICRT)
7. Patil DS, Patil Y, Kirange Y (2023) Study on growth of electric vehicles in India: a review. In: 3rd Asian conference on innovation in technology (ASIANCON) Pune, India
8. Hong L, Lianhong L, Jie X, Juan Z (2012) New conception of air pollution control in China. In: 2012 Third international conference on digital manufacturing and automation

Sign Languages and AI: A Survey of The Impact of Artificial Intelligence on Deafhood

Niharika Trivedi, Yashaswini Trivedi, Rushik Parmar, and Prithwijit Guha

Abstract Sign languages are not only a crucial mode of communication for the deaf community, but are also an integral part of the deaf culture. The integration of sign languages and artificial intelligence (AI)—a technological domain that has seen rapid advancements in recent years—has been a pivotal step in the process of making the world more inclusive and accessible for the deaf. This article provides a comprehensive analysis of the AI technologies being utilized for natural language processing, machine learning, and gesture recognition while exploring the ethical ramifications of the usage of AI in linguistic context.

Keywords Artificial intelligence (AI) · Machine learning (ML) · American Sign Language (ASL) · Child of Deaf Adult(s) (CODA)

1 Introduction

According to the World Health Organization (WHO), several factors can cause hearing loss and natural hearing loss can occur at any point during a person's lifetime: be it the prenatal period, perinatal period, childhood, adolescence, adulthood, or old age. Other outside factors such as trauma to the head or ears, ototoxic medicines

N. Trivedi (✉)
UC San Diego, San Diego, CA, USA
e-mail: nitrivedi@ucsd.edu

Y. Trivedi
Disha Delhi Public School in Kota, Kota, Rajasthan, India

R. Parmar · P. Guha
Indian Institute of Technology, Guwahati, Assam 781039, India
e-mail: pguha@iitg.ac.in

P. Guha
e-mail: rushik_parmar@iitg.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_20

and chemicals, loud noises, and viral infections can also cause hearing loss [1]. It is estimated that over 430 million people around the globe suffer from disabling hearing loss, and the count is predicted to grow upto 700 million people by 2050 [5]. Thus, it is imperative that we understand the requirement of a system that supports such a sizeable population of the world and form a collective understanding of sign languages from a cultural as well as scientific point of view.

There are multiple sign languages found all around the world, and while all of them differ in some form or another, the one aspect that remains consistent across all geographically separated sign languages is that they are an integral part of the deaf communities.

Ideally, the deaf community as a whole should feel included, not alienated. However, due to the fast-paced lifestyle this age's rapid advancements (technological and otherwise) have created, having people mandatorily learn a selection of sign languages is neither a viable solution nor is it one which is fully inclusive. Moreover, an undeniable fact is that for any community, the native language is a precious heirloo—development of a brand new, universal sign language to be taught to everyone in schools and universities would threaten to erase the existence of one of the key features of the deaf culture.

To begin the formulation of a practical solution we must lean on technology—more specifically, artificial intelligence and machine learning (AI/ML). Incorporation of AI/ML into sign language recognition and replication has been a critical step toward making communication between deaf people and hearing people—as well as between deaf communities from different regions—smoother, thus making the world a kinder and more cohesive place.

2 Deafhood, Sign Language, Deafblindness, and CODA

In order to delve deeper into the idea of artificial intelligence as a tool for the study of linguistic domains like sign languages, we must first look at the cultural and social standing of the deaf community around the world.

2.1 Deafhood

Any discourse on sign languages would be incomplete without acknowledging the concept of deaf gain. This idea emerged from a redefinition of hearing loss, where "loss" implies a deficit, "deaf gain" promotes inclusivity. Rather than focusing solely on perceived deficiencies, deaf gain emphasizes the unique strengths, experiences, and cultural contributions of deaf individuals and communities. By reframing the narrative, deaf gain celebrates the diversity of deaf culture, highlighting the valuable insights derived from sign languages and navigating the world without relying on sound. This perspective encourages broader recognition and appreciation of the

many ways in which deaf people enrich society, challenging misconceptions, and promoting equality and inclusion for all.

2.2 Sign Language

The term "sign language" shares similarities with spoken language in that both are prevalent across various regions globally [6]. Just like spoken language, sign language undergoes a gradual evolution in its grammar and vocabulary over an extended period, establishing itself as a legitimate language. Sign language serves as a universal means of communication among the deaf, as it requires no auditory perception for comprehension and no vocalization for expression. Sign languages [7] typically involve the coordinated use of hand shapes, orientations, hand and body movements, and facial expressions to effectively convey a speaker's thoughts.

Sign languages are classified as natural languages and have the proper linguistic components other spoken languages do: sublexical components like phonemes and morphemes as well as lexical structure, grammar, and syntax [2, 23]. The advent of sign language as a concept could be traced back to merchants using signs as a way to communicate in busy marketplaces, or to the Middle Ages priests, who would take (either permanent or temporary) vows of silence and used monastic sign languages and fingerspelling [3, 24]. We have written records dating as far back as the fifth century BCE with references to a sign language: in Plato's Cratylus, Socrates says, "If we hadn't a voice or a tongue, and wanted to express things to one another, wouldn't we try to make signs by moving our hands, head, and the rest of our body, just as dumb people do at present?" [4, 25].

2.3 Deafblindness

Deafblindness [8] refers to a condition in which an individual has both significant vision and hearing impairments. People who are deafblind may experience varying degrees of vision and hearing loss, and the combination of these impairments can significantly impact communication, mobility, and daily living. Deafblind individuals often rely on alternative methods of communication, such as tactile sign language, braille, or other tactile and auditory modes. A technique of particular interest, known as the "Deafblind Tactile Fingerspelling Alphabet" in Australia or the "Deafblind Manual Alphabet" in the UK, involves twenty-six tactile symbols that correspond to the letters of the Latin alphabet. Figure 1 illustrates different gestures associated with each English alphabet, as represented in the Australian deafblind manual.

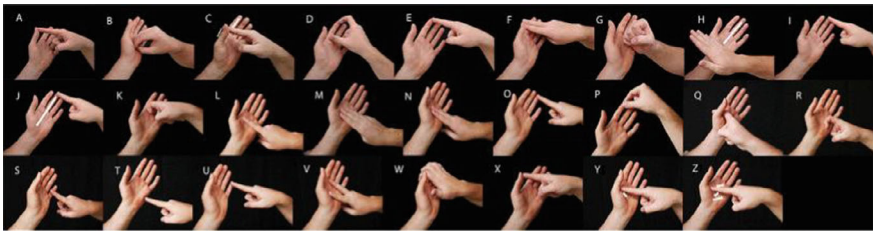


Fig. 1 Letters of the Australian deafblind manual alphabet [8]

2.4 CODA:Child of Deaf Adult(s)

CODA refers to individuals with at least one deaf parent. They may or may not be deaf themselves but often grow up exposed to and familiar with deaf culture and American Sign Language (ASL). These individuals can play a crucial role in nurturing understanding and connection between the deaf and hearing communities, both of which predominantly utilize visual-spatial language as a means of communication.

American Sign Language or ASL is one of the common visual-spatial languages primarily used by deaf communities. It uses facial expressions, body postures, and gestures combined with manual signs to communicate. Figure 2 illustrates ASL gestures associated with each letter and number. Here's a brief overview of ASL:

- The Gallaudet Dictionary of American Sign Language is a comprehensive reference tool created by an expert panel led by Clayton Valli, containing more than 3000 ASL sign illustrations and a full-color DVD featuring native ASL signers demonstrating each sign [9].
- The ASL-LEX 2.0 Project is a publicly available, large-scale lexical database for ASL that includes 2723 ASL signs, providing details on phonological descriptions, frequencies, iconicity, transparency, and more [10].
- Research into ASL recognition often involves extracting features from hand poses, such as distances between joint points and angles between vectors and axes, and then utilizing classifiers like support vector machines (SVM) and light gradient boosting machine (GBM) to classify characters [11, 12].
- Large-scale multimodal datasets, such as How2Sign, consist of sign language videos along with accompanying data like speech, English transcripts, and depth information, aiming to facilitate advancements in sign language recognition, translation, and production[13].

3 AI- and ML-Based Sign Language Recognition

AI refers to the ability of machines to mimic human cognitive functions, such as learning, problem-solving, and decision-making, typically through computer sys-

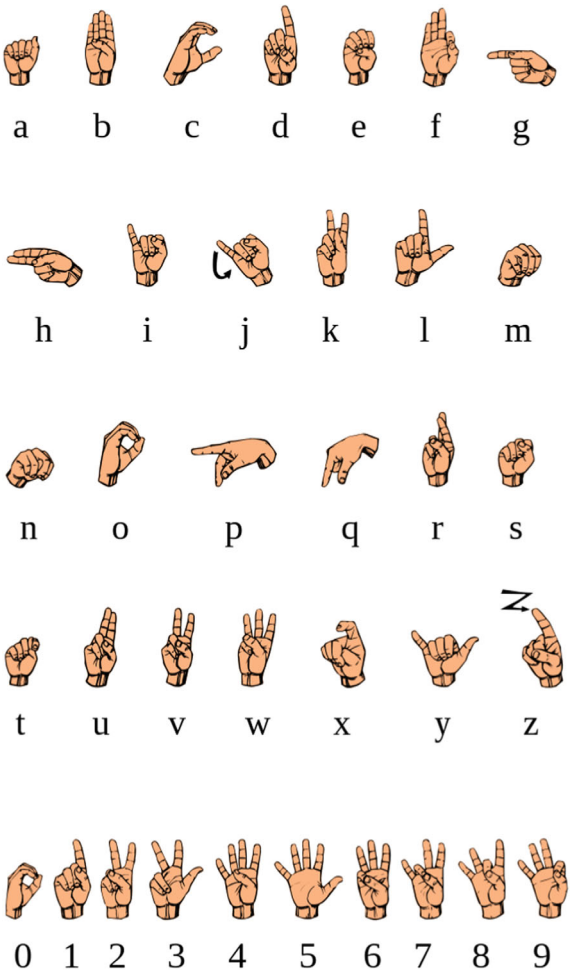


Fig. 2 American Sign Language [17]

tems. AI techniques empower machines to execute tasks that traditionally demand human intelligence, including activities like visual perception, speech recognition, decision-making, and language translation. In sign language recognition, AI systems are designed to understand and interpret hand gestures and body language.

ML is a subfield of AI that allows machines to learn and improve without explicit programming. In this context, ML algorithms are trained on vast datasets of sign language videos and images, enabling them to recognize different signs and translate them into spoken language or text.

Benefits of AI and ML in Sign Language Recognition:

- Accessibility: AI and ML can create real-time sign language translation tools, enabling seamless communication in various situations, like education, employment, and healthcare. Inclusivity: These technologies can foster social inclusion

by facilitating communication between deaf and hearing individuals, promoting a more inclusive society.
- Accuracy and Speed: With ongoing research and development, AI and ML models are constantly learning and improving, leading to higher accuracy and faster sign language recognition.
- Real-Time Recognition: AI techniques enable real-time processing of sign language gestures, facilitating instant communication between individuals using sign language and those who understand it.
- Cost-Effectiveness: Vision-based approaches are often more cost-effective than sensor-based methods, making them more accessible for widespread use in sign language recognition systems.

In the context of sign language recognition using vision-based and deep learning techniques, AI and ML play crucial roles:

- Vision-Based Techniques: Vision-based approaches utilize cameras or sensors to capture hand gestures or movements, which are then processed to recognize sign language. These techniques can extract features from images or videos to interpret sign language gestures accurately [14].
- Deep Learning: Deep learning, a subset of ML, involves training artificial neural networks with large amounts of data to recognize patterns and make decisions. Deep learning models, such as convolutional neural networks (CNNs), have shown significant success in tasks like image recognition, making them valuable for sign language recognition [12].

4 Approaches for AI/ML-Based Sign Language Recognition

Numerous studies have explored sign language recognition through AI/ML methodologies. The predominant strategy involves obtaining data from images or videos, preprocessing the data, extracting meaningful features, and employing a mathematical model for classification, as depicted in Fig. 3. This methodology can be broken down further based on the specific steps in the classification process, as detailed below.

1. Appearance-based SLR: This approach uses 2D images or videos as input and extracts features such as hand shape, orientation, motion, and facial expression [15, 16]. It relies on various methods such as skin color segmentation, edge detection, Fourier descriptors, and template matching. It also uses machine learning classifiers such as HMM, SVM, and RF to recognize signs.
2. Deep learning-based SLR: This approach uses advanced neural network architectures such as 3D CNN, C3D, 3D ResNet, and Dilated CNN to process spatiotemporal information from sign language videos. It also uses hybrid models such as

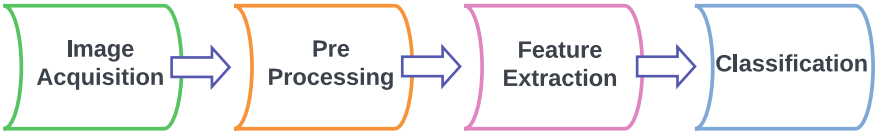


Fig. 3 Classification model

Table 1 Comparison between state-of-the-art methods

Work	Dataset/Image acquisition	Pre-processing	Model	Accuracy
[18]	Youtube, Webcam	AlexNet	Faster R–CNN (CNN-LSTM)	42.00
[19]	ILSVRC dataset	VGG16 model	Deep CNN	96.00
[20]	RWTH-PHOENIX -Weather 2014	VGG-S/ GoogLeNet	CNN-BLSTM	38.70
[21]	RWTH-PHOENIX -Weather	3D-ResNet	CNN & Dilated Convolutions	37.30
[22]	Chinese sign language (CSL)	C3D model	(HLSTM)& 3D CNN	92.40

CNN-HMM, CNN-BLSTM, and CNN-LSTM to handle sequential and contextual information. It employs methods like transfer learning, data augmentation, and attention mechanisms to enhance the models' performance and resilience [18–21].

The comprehensive summary of various works in the literature, utilizing diverse datasets and employing distinct classification processes, is presented in Table 1.

5 Ethical Ramifications

AI-related technology has the potential to be integrated into almost every field of work. However, the fact that it also has the capacity to be ethically irresponsible must always be taken into consideration. In the case of linguistics and sign languages, the list of ethical ramifications is substantial.

1. Representational accuracy: Errors in translation accuracy could result in misrepresentation, confusion, and misinterpretation, potentially affecting individuals' ability to access information and services.
2. Algorithmic bias: The behaviors of the AI systems are reliant on the data they are trained on or the design choices made during their development. Such algorithmic

bias can lead to inaccurate or discriminatory outcomes in language understanding, generation, translation, or sentiment analysis tasks.

3. Threat to inclusivity: People cannot and should not be divided into the binary of "the hearing" and "the deaf." There exists a whole spectrum of deafness. Usage of AI might create barriers or exclude certain groups within the deaf community.
4. Risk of cultural appropriation: Sign languages exhibit cultural subtleties and regional distinctions. It runs the risk of inappropriate adoption of deaf culture.
5. Overpowering the workforce and lack of individual progress: AI should aid, not replace, human sign language interpreters and educators. Over-reliance on AI could diminish job opportunities and may prevent the linguistic interactions and growth of the deaf community. This may result in a lack of meaningful personal development for many.

These consequences can be regulated (or at least minimized) if the developers are conscientious. The development stage must include rigorous testing, and the algorithm must be continuously updated with inclusive and diverse datasets. Feedback from the deaf community must be taken into consideration to ensure respectful representation. AI must be understood as a tool being leveraged to supplement educational resources and facilitate linguistic interactions. Such precautions are necessary to ensure that the AI applications align with cultural values and norms of the deaf community.

6 Future of AI and Sign Language Recognition

The future of AI and sign language recognition holds promising developments, driven by ongoing research and technological advancements. Several trends and potential directions can be anticipated:

- Enhanced Accuracy and Robustness: Continued improvements in deep learning models and algorithms are likely to lead to higher accuracy and robustness in sign language recognition. This includes better handling of variations in signing styles, lighting conditions, and background noise.
- Multimodal Approaches: Future systems may integrate multiple modalities, combining vision-based recognition with other sensors like depth sensors or accelerometers. This approach could provide a more comprehensive understanding of sign language gestures and enhance recognition accuracy.
- Gesture Understanding Beyond Signs: AI may evolve to recognize not only individual signs but also the nuances of facial expressions, body movements, and non-manual markers used in sign languages. This broader context understanding can improve the overall interpretation of signed communication.
- Customization and Personalization: AI systems could become more customizable and adaptive to individual users. Personalized sign language recognition models may take into account the user's signing style, preferences, and unique gestures, making communication more tailored to each person.

- Edge Computing for Real-time Processing: Advances in edge computing may lead to on-device processing, enabling real-time sign language recognition without relying heavily on cloud-based services. This can enhance privacy, reduce latency, and make the technology more accessible in various settings.
- Incorporating Augmented Reality (AR) and Virtual Reality (VR): Combining sign language recognition with AR and VR technologies has the potential to offer users immersive experiences. This could be especially beneficial for language learning, remote communication, or interpreting services.
- Global Standardization: Efforts may be made toward standardizing sign language datasets and recognition models on a global scale. This could facilitate collaboration, benchmarking, and the development of universally compatible sign language recognition systems.
- Inclusive Design and Accessibility: There could be a growing emphasis on designing AI systems with inclusivity in mind. Ensuring accessibility for individuals with diverse signing styles, abilities, and cultural backgrounds will be a key consideration in the development of future sign language recognition technologies.
- Human-AI Collaboration: Future systems may focus on enhancing the collaboration between humans and AI in sign language interpretation. This could involve the development of tools that support human sign language interpreters or provide assistance in specific contexts.
- Ethical and Responsible AI: As with any AI application, ensuring ethical use, addressing biases, and respecting privacy in sign language recognition systems will become increasingly important. Research and development efforts will likely prioritize responsible AI practices.

In summary, the future of AI and sign language recognition holds exciting possibilities for improved accuracy, multimodal approaches, personalization, real-time processing, integration with emerging technologies, global standardization, inclusive design, human-AI collaboration, and a focus on ethical considerations. These developments hold the promise of substantially improving communication and accessibility for individuals in the deaf and hard-of-hearing community.

7 Conclusion

This article offers an in-depth examination of the AI technologies applied in natural language processing, machine learning, and gesture recognition. It delves into the ethical implications of utilizing AI in linguistic contexts. Additionally, it explores the cultural and social dimensions of sign languages and the deaf community. The article addresses challenges and opportunities associated with AI and ML techniques in sign language recognition. It also anticipates future trends and directions in AI and sign language recognition, underscoring potential benefits and risks for communication and accessibility within the deaf and hard-of-hearing community.

References

1. Deafness and hearing loss https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss
2. Dumbre A, Jangada S, Gosavi S, Gupta J (2022) Classification of Indian sign language characters utilizing convolutional neural networks and transfer learning models with different image processing techniques. In: 2022 IEEE world conference on applied intelligence and computing (AIC), Sonbhadra, India, pp 423–430. https://doi.org/10.1109/AIC55036.2022.9848930
3. History of sign language (2024) In Wikipedia. https://en.wikipedia.org/wiki/History_of_sign_language
4. Auslan: now and then-history of sign language. https://history.waad.org.au/history-of-sign-language
5. Estimating the worldwide burden of health loss due to hearing loss. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9897997/
6. Kishore PVV, Kumar PR (2012) A video based Indian Sign Language Recognition System (INSLR) using wavelet transform and fuzzy logic. Int J Eng Technol 4(5):537
7. ChaLearn Looking at People (2014). https://gesture.chalearn.org/2014-looking-at-people-challenge
8. Duvernoy B et al (2023) HaptiComm: a touch-mediated communication device for deaf-blind individuals. IEEE Robot Autom Lett 8(4):2014–2021. https://doi.org/10.1109/LRA.2023.3241758
9. Valli C (2021) The Gallaudet dictionary of American Sign Language
10. Sehyr ZS, Caselli N, Cohen-Goldberg AM, Emmorey K (2021) The ASL-LEX 2.0 Project: a database of lexical and phonological properties for 2723 signs in American Sign Language. J Deaf Stud Deaf Educ 26(2):263–277. https://doi.org/10.1093/deafed/enaa038. PMID: 33598676; PMCID: PMC7977685
11. Shin J, Matsuoka A, Hasan MAM, Srizon AY (2021) American Sign Language alphabet recognition by extracting feature from hand pose estimation. Sensors (Basel) 21(17):5856. https://doi.org/10.3390/s21175856. PMID: 34502747, PMCID: PMC8434249
12. Shikhar S, Krishan K (2021) ASL-3DCNN: American Sign Language recognition technique using 3-D convolutional neural networks. Multimedia Tools Appl 80:26319–26331
13. Duarte A et al (2021) How2sign: a large-scale multimodal dataset for continuous American Sign Language. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition
14. Shin J, Matsuoka A, Hasan MAM, Srizon AY (2021) American Sign Language alphabet recognition by extracting feature from hand pose estimation. Sensors (Basel) 21(17):5856. https://doi.org/10.3390/s21175856. PMID: 34502747; PMCID: PMC8434249
15. Camgoz NC, Hadfield S, Koller O, Bowden R (2017) SubUNets: end-to-end hand shape and continuous sign language recognition. In: 2017 IEEE international conference on computer vision (ICCV). Venice, pp 3075–3084
16. Kika A, Koni A (2018) Hand gesture recognition using convolutional neural network and histogram of oriented gradients features. In: CEUR workshop proceedings, vol 2280, CEUR-WS, pp 75–79
17. American Sign Language (2024) In Wikipedia. https://en.wikipedia.org/wiki/American_Sign_Language
18. Shi B (2018) American Sign Language fingerspelling recognition in the wild. In: IEEE spoken language technology workshop (SLT). Athens, Greece, 145–152
19. Masood S, Thuwal HC, Srivastava A, Satapathy S, Bhateja V, Das S (2018) American sign language character recognition using convolution neural network. In: Smart computing and informatics. Smart innovation systems and technologies, vol 78. Springer, Singapore
20. Cui R, Liu H, Zhang C (2017) Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In: 2017 IEEE conference on computer vision and pattern recognition. CVPR, Honolulu, HI, pp 1610–1618

21. Pu J, Zhou W, Li H (2018) Dilated convolutional network with iterative optimization for continuous sign language recognition. In: International joint conference on artificial intelligence, IJCAI, pp 885–891
22. Guo D, Zhou W, Li H, Wang M (2018) Hierarchical LSTM for sign language translation. In: AAAI conference on artificial intelligence, North America
23. Lillo-Martin D, Henner J (2020) Acquisition of Sign Languages. Annu Rev Linguist 395–419. https://doi.org/10.1146/annurev-linguistics-043020-092357. Epub 2020 Oct 12. PMID: 34746335, PMCID: PMC8570554
24. Bruce Scott G (2001) The origins of Cistercian Sign Language. Cîteaux 52(3/4):193–209
25. Plato's Cratylus (Stanford Encyclopedia of Philosophy) (2024). https://plato.stanford.edu/entries/plato-cratylus/

Cosine Similarity Preserving Curse of Dimensionality Reduction for Managing Computational Complexity

Tefo Kgosietsile

Abstract High dimensional data in information retrieval and text mining is increasingly in demand. Information retrieval mechanisms in vector databases are performing well in managing high dimensional data. However, data sparsity remains a significant reoccurring issue forcing the data field to resort to traditional data retrieval, which eventually results in an increased curse of dimensionality. An elevated level of the curse of dimensionality eventually produces models that suffer from computational complexity. Data sparsity, being the root cause of the problem, occurs when data points are farther apart and therefore difficult to measure their similarity. It is on this premise that most conversational platforms are suffering from data hallucinations and unjustified responses throughout the search retrieval process. The curse of dimensionality has also been reoccurring in most high dimensional data retrieval systems despite the quite impressive performance of vectors. The prevalence of exponential increase in dimensions poses processing as well as analysis challenges which equates to computational complexity. To address this, reduction mechanisms are the core possible solutions. This research investigates the effectiveness of similarity metrics to aid dimensionality reduction, all in pursuit of managing computational complexity. We therefore designed a test plan to assess the suitability of various similarity metrics and compared them against core features. A total of 8 similarity metrics against 5 core features were evaluated through a matrix table, and the results indicated that cosine similarity is most suitable for solving the dimensionality issues and lower computational complexity.

Keywords Cosine similarity · Computational complexity · Curse of dimensionality · Similarity metrics

T. Kgosietsile (✉)
University of Botswana, Gaborone, Botswana
e-mail: tefokgosietsile@yahoo.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_21

1 Introduction

Computational complexity can be solved through dimensionality reduction, and similarity metrics employ powerful mechanisms to solve complexity issues. The process of filtering irrelevant data, clustering similar data together, approximating the required computations, and reducing dimensions without losing the original data manages algorithm complexity. Optimized solutions are vital for intelligent decision-making [1] and allow models to learn and improve performance [2]. The aim is to ensure robustness, efficiency, and improved performance of information retrieval systems in vector databases. Most methodologies and techniques used in dimensional indexing are flawed and need to be updated [3, 4]. Therefore, we need a versatile and efficient mechanism to oversee this issue. Large volumes of data are generated daily, and this data can be sparse, and unstructured with high features, which in turn decreases the efficiency of information retrieval and mining. High dimensional data has drastically increased [5], and this complicates machine learning analysis [6, 7], leading to increased complexity [8]. The efficiency and accuracy of data retrieval are mandatory to deal with these problems [9, 10]. However, transformations of this high dimensional to low may lose originality. Algorithms need to be designed such that they do not overcomplicate the existing problem but rather carefully eliminate the drawbacks [11]. The efficiency and effectiveness of data retrieval techniques must be experimentally validated [12, 13]. Constructing multiple data dimensions is an efficient management strategy; however, the problem lies with representations to classify them, traditionally vector embeddings performed well until they struggled to capture semantic relations of texts and evaluate similarity despite frequently reoccurring texts within data [14]. Therefore, context comparison is relevant [15]. As such, a similarity measure based on texts is needed.

The similarity between two vectors can be measured using the similarity metrics. This is because similarity metrics can count the frequency of words within a document. Vector representations manage different data structures. Therefore, the existence of the reoccurring curse of dimensionality which eventually leads to computational complexity may be solved through the application of similarity metrics. Selection of the right similarity metric involves key components to evaluate, assess, and determine its suitability. The type of data used, specific application, and complexity are some of the factors to consider when selecting the most appropriate metric to improve performance. As there are different metrics, each with advantages and disadvantages, the choice may depend on the purpose of the model as well. In short, it is impossible to find a one-size-fits-all metric [16], and detecting and clustering tasks that constantly switch can be difficult [17, 18].

This research aims to find the best one for dimensionality reduction to eventually help lower computational complexity. A curse of dimensionality occurs when the number of dimensions increases exponentially, or data points become sparse making it difficult to apply text mining during information retrieval. Increased curse of dimensionality results with increased data sparsity, high volumes of space, and additionally added algorithm features or algorithms overfitting training data, and all this

increases computational complexity. As a result, increased complexity brings inaccurate generalizations, unreasonable patterns, poor relationships, increased memory requirements, and more processing time [19, 20]. We argue that cosine similarity in information retrieval and text mining systems mitigates the curse of dimensionality and eventually reduces the computational complexity of programs.

Working with high dimensionality brings both advantages and disadvantages as retrieval systems use several processing computations [21]. This is because data has many features. Challenges involved include the curse of dimensionality and difficulty of data interpretation which may affect predictions [22] and cause overfitting due to learning algorithms failing to generalize to new predictions. Dimensionality reduction and combining prediction models are the two main strategies that may be employed to solve drawbacks brought about by high dimensions. Despite all the drawbacks, the importance of high dimensional data remains outstanding, as it provides accurate and informed decisions. In a situation where data becomes difficult to analyze because of its large dimensions, similarity metrics play a valuable role in measuring the closeness or farthest between data points. Similarity metrics offer several benefits to all these drawbacks. This study carefully investigates high dimensional data metrics to find how the best metrics can be applied to reduce the curse of dimensionality to lower computational complexity. Computational complexity is a theory of computation, and exploring it uncovers the effectiveness of problem-solving [8]. It further covers the number of resources required to solve a problem. It is a measure of problem complexity in terms of how many resources are needed to solve a task through an algorithm. For example, retrieving data is a task that needs resources, such as input data, query structure, output format, and processing the request [23]. Time complexity is measured using $O(n)$-Big notation, to mean the number of algorithm steps concerning input size growth. All this helps determine the solvability, intractability, or feasibility of algorithms.

2 Literature Review

The curse of dimensionality is the exponential increase of data dimensions as well as the computational efforts of processing and analyzing it as incurred by most datastore, search retrieval, or text mining mechanisms. The addition of dimensions throughout programming has a bearing on the complexity of algorithms; this equates to more features. A feature or attribute of an object represents a dimension, and a collection of dimensions forms data points. In vector databases, all this creates vectors defining data points. Vectors are mathematical objects which are more like scalable arrays in computer science. Their ability to scale with an increase in demand is what qualifies them as the best data containers for high dimensional data. They are characterized by magnitude and direction, where magnitude denotes length and size, and direction denotes their angle direction reference (θ). Dimensionality increase means more data points are needed for good performance and as such an algorithm exponentially increases making them more complex. Computational complexity occurs as more

combinations of data points are needed for high dimensional data. The mathematician Richard E Bellman coined the term "curse of dimensionality" to mean the exponential increase of data dimensions as well as its explosive nature [24].

The curse of dimensionality occurs when dealing with high dimensional vector representations of large documents, texts, images, and objects. In the case of vector databases, this manifests in the difficulty of distinguishing similarity between data points. This is a form of distance metrics problem that occurs in most used traditional metrics, such as Euclidean distance. Euclidean distance measures the distance between two points [16]. Data sparsity due to dimensional increase also causes more widespread data points. Most data retrieval depends on indexing and searching algorithms; however, they become inefficient in high vector representation. Algorithms become computationally complex as dimensionality increases. As data stores continue to manage various types of data, an efficient retrieval mechanism is mandatory, and dimensionality and data volume exponentially increase; thus, reduction mechanism of vectors is the only possible option if we are to get good performance, thus mitigating the curse of dimensionality. The idea is to implement distance metrics to measure the similarity distance between feature vectors that give accurate approximations of similarities [7]. This will then allow us to solve the drawbacks of dimensionality increase with the right similarity metric. The efficiency of information retrieval is a serious economic to web search engines [9], where many computers process queries. Computational efficiency and robustness to noise, with normalized data, play a vital role in complexity reduction [12] and argue that query evaluation techniques must be continually refined to improve their design and implementation. Large web search engines employ sophisticated techniques to ensure computational efficiency and scalability [21].

2.1 Curse of Dimensionality

When a feature in a dataset grows bigger than data points, the curse of dimensionality occurs. This brings alongside drawbacks, such as the model struggling to learn the relationship between objects, or grouping vectors together making it difficult to utilize space and find similar vectors which eventually leads to the difficulty of context similarity. All these problems pose a threat, forcing scientists to resort to traditional mechanisms. This makes context similarity and awareness difficult to perform. Overfitting and underfitting are common problems that occur in this case; as a result, pattern identification is impossible. Since the meaning of words in high dimensions are modeled as data points, when dimensionality increases then the volume of space also increases, and the data sparsity occurs as objects are too far apart and dissimilar, thereby preventing content similarity which is used in text retrieval. Semantic, sparse, and dense vector models are all affected by these problems. Large volumes of data with complex patterns make it difficult to identify associations between them. When data significantly deviate from the norm, complexity sets in and embeddings that

keep representing the meanings of words also struggle to capture semantic meanings. The dimensions in vectors correspond to word features. As a result, when there is a large attribute of words, equally vector dimensionality goes high as vectors are numeric representations of data points in embedding techniques, contrary to the traditional data store where rows and columns are used for search retrieval. It is on this premise that the curse of dimensionality increases and as such leads to the complexity of algorithms throughout request processing. This means dimensionality reduction eventually reduces computational complexity. A well-managed level of complexity offers benefits, such as high scalability, improved performance, and efficient data storage, all this in pursuit of improving the search retrieval process. High dimensional data systems are prone to the curse of dimensionality, models that overfit and underfit the training data, and leave very sparse, empty, or less data in dimensions.

The curse of dimensionality in similarity search mechanisms manifests when the number of objects within a dataset exponentially grows regardless of the underlying dimensionality. According to Bellman and Kalaba [2], these defeats dynamic optimization and complicates algorithms, especially in machine learning where the aim is to read the number of data objects in multidimensional vector space. A similarity search query performance in multidimensional indexes is dependent on the closeness of the query to its further neighbor, and as such objects are pushed to the boundaries of data space increasing dimensionality. Therefore, even retrieval of the closet queries involves searching through a large data space, thus making the search less efficient and complex even worse compared to sequential dataset search. Embedding data within vector representation was initially meant to lower dimensional space and lower computational complexity. As a result, a very robust dimensionality reduction technique is needed, and this is the reason cosine similarity is mandatory as a matter of reducing computational complexity. Contemplating the "nearest neighbor" problem by Beyer et al. [3] revealed that as the effect of dimensionality increases it triggers the distance between the nearest data point to approach the farthest data point. This means contrast in distance to different data points does not exist. However, this is not altogether a bad outcome as it may mean that data is perfectly correlated, uniformly distributed, or on the contrary wrong distance metric was applied. As a result, if this occurs then core tasks, such as clustering, pattern identification, and classification are rendered obsolete, which may force pre-processing or re-selection of the most appropriate distance metric. Thus, data sparsity, despite indexing, remains a major challenge in both information retrieval and storage.

Algorithms should be able to analyze data with several dimensions effectively. However, complexity increases as the dimensions significantly increase, and as a result, analyzing high dimension space may give abnormal results. This is the curse of dimensionality in information retrieval systems. In this case, more computational or processing efforts are needed to get the required outcome. Ironically, a model with comparatively higher dimensions is more accurate than those with low ones. It is for this reason that solving the problem needs careful attention. For example, if we have a dataset of sixty features and we intend to build 2 separate models from the same dataset. Models can be Algorithm 1 with 3 dimensions and Algorithm 2 with 2 dimensions since they read from the same dataset; this means Algorithm

1 has more information because its number of dimensions is slightly higher than Algorithm 2 and as such algorithm 1 is more accurate than Algorithm 2, sadly it is also the most susceptible to curse of dimensionality as they significantly increase. The greater the number of features, the higher the accuracy. However to a certain specific threshold, accuracy will no longer increase despite dimensionality increase. This is because now the model has been trained with a lot of data rendering it incompetent even when fed the correct data. Accuracy goes down regardless of increasing dimensions after a certain point, and it is in this case where the curse of dimensionality occurs. The curse of dimensionality increases memory requirements and influences efficiency and accuracy [10]. Dimensionality reduction has become a key component of pattern recognition in data mining. however, this brings drawbacks of data originality loss. It is possible to reduce data to smaller variables without major losses [5]. Dimensionality reduction techniques must be properly implemented without tampering with the quality of data.

2.2 Challenges of High Dimensional Data

Pattern finding in data with high numbers of features can be difficult. The increase in the dimensions requires more processing power for analysis. The model needs to be fed more training data. Accurately generalizing the model requires more data as more dimensions make more space. For understanding and improving data analysis, data can be clustered or grouping the related documents. Despite clustering techniques and pattern recognition, text mining remains a major problem. Mainly algorithms encounter scalability issues and inaccuracy due to high dimensions even the presence of bugs in the code [18, 25].

Some challenges and complications arise because of the exponential growth of dimensions, which may affect analyzing and organizing data spaces leading to the curse of dimensionality. The increase in features or attributes of data automatically increases data needed for generalizations. As such data becomes sparse, more computational resources are needed, the model fails to generalize to new patterns, distances lose meaning, and difficult data visualization, and all this eventually leads to decreased performance yet high computational complexity. Algorithms need to be improved to reduce random variables while keeping originality or transforming variables into principal components. According to Chen [6], the curse of dimensionality is an obstacle to dynamic optimization, and it renders applications complicated and affects query performance because the distances between the nearest and farthest data points are similar. As such reducing data to vectors in lower dimensions plays a vital role. High dimensional datasets pose mathematical challenges [19]. While simplifying complexity eases retrieval, indexing, classifying, and recognizing patterns are more manageable.

2.3 Dimensional Reduction

Dimensionality reduction is a technique that can be used to reduce problems caused by high dimensional data. This involves reducing the number of features within a dataset and still retaining information. In this case, it can be through the transformation of high to low-dimensional data space without losing the original data. Then complexity of the model gets reduced, simplifying the algorithms, and thereby model training and analysis become easier. Reduction leads to improved performance and reduces overfitting. Data visualization and pattern identification become easier. There are various ways to reduce dimensionality, such as principal component analysis, linear dissemination analysis, singular value decomposition, and t-distributed stochastic neighbor embedding. However, they also carry their weaknesses and strengths. As stated [22], this may involve pre-processing to remove redundant dimensions, and irrelevant data to improve accuracy and reduce the training resources.

3 Methodology

The commonality between data points can be quantified by mathematical functions called similarity metrics. The choice of selection depends on various factors, and a careful examination is mandatory before selecting the right one. To reduce dimensionality and computational complexity by employing appropriate similarity metrics, the following metrics are contemplated. We developed a test plan to assess the suitability of metrics for the proposed logic. Below is a test plan of 8 similarity metrics mapped to 5 core features (Table 1).

Computational Complexity Reduction Through Cosine Similarity

- Complexity occurs due to unnecessary features within datasets. As indicated in the matric Table 2, cosine similarity targets angles between two vectors. It identifies the relevant ones by ranking them according to priority. As a result, computational processing ignores those that contribute nothing, reducing requirements, such as time and memory; thus, complexity goes down.
- Cosine metric targets relative magnitude, as opposed to the absolute values, unlike other metrics; therefore, only high-priority relationships are retained reducing irrelevant representations while capturing original features. The computation of having to go through all the representations is managed specifically and precisely for the relevant search retrieval, reducing complexity.
- Cosine similarity as outlined in Table 3 is insensitive to scaling score; this means despite increasing dimensions, comparisons, and search of data points is less likely to be affected by noisy datasets. No additional computation is needed for managing irregularities due to sensitivity to scaling. This is one of the powerful features that outstands the cosine metric from the rest. For example, the Hamming Distance

Table 1 Test plan

Task	Description	Outcomes
Aims	To identify the relevant similarity metric to reduce dimensionality and computational complexity	
Objectives	Examine metric approach and application Compare metrics against expected features Select the right similarity metric Apply the metric to improve performance	
Test metrics	Hamming, Euclidean, Earth mover's distance (Wasserstein), Edit distance (Levenshtein), Mahalanobis distance, Pearson correlation coefficient, Cosine, and Jaccard similarity	
Test feature	Scaling score, Measure, Datatype, Target, Application	
Deliverables	Curse of dimensionality reduction, decreased complexity	
Weight score	Maximum = 1, minimum = 0	
Expected features	Relevant metrics applicable for high dimensional data, not sensitive to scaling score, measure vector distance, handle vector datatypes, target vector embeddings	

NB. The selected metric must have high support for expected features (Table 1)

Table 2 Test plan execution

	Scaling score	Measure	Target	Datatype	Application
Hamming distance	Affected by compared string length	String length	Strings comparison	Binary	Binary mutation, sequential
Euclidean distance	Magnitudes affect scores	Straight-line distance in two points	Difference of magnitude	General, numeric	Physical measure
Pearson correlation coefficient	Influenced by compared data scale	Linear relationships analysis	Linear relationships	Numbers	Pattern identification, statistical
Cosine similarity	Irresistible to vector magnitudes	The angle between two vectors	Directional metric	Vector representation	High dimensional text, image analysis
Jaccard similarity	Compared set size affect score	Similarity between item sets	Item sets comparison	Datasets	Document similarity
Earth mover's distance	Magnitude influenced	Distribution transformations	Data transformation distribution	Data represented as distributions	Distribution comparison
Edit distance	Magnitude insensitive	Edit operations	Transform character strings	Character strings	Character string compare
Mahalanobis distance	Dimensionless, scaling insensitive	Distance in two multivariate data	Compare multivariate	Multidimensional texts	Classify clusters, data points

is sensitive to scaling, which means comparing strings' actual length similarity affects the similarity score, though this is its main mandate, as such the difference inflates the length variance. Thus, the score it gives is a proportional mismatch rather than the actual length of comparing strings. Accounting for mismatches involves the additional complexity of normalizing the scores. Cosine offers the opposite.

Figure 1 shows a relation of how cosine leads to dimensionality reduction, which eventually leads to complexity reduction without losing the original data. The major categories of computational complexity involve time and memory requirements, which may include communication and input–output. As denoted in Fig. 1, there is a strong relationship between all these concepts: cosine similarity, dimensions, and complexity. Complexity can be influenced by resource allocations and design. In this case, cosine similarity is woven into algorithms to filter out irrelevant data that increases dimensions. It is embedded to cluster similar features together and ignore trivial ones, thus reducing memory requirements, and processing time due to fewer features to manage. It goes further to approximate complexity thus differentiating complex ones from the rest, so resource allocation is based on informed choice. This allows dimensionality reduction which then reduces the complexity of algorithms while achieving the required outputs. Computational efficiency, simplicity, and robust algorithms are attained due to versatile similarity metrics. The choice of metric selection should be based on rigorous investigation. Traditionally algorithms were struggling to give accurate predictions and patterns which led to hallucinations or incorrect outputs. Sometimes the mismatches are so tiny that the user cannot see them with the naked eye, hence depending on the wrong data for decision-making. Applying these techniques in fraud detection, classification, and prediction models improves performance. Algorithms need to employ the relevant metrics and be clear about their application and purpose to avoid complicating what would otherwise

Table 3 A similarity metric tested against a feature is given 1 point for good performance

	Scaling score	Measure	Target	Datatype	Application	Total
Hamming distance	0	0	1	0	0	1(20%)
Euclidean distance	0	0	0	1	0	1(20%)
Pearson correlation coefficient	0	0	0	0	1	1(20%)
Cosine similarity	1	1	1	1	1	5(100%)
Jaccard similarity	0	1	1	1	0	3(60%)
Earth mover's distance	0	0	1	1	0	2(40%)
Edit distance	1	0	0	0	0	1(20%)
Mahalanobis distance	1	1	1	1	0	4(80%)

There are 5 features, each rated at 20%, adding 100% for high-performing ones

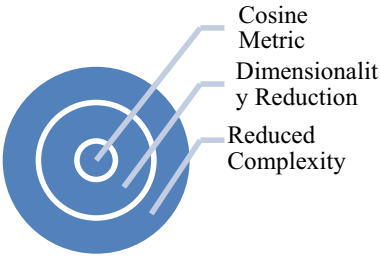


Fig. 1 Relation between concepts

be easier. Each has strengths and weaknesses, e.g., Mahalanobis metric proved to be robust to noise, and computationally expensive while Euclidean distance was found to be sensitive to the curse of dimensionality. Algorithms based on Euclidean struggle to learn from data as dimensions exponentially increase, so it becomes less useful in multidimensional data, as data points are farther apart thus failing to cluster them. So additional techniques were needed, such as combining it with the Manhattan Distance. Ironically, this brings more complexity, or one must apply reduction techniques before employing similarity measures, though there are still some drawbacks. The right technique with cross-cutting benefits across all these problems is mandatory for use.

Intractable programs fail with resource scheduling, and traditionally heuristics and parallel computing were used to speed up the search process. Occasionally, researchers use resource scheduling to assign priority to mitigate the exponential growth of datasets. Though heuristics and parallel computing reduce the burden, they are not optimal to the core complexity problems. For example, parallel computing ensures multiple processing, while heuristics divide the search space into halves each assigned to a different processor. Researchers debated this issue further with some stating that it is not the best solution as it carries additional drawbacks. As such, the way forward was to revert to combining the two approaches, to speed up search and retrieval of data. Parallel computing increases power consumption, hardware costs, communication overheads, and incorrect load balancing. Heuristics despite their quick performance are inaccurate and rely on assumptions, which leads to biases, limited adaptations to various domains, or unjustifiable outputs. Therefore, we propose applying cosine similarity to mitigate the curse of dimensionality, which eventually reduces the complexity.

3.1 How Cosine Matrices Reduce Curse of Dimensionality

It helps capture semantic similarity in high dimensional data, making it very important in grouping and clustering-related features, and cosine similarity identifies those that share the same meanings. Most search retrieval prioritizes meaning over individual words. As indicated in the above matrix tables, it targets direction as opposed

to magnitude between vectors. The addition of irrelevant features does not change the angle direction angle; therefore, the results are more accurate compared to other metrics. It is not affected by scaling compared to other metrics like Euclidean, and cosine metric measures the distance between two data points despite varying magnitude, making it suitable for high dimensional data points. As it ignores noisy and irrelevant features, it becomes efficient in managing data sparsity and this is particularly helpful in solving the curse of dimensionality. However, cosine similarity becomes less efficient when dealing with multidimensional data with random features and high probabilities. Cosine similarity remains the best metric for better accuracy and acceptable performance [26]. Measuring the similarity of text in documents is key to modern data retrieval problems, e.g., cosine similarity performs well in distance measuring while Euclidean distance fails to handle such issues [20]. This signified the importance of better choice in data mining [13]. As stated by [15], context comparison of tasks is as accurate as mining task descriptions and that makes it possible to predict relationships. Therefore, context similarity complements content similarity [1]. Attempting a task involves the need to know the required resources [4], context awareness recommenders are vital [23], and tasks influence behavior [25]. As developers are working on various tasks, switching between tasks is crucial but it carries the cognitive burden of memorizing a larger volume of data [17].

4 Results Analysis

See Table 3 and Fig. 2

5 Conclusion

Computational complexity can be managed differently, as it arises due to many reasons, such as exponential growth of data dimensions. As dimensionality increases data points become sparse and dissimilar, thus semantic search and retrieval becomes difficult. This can be measured by similarity metrics. It is vital to evaluate metrics to find the most suitable one. The relevant one should encompass among other features, support high dimensions, not sensitive scaling, able to measure vector distance and target vector embeddings (as in Table 3). Therefore, comparison similarity metrics indicate that the relevant one is the cosine metric. It accurately manages the curse of dimensionality compared to others (as shown in Fig. 2). This is considered the best for reducing computational complexity.

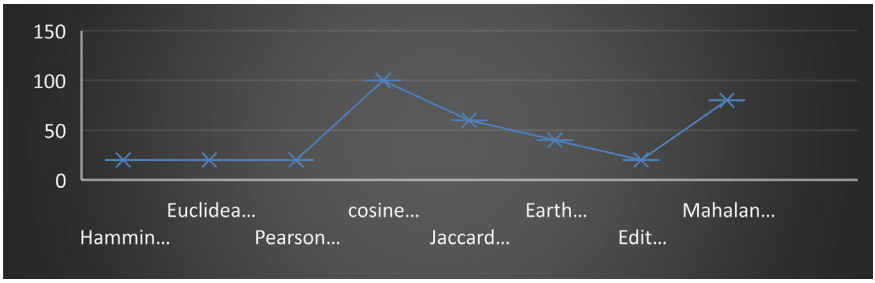


Fig. 2 Chart representing metrics analysis

References

1. Augusto JC (2022) Contexts and context-awareness revisited from an intelligent environments perspective. Appl Artif Intell 36(1). https://doi.org/10.1080/08839514.2021.2008644
2. Bellman R, Kalaba R (1959) On adaptive control processes. IRE Trans Autom Control 4(2):1–9
3. Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful? Database theory—ICDT'99: 7th international conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7, pp 217–235
4. Bontcheva K, Tablan V, Cunningham H (2014) Semantic search over documents and ontologies. In Ferro N (ed) Bridging between information retrieval and databases: PROMISE winter school 2013, Bressanone, Italy, February 4–8, 2013. Revised Tutorial Lectures. Springer, pp 31–53. https://doi.org/10.1007/978-3-642-54798-0_2
5. Chauhan D, Mathews R (2020) Review on dimensionality reduction techniques. In: Proceeding of the international conference on computer networks, big data and IoT (ICCBI—2019), pp 356–362.https://doi.org/10.1007/978-3-030-43192-1_41
6. Chen L (2009) Curse of dimensionality. In Liu L, Özsu MT (eds) Encyclopedia of database systems. Springer, US, pp 545–546. https://doi.org/10.1007/978-0-387-39940-9_133
7. Eğecioğlu Ö, Ferhatosmanoğlu H (2000) Dimensionality reduction and similarity computation by inner product approximations. In: Proceedings of the ninth international conference on information and knowledge management, pp 219–226
8. Gao Q, Xu X (2014) The analysis and research on computational complexity. In: The 26th Chinese control and decision conference (2014 CCDC), pp 3467–3472.https://doi.org/10.1109/CCDC.2014.6852777
9. Hawking D, Moffat A, Trotman A (2017) Efficiency in information retrieval: introduction to special issue. Inf Retrieval J 169–171. https://doi.org/10.1007/s10791-017-9309-7
10. Jia W, Sun M, Lian J, Hou S (2022) Feature dimensionality reduction: a review. Complex Intell Syst 8(3):2663–2693. https://doi.org/10.1007/s40747-021-00637-x
11. Kushwaha N, Pant M (2020) Textual data dimensionality reduction-a deep learning approach. Multimedia Tools Appl 79:11039–11050
12. Lin J, Trotman A (2017) The role of index compression in score-at-a-time query evaluation. Inf Retrieval J 20(3):199–220. https://doi.org/10.1007/s10791-016-9291-5
13. Liu L, Özsu MT (2009) Encyclopedia of database systems, vol 6. Springer, New York, NY, USA
14. Ma B, Zhuge H (2024) Automatic construction of classification dimensions by clustering texts based on common words. Expert Syst Appl 238:122292. https://doi.org/10.1016/j.eswa.2023.122292
15. Maalej W, Ellmann M, Robbes R (2017) Using contexts similarity to predict relationships between tasks. J Syst Softw 128:267–284. https://doi.org/10.1016/j.jss.2016.11.033

16. Malkauthekar MD (2013) Analysis of Euclidean distance and Manhattan distance measure in Face recognition. In: Third international conference on computational intelligence and information technology (CIIT 2013), pp 503–507
17. Meyer AN, Satterfield C, Züger M, Kevic K, Murphy GC, Zimmermann T, Fritz T (2022) Detecting developers' task switches and types. IEEE Trans Softw Eng 48(1):225–240. https://doi.org/10.1109/TSE.2020.2984086
18. Milenova BL, Campos MM (2002) O-Cluster: scalable clustering of large high dimensional data sets. In: 2002 IEEE international conference on data mining, 2002. Proceedings, pp 290–297. https://doi.org/10.1109/ICDM.2002.1183915
19. Shen HT (2009) Dimensionality reduction. In Liu L, Özsu MT (eds) Encyclopedia of database systems. Springer, US, pp 843–846.https://doi.org/10.1007/978-0-387-39940-9_551
20. Sohangir S, Wang D (2017) Improved sqrt-cosine similarity measurement. J Big Data 4(1):25. https://doi.org/10.1186/s40537-017-0083-6
21. Tolosa G, Feuerstein E, Becchetti L, Marchetti-Spaccamela A (2017) Performance improvements for search systems using an integrated cache of lists + intersections. Inf Retrieval J 20(3):172–198. https://doi.org/10.1007/s10791-017-9299-5
22. Velliangiri S, Alagumuthukrishnan S, Thankumar joseph SI (2019) A review of dimensionality reduction techniques for efficient computation. Procedia Comput Sci 165:104–111. https://doi.org/10.1016/j.procs.2020.01.079
23. Vo CC, Torabi T, Loke SW (2009) Towards context-aware task recommendation. Joint Conf Pervasive Comput (JCPC) 2009:289–292
24. Warwick K, Karny M (1997) Computer intensive methods in control and signal processing: the curse of dimensionality. Springer Science and Business Media
25. Ying AT, Robillard MP (2011) The influence of the task on programmer behaviour. In: 2011 IEEE 19th international conference on program comprehension, pp 31–40
26. Zeinali H, Mirian A, Sameti H, BabaAli B (2015) Non-speaker information reduction from Cosine Similarity Scoring in i-vector based speaker verification. Comput Electr Eng 48:226–238. https://doi.org/10.1016/j.compeleceng.2015.09.003

Comprehensive Mathematical Analysis and Derivation of Relationship Between Azimuth and Elevation Angles

Shivam Bhutiani, Param, and Gaurav Chugh

Abstract The use of azimuth and elevation angles is prevalent across various fields and applications, each leveraging these angles to address specific challenges and achieve precise outcomes. Some of the systems that require an intricate use of these two angles are navigation systems, satellite communication systems, remote sensing systems, optical systems, etc. Azimuth and elevation angles can be used together to precisely specify the direction or position of an object in three-dimensional space, especially in relation to an observer or reference point. In this comprehensive study, we delve into the determination of an intricate mathematical relationship between azimuth and elevation angles. Through complex, yet basic trigonometry and limits, an attempt has been made to meticulously derive a mathematical relationship between azimuth and elevation angles. Along with the mathematical derivation, we also aim to mathematically prove why azimuth angle tends to become undefined or indeterminate at the singularity point, that is, when the elevation angle approaches 90°.

Keywords Azimuth · Elevation · Trigonometry · Indeterminate

1 Introduction

In recent years, the exploration and utilization of azimuth and elevation angles have gathered substantial attention across various scientific and technological domains [1]. The accurate determination of the spatial relationship between azimuth and elevation plays a pivotal role in applications ranging from satellite communication to radar systems and geospatial mapping, etc. [2] This research work makes a rigorous examination, aiming to contribute to a comprehensive mathematical analysis and derivation

S. Bhutiani (✉)
SRM University—AP, Guntur, India
e-mail: shivambhutiani.py@gmail.com

Param · G. Chugh
Space Applications Centre, Indian Space Research Organization, Ahmedabad, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_23

of the intricate relationship between azimuth and elevation angles specifically in the Cartesian plane and spherical planes.

The motivation behind this mathematical study arises from the increasing demand for precise spatial orientation and directional control in various technological and scientific applications that heavily depend on azimuth and elevation angles. Therefore, there is always an ever-growing need for an in-depth mathematical analysis that not only provides practical insights for optimizing systems and processes reliant on azimuth and elevation information but also gives us a solid mathematical formula/relationship between the two which still does not exist in the way as derived in this paper.

Basically, azimuth is a fundamental angular measurement used in navigation and celestial mechanics to describe the horizontal direction of an object or point in the celestial sphere. Measured clockwise from the north, azimuth spans 360°, provides a complete circle of directional reference [3]. In practical applications, azimuth is employed to navigate and establish the horizontal position of objects, serving as a cornerstone in fields such as surveying, radar systems, and satellite communication. Elevation, on the other hand, is a vertical angular measurement that indicates the height or altitude of an object above a specified reference plane. Measured in degrees, elevation angles range from 0° at horizon to 90° directly overhead. This parameter is essential in various fields, including astronomy, satellite communication, etc. Elevation angle complements azimuth angle by providing the vertical perspective, collectively offering a three-dimensional description of an object's position in space [3]. The combination of azimuth and elevation is integral for tasks such as pointing antennas, optical telescopes, and tracking celestial bodies. The impact of measurement errors on the determination of azimuth and elevation angles cannot be overlooked. By quantifying and addressing these uncertainties, we can gain a more realistic understanding of the practical implications of the derived mathematical relationships. Incorporating strategies to mitigate measurement errors is essential for ensuring the accuracy and reliability of azimuth and elevation angle calculations in various applications.

Azimuth and elevation angles are fundamental parameters in geospatial mapping and navigation applications. In geodesy and cartography, azimuth angles are used to determine the direction of features relative to a reference point, facilitating the creation of accurate maps and spatial databases. Elevation angles, on the other hand, are crucial for terrain analysis, line-of-sight calculations, and the planning of infrastructure projects such as roads, buildings, and communication networks. In astronomy and celestial navigation, azimuth and elevation angles are indispensable for observing celestial objects and navigating spacecraft or telescopes. Astronomers use azimuth and elevation coordinates to track the movement of stars, planets, and other celestial bodies across the sky. Similarly, spacecraft rely on azimuth and elevation angles to orient their antennas and sensors toward communication satellites or planetary targets. Azimuth and elevation angles play a vital role in defense systems and surveillance applications, where precise targeting and monitoring are essential. Military radar systems utilize azimuth and elevation angles to detect and track incoming threats such as missiles, aircraft, and unmanned aerial vehicles.

Through this research work, we aim to enhance the understanding of these angles, their inter-relationship, and the condition where azimuth tends to be indeterminate when elevation approaches 90°, eventually providing a valuable resource for future research, engineers and practitioners grappling with the challenges posed by these angles in their respective domains.

2 Mathematical Derivation

The mathematical derivation of the relationship between azimuth and elevation involves Cartesian-spherical plane elements, i.e., the *XYZ* plane. Figures 1 and 2 are the pictorial illustrations of what we are working with. These two figures depict the azimuth angle (α) and elevation angle (β) as plotted along the *X*-axis, *Y*-axis, and *Z*-axis, respectively. In spherical coordinate system, azimuth is measured in the *XY*-plane, and elevation is measured from the *XY*-plane to *Z*-axis.

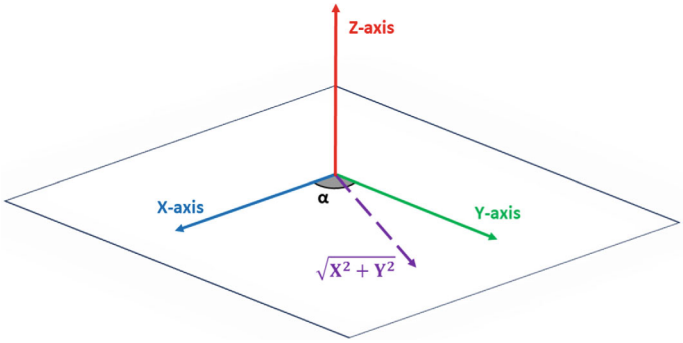


Fig. 1 Azimuth angle on *XYZ* plane

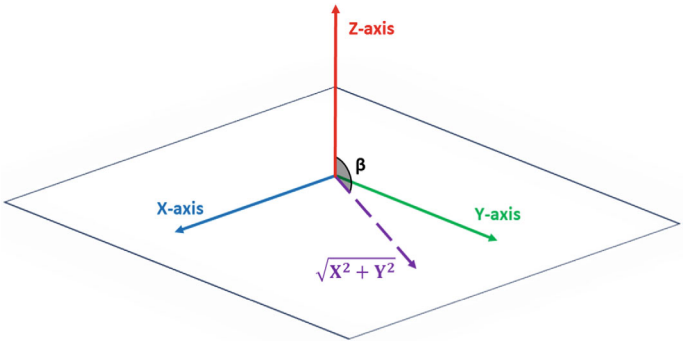


Fig. 2 Elevation angle on *XYZ* plane

As seen in the above figures, azimuth is the angle between X and Y-axis, elevation is the angle between Z-axis and the radial normalized axis formed on the XY-plane. The elevation angle is the angle above or below the horizontal plane, so we are interested in the ratio of the vertical distance to the radial distance in the XY-plane. In other words, the expression $\sqrt{X^2+Y^2}\sqrt{X^2+Y^2}$ is like the base of a right triangle formed in the XY-plane, and Z-axis is the height of that triangle. The following is the mathematical derivation of the relation between azimuth and elevation angle.

$$\tan\alpha = \frac{Y}{X} \tag{1}$$

$$Y = X.\tan\alpha \tag{2}$$

$$\tan\beta = \frac{Z}{\sqrt{X^2+Y^2}} \tag{3}$$

$$\tan\beta = \frac{Z}{\sqrt{X^2+(X.\tan\alpha)^2}} \tag{4}$$

$$\tan\beta = \frac{Z}{\sqrt{X^2+X^2.\tan^2\alpha}} \tag{5}$$

$$\tan\beta = \frac{Z}{\sqrt{X^2.(1+\tan^2\alpha)}} \tag{6}$$

$$\tan\beta = \frac{Z}{\sqrt{X^2.\sec^2\alpha}} \tag{7}$$

$$\tan\beta = \frac{Z}{X.\sec\alpha} \tag{8}$$

$$X.\tan\beta = Z.\cos\alpha \tag{9}$$

$$\beta = \tan^{-1}(\frac{Z.\cos\alpha}{X}) \tag{10}$$

The above Eqs. (9) and (10) are derived after multiple trigonometric and substitution operations on the equations obtained from the pictorial representation of azimuth and elevation angles. These two equations give us a straight relationship between azimuth and elevation angles.

3 Azimuth When Elevation Approaches 90°

The azimuth angle becomes indeterminate when the elevation angle approaches 90° (or − 90°) in certain coordinate systems, particularly those that use spherical coordinates. This indeterminacy is a consequence of the coordinate system and the singularity that occurs at the poles. In spherical coordinates, when the elevation angle is 90°, the azimuth point is at the pole, and any direction in the XY-plane is a valid azimuth.

The issue arises because the inverse tangent function used to calculate the azimuth becomes indeterminate at (± 90°) due to its periodic nature. The arctangent function cannot uniquely determine an angle when the ratio Y/X becomes infinite (as in the case when X approaches 0 at the pole). In other words, the azimuth angle becomes undefined or indeterminate. The following equations are a mathematical representation depicting the tendency of azimuth to become indeterminate when elevation approaches 90°.

$$\tan\alpha = \frac{Y}{X} \tag{11}$$

$$\alpha = \tan^{-1}\left(\frac{Y}{X}\right) \tag{12}$$

$$\tan\beta = \frac{Z}{\sqrt{X^2 + Y^2}} \tag{13}$$

$$\beta = \tan^{-1}\left(\frac{Z}{\sqrt{X^2 + Y^2}}\right) \tag{14}$$

$$\lim_{\beta \to 90^\circ} \frac{Z}{\sqrt{X^2 + Y^2}} = \text{undefined} \tag{15}$$

$$\lim_{\beta \to 90^\circ} \frac{Y}{X} = \text{undefined} \tag{16}$$

$$\tan\alpha = \text{undefined} \tag{17}$$

$$\alpha = \text{undefined} \tag{18}$$

The above mathematical derivation shows that azimuth becomes undefined or indeterminate when elevation angle approaches 90°. Now the question arises, if tan α becomes undefined, then α should be equal to 90° or 270°. When X becomes 0, the ratio Y/X indeed becomes undefined, but it does not necessarily imply that the azimuth is 90 or 270°. Instead, in the context of the inverse tangent function, the azimuth is usually considered as indeterminate.

In mathematical terms, as X approaches 0, the inverse tangent of Y/X approaches ($\pm$ 90°). However, the direction in the XY-plane is not uniquely determined in this case. The inverse tangent function itself cannot distinguish between the directions 90° and 270° because both directions have the same ratio of Y/X. Therefore, the azimuth angle is often considered indeterminate at the pole, and alternative coordinate systems or representations are used to avoid this issue. When working in coordinate systems other than Cartesian, it is essential to understand the conventions and definitions used for azimuth angles in those systems. For example, in spherical coordinates, the azimuth angle may be defined independently of the inclination or polar angle, allowing for a more continuous representation of directions in space. In cylindrical coordinates (ρ, θ, z), where ρ is the radial distance in the XY-plane, θ is the azimuthal angle (measured from the positive X-axis), and z is the height, the azimuth angle is already defined as θ. In the geographic coordinate system, azimuth angles are commonly used in navigation and geodesy. Azimuth in this system represents the compass direction measured clockwise from true north. To calculate azimuth angles in this system, one typically converts between latitude and longitude coordinates and then uses trigonometric formulas to calculate the azimuth angle between two points on the Earth's surface.

4 Empirical Validation and Efficiency

In this section, we present the empirical validation and efficiency of the derived mathematical relationships between azimuth and elevation angles. Empirical validation and efficiency are essential to verify the applicability and accuracy of the derived formulas in real-world scenarios.

The azimuth, elevation data were simulated, calculated, and collected from a space simulation tool by NASA, i.e., General Mission Analysis Tool (GMAT). GMAT is a powerful free-to-use tool that can be used to simulate powerful space missions and calculate related parameters like X, Y, Z position vectors, their respective velocity vectors, as well as complex parameters like azimuth, elevation, and range. The data collection process involves the determination of azimuth and elevation angle of a 600 km low-earth orbit (LEO) satellite from a ground station. The azimuth and elevations values that were obtained after simulations in GMAT are listed in Table 1.

Table 1 Azimuth and elevation angles variations

Serial number	Azimuth (°)	Elevation (°)
1	236.328	14.373
2	245.231	35.761
3	347.229	67.375
4	45.395	1.829
5	312.700	10.725

After plugging these values in the above-derived formulas along with other calculated values of X, Y, and Z, we can obtain fairly the same values as computed in after the simulations for an efficiency of 4 out of 5 times with the exception being the case when azimuth is 45.395° and elevation is 1.829°. This allows us a good efficiency value of 80% for the above derived formulas. It is advised to double-check the obtained values with different angle determination methods so as to avoid any errors that might prove to be very crucial in high-efficiency relying systems.

5 Conclusion

This paper has presented a thorough and comprehensive mathematical analysis, providing a detailed derivation of the relationship between azimuth and elevation angles. Through systematic exploration and application of trigonometric principles, the study has elucidated the intricate connection between these crucial spherical coordinates. By employing precise mathematical expressions and leveraging the atan2 function to address quadrant ambiguities, the derived formulas offer robust solutions applicable across various disciplines, including astronomy, navigation, and engineering. This comprehensive analysis not only contributes to a deeper understanding of the geometric intricacies involved but also lays a solid foundation for practical applications requiring precise determination of directional coordinates in three-dimensional space. Moving forward, this research provides a valuable resource for further investigations and applications in diverse fields where azimuth and elevation angles play a pivotal role.

Also, the integration of azimuth and elevation angles with emerging technologies like machine learning and artificial intelligence holds great promise for enhancing their utility and applicability in diverse fields. Future research could explore novel approaches for leveraging machine learning algorithms to optimize angle determination processes, automate pattern recognition tasks, and improve the accuracy of angle predictions in real-time applications. Furthermore, the integration of azimuth and elevation angles with sensor fusion techniques, Internet of Things (IoT) platforms, and distributed computing architectures could enable the development of intelligent systems capable of autonomously sensing, analyzing, and responding to spatial information in dynamic environments.

Acknowledgements This research work is solely an unaffiliated work done individually by the authors of this paper and is not done under or for any specific organization.

References

1. Nielsen RO (1994) Azimuth and elevation angle estimation with a three-dimensional array. IEEE J Oceanic Eng 19(1):84–86

2. Sato H, Morimoto M, Sato H (2020) Perception of azimuth angle of sound source located at high elevation angle: effective distance of auditory guide signal. Appl Acoust 159:107084
3. Liang J (2009) Joint azimuth and elevation direction finding using cumulant. IEEE Sens J 9(4):390–398

Hierarchical Quantum Classification

Prashant Gohel and Manjunath Joshi

Abstract Quantum computing in machine learning gives us a new tool to address few complex problems in the domain of artificial intelligence (AI). In this work, we use the hybrid, i.e., a combination of quantum and classical Quantum convolution neural network (HQCNN) in a hierarchical way for image classification. A four class classification is obtained hierarchically using three HQCNNs. In phase 1, we perform binary classification using HQCNN1 to get the two class classification with each class a mix of two individual classes. We then further classify each of the images into their individual categories with the use of HQCNN2 and HQCNN3 in phase 2 by performing two more binary classifications. Two experiments are conducted to show the efficacy of our proposed approach. In the first experiment we use X-ray images of organs consisting of Chest, Hand, Ankle, and Knee images which are taken from UNIFESP dataset. In our second experiment we use Deer, Horse, Airplane, and Ship images drawn from CIFAR10. We implement our proposed approach on Pennylane Software Development Kit (SDK), a simulation tool that can be implemented on a classical machine. Comparison of results with the classical method clearly indicates that the training using the proposed approach takes fewer number of epochs with improvement in accuracy and the convergence is also stable when compared to the classical approach.

Keywords Quantum machine learning · Hierarchical classification · Variational quantum circuit

P. Gohel (✉)
HCL Technologies, Bengaluru, Karnataka, India
e-mail: prasant_gohel@daiict.ac.in
URL: https://in.linkedin.com/in/prashant-gohel-a3770429

M. Joshi
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India
e-mail: mv_joshi@daiict.ac.in
URL: https://sites.google.com/site/manjunathvjoshi/

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_24

1 Introduction

The use of classical machine learning and deep learning-based solutions in healthcare and the medical ecosystem has increased with a larger impact. The capability of CNN for deep feature extraction is the key to increased use of CNN in medical image data analysis [1]. CNN based models have also performed better for COVID-19 detection based on the analysis of Chest X-ray images [2]. There is a surge in the use of deep learning (DL)-based approaches to solving the problem of automated organ detection and classification in medical surgery that can help to reduce organ search time. However, the deep learning-based classical CNNs fail to learn complex patterns in images with low computing costs. This is because the training time increases drastically even while using the high end Graphical Processing Units (GPUs).

Quantum computing has the advantage of enhanced computing power. It can solve intractable problems like Shore's algorithm [3] for cracking the RSA algorithm (public-key cryptosystem), which is extensively used in encrypted financial and secure transactions. Using Quantum computing, one can quickly solve factoring problems that are considered NP-hard [4]. Quantum computing has inherent properties like superposition and entanglement, so it outperforms the classical way of computing. One can also explore both classical and Quantum fields to solve problems at a large scale [5]. Recent research on QML shows that Quantum computing enhances performance in terms of learning capacity, convergence time, accuracy, recall as well as feature representations.

Our work in this paper is based on HQCNN [6], This represents a hybrid neural network in which input data is encoded into Quantum states and processed using Quantum and classical CNNs to perform the classification. HQCNN [6] enhances the performance of classical CNN by leveraging the powerful Quantum computational capability with the Quantum data. It involves variational (parameterized) Quantum circuits that perform Quantum operations. .

2 Preliminaries and Variational Quantum Circuits

Quantum circuits are the foundation for Quantum machine learning. By using Quantum circuits, one may construct variational (parameterized) Quantum circuits that work similarly to deep neural networks with parameters. Here, we review a few terms used in Quantum-based computing described in the subsequent sections.

- **Qubits:** Qubits are similar entities to classical bits but in the Quantum domain. Quantum bits also known as Quantum states take complex values in Hilbert space and are denoted as $|\psi\rangle$. The states $|0\rangle$ and $|1\rangle$ constitute the basis which represent the vectors $[1\ 0]^T$ and $[0\ 1]^T$, respectively. Any Quantum state can be represented as a linear combination of basis states $|0\rangle$ and $|1\rangle$ as given in Eq. 1.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{1}$$

Here, α and β represent the amplitudes associated with $|0\rangle$ and $|1\rangle$ respectively. Note that $|\alpha|^2$ and $|\beta|^2$ represent the probabilities of Quantum state to be in $|0\rangle$ and $|1\rangle$ state, respectively and $|\alpha|^2 + |\beta|^2$=1.

- **Superposition:** Superposition is a Quantum mechanics property that allows a Qubit to be in multiple states. Qubit state can be in any of the superposition (combined) states. The superposition of two Qubit states can be written as in Eq. 2.

$$|\psi\rangle = \frac{\alpha|00\rangle + \beta|11\rangle}{\sqrt{2}}. \tag{2}$$

Here, $|00\rangle$ represents the tensor product of $|0\rangle$ and $|0\rangle$ that gives $[1\ 0\ 0\ 0]^T$ and $|11\rangle$ corresponds to $[0\ 0\ 0\ 1]^T$.

- **Entanglement:** Entanglement is an essential property of Quantum states that accounts for parallel processing in Quantum computing. Here, the entire state represents a single entangled state (inseparable whole) [7]. In an entangled state, we cannot express one sub-constituent of the state without referencing other(s). This property of Quantum system speeds up the computation power of Quantum computing exponentially.
- **Measurement:** The measurement operation is executed as the last step in a Quantum circuit output, which gives classical bits. During measurement, states collapse from their superposition state under the effect of the decoherence process. For example, from the linear combination of $|0\rangle$ and $|1\rangle$ to the specific state.
- **Variational Quantum Circuit:** It is a parameterized circuit in which the parameters are learnt during training.

3 Literature Review

Image classification is a computer vision problem that can be solved by using the manual feature extraction method or by making use of deep learning approaches. There are numerous applications of image classification, such as autonomous driving, medical diagnostics. The image classification pipeline consists of two major steps: feature extraction and classification.

Prior to the use of deep convolutional neural networks (CNN), the number of methods based on handcrafted features, such as Scale-Invariant Feature Transform (SIFT) [8], Speeded Up Robust Features (SURF) [9], and Features from Accelerated Segment Test (FAST), were used for classification. SIFT represents a feature vector that extracts the local features in images. The SURF method is a fast and robust algorithm for local similarity invariant representation. FAST is a similar feature representation proposed by Rosten and Drummond [10] for identifying interest points in an image. The use of such features for classification is known as feature

engineering. These features can be used as input to approaches such as K Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision tree, naive Bayes, etc., for training a classifier. All these correspond to traditional machine learning methods. However, in deep learning-based approaches such as CNN, there is self learning of features and classification, thus eliminating the need for feature engineering. There are many advanced benchmarking models of CNN like VGGNET [11], Imagenet [12], AlexNet, etc. [13] etc. Attention based image classification models like ViT [14] have also been proposed that outperform the classical CNN. In recent times, researchers have explored Quantum methods to solve the classification problems in which they exploit the power of both classical machine learning and Quantum computing. Henderson et al. proposed a VQC based hybrid model for feature extraction from images [6]. This hybrid model takes classical input as images and transforms it into Quantum space. The transformed features after measurement (i.,e decoding operation) are used as input for training a classical CNN. The authors test their model on a commonly used simple MNIST dataset and do not consider the hierarchical way of classification as proposed in our approach. With the success of Quantum computing for machine learning tasks, there have been several variations of quantum algorithms developed for solving computer vision related applications. Some of them include Quantum reservoir computing (QRC) [15], Quantum circuit learning (QCL) [16], Quantum kitchen sinks (QKS) [17], Quantum variational classifiers, etc.

In our proposed approach, we use a hierarchical approach for image classification. To the best of our knowledge, researchers have not yet explored the idea of a hierarchical approach for classification while using the Quantum methods. Note that classical to Quantum transformation of input images is a key to enhanced performance while using the Quantum approaches. The novelty of our hierarchical approach is that first, it performs binary classification on the given data based on their distinct features; for example, using HQCNN1, separate the upper & lower body part images when we consider organ classification with X-ray images. Subsequent binary classifications using HQCNN2 and HQCNN3 separate the Hand and Chest in the upper body part and the Ankle and Knee in the lower body part. The advantage here is that the classification of organs can be carried out by using the easily discriminating features of upper and lower body parts first (HQCNN1) and then performing the classification on smaller subsets (HQCNN2 and HQCNN3) to obtain the individual organ classification. Note that in case we use a single 4 class classifier (multi class classification), the accuracy suffers since the classifier has to discriminate all 4 classes with one classifier only. We would like to mention here that though we use 3 binary classifiers, we train the HQCNN2 and HQCNN3 simultaneously to restrict the training time.

4 Proposed Approach

Since our work is based on the hierarchical approach with the use of HQCNN, we first briefly discuss the same. Henderson et al. [6] proposed an architecture of HQCNN to achieve better performance for kernel estimation in high dimensionality.

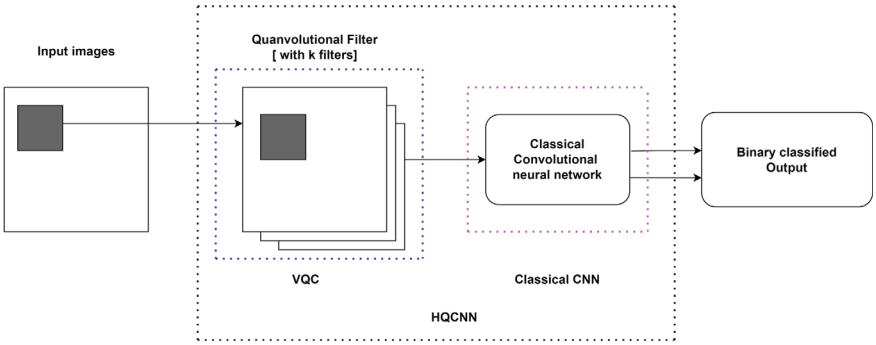


Fig. 1 HQCNN topology

HQCNN enhances CNN's performance by leveraging powerful Quantum computational aspects in transformed data. The proposed approach uses 3 HQCNNs, and the block diagram of one of the HQCNNs is shown in Fig. 1. HQCNN has two main components, i.e., VQC and the classical CNN. VQC has three major components that can be better understood using Fig. 2. The three components are: (i) Classical to Quantum data encoding (or embedding) circuit, (ii) Quantum circuit with variational parameters, and the (iii) measurement component. Note that in Fig. 1, we have a block representing the VQC with all three components mentioned above. The CNN-based VQC has a stack of Quantum convolutional filters, which are learned by processing the Quantum data obtained by encoding the classical input. These Quantum convolutional filters are also known as Quanvolutional filters. Each Quanvolutional filter uses a different configuration of variational parameters, which are learned during the filtering process. Quanvolutional layers in VQC are nothing, but Quantum circuits that perform operations on Quantum data and convert it into complex feature maps in the Quantum domain. The following steps correspond to the encoding and processing of data.

- As a first step of the encoding process, local subset of the input images is used. A small local patch u_x of n $\times$ n pixels is extracted from each input image.
- Each u_x patch in the classical domain is transformed by encoding into Quantum space using the encoding $e()$ in VQC. For each Quanvolutional filter, a specific encoding function is defined. Qubits initialized to $|0\rangle$ are employed while encoding. The initial state after encoding is given by $i = e(u_x)$.
- The encoded Quantum bits are then processed by K filters and the output is measured to get back the processed data in classical domain. The measurement operation can be represented as $f_x = d(o_x) = d(q(i_x)) = d(q(e(u_x)))$.
- Variational (filter) parameters and the classical CNN parameters are varied during the learning process to get the desired result.

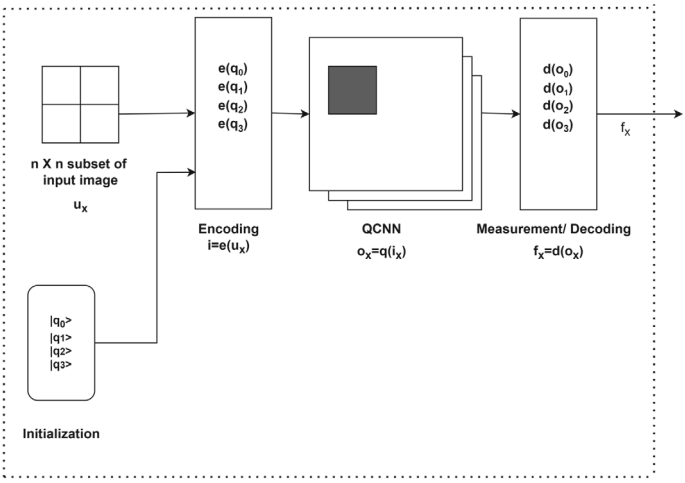


Fig. 2 Detailed block schematic of variational quantum circuit (VQC)

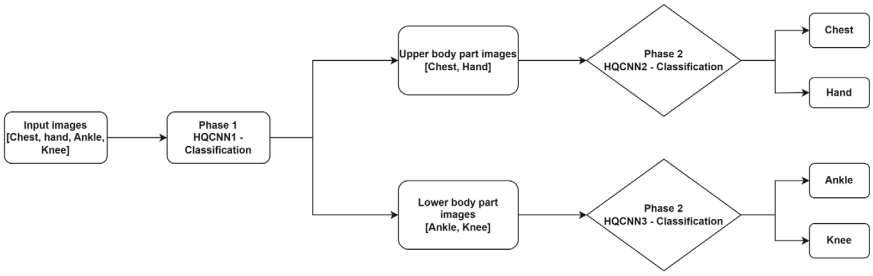


Fig. 3 Proposed hierarchical approach

4.1 Hierarchical Classification

Let us now discuss the proposed hierarchical approach for classification for organ classification by making use of Quantum processing. Note that the same architecture can be used to train when we use the CIFAR10 dataset.

The block schematic of our approach is depicted in Fig. 3 considering the input image as X-ray image.

Referring to Fig. 3, in our hierarchical classification, we first classify the images into two categories, i.e., upper and lower body parts, thus separating the Chest and Hand belonging to an upper body part and the Ankle Knee belonging to the lower body part. The phase 1 binary classification (upper body parts vs. lower body parts) is performed by HQCNN1. After this, further classification into individual classes is performed in the second phase. HQCNN2 performs Chest versus Hand classification, while HQCNN3 classifies Ankle and Knee. As already mentioned, the use of a

hierarchical approach enhances the feature discrepancy among classes, and hence the classification accuracy is improved. We want to mention here that the hybrid approach is suitable for the currently available noisy intermediate scale Quantum devices [18].

5 Experimental Setup and Results

The experimental setup is designed to compare the performance of classical and Quantum hierarchical classifications. We also compare the performance against the non-hierarchical case. For comparisons, We use quantitative measures such as accuracy and loss. As already mentioned, we conducted two experiments to show the effectiveness of the proposed quantum approach. Our first experiment is based on the hierarchical classification of X-ray images of body parts, and the second experiment is to choose images of 4 classes (Airplane, ship, deer, and horse) of the CIFAR-10 dataset and classify them into animal and automobile classes in phase 1 and then further classify as Airplane versus ship and Deer versus horse subclasses in phase 2. For organ classification we used UNIFESP X-ray dataset [19] for which sample images are shown in Fig. 4. This dataset comprised multiple classes such as Chest, Hand, Kidney, Leg, Shoulder, Elbow, Thumb. However, in our experiment, we considered only four classes since our priority was to test the power of the hierarchical Quantum approach for classification. This also has facilitated us to make use of binary classifiers only. Note that we only consider the true positives and negatives after phase 1 and thus wrongly classified images are removed from the training data for phase 2. Binary cross entropy between the probability of the true class and the predicted class is used as the loss function. We used Adam's optimization to minimize the loss function for the hierarchical as well as for the classical. Accuracy is measured as a fraction of correct classification.

All our Quantum-based simulations are carried out by using a software development kit (SDK) called Pennylane, which we run on a classical machine with an Intel Core i5 processor running at 2.60 GHz with 16 GB memory. Here, we use 4 Qubits and Quantum circuits with 4 layers. An angle encoding approach with phase value as π is used for converting the classical data to Quantum states [5]. Quanvolutional layers (please see K filters as shown in Fig. 1) are applied on these encoded states.

(a) Chest (b) Hand (c) Ankle (d) Knee

Fig. 4 Sample X-ray images from training images from UNIFESP dataset

Table 1 Accuracy comparison for phase 1 (Upper vs. Lower body parts) classification between the classical and quantum after 22 epochs on UNIFESP dataset

Upper vesus Lower	Classical	Quantum
Train	0.9523	1.0000
Test	0.9137	0.9459

Table 2 Accuracy comparison for phase 2 classification (Chest vs. Hand) and (Ankle vs. Knee) after 22 epochs on UNIFESP dataset

Type of organ classification	Classical	Quantum
Train (Chest vs. Hand)	0.9233	0.9771
Test (Chest vs. Hand)	0.9167	0.9637
Train (Ankle vs. Knee)	0.8746	0.9038
Test (Ankle vs. Knee)	0.8429	0.8971

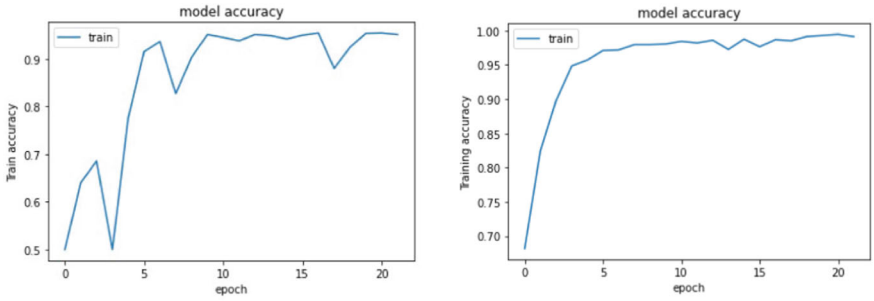
Table 3 Accuracy comparison for non-hierarchical multi class (4 class) classification after 22 epochs on UNIFESP dataset

Non-hierarchical	Classical	Quantum
Test	0.8703	0.9243

Tables 1, 2 and 3, show the accuracy comparisons for experiments on X-ray images. In Table 1, we compare the classification accuracy between the classical and hierarchical approaches for Phase1. The result is shown for 22 training epochs only. As seen from the Table 1, in just 22 epochs the Quantum approach achieves a training accuracy of 1.0000 (i.,e 100%) where as the classical approach gives us only 0.9523. As far as the testing accuracy is concerned, we get an accuracy 0.9459 for Quantum case while the classical approach gives 0.9137. Note that the testing accuracy is not relevant in Phase1, since we use only the correctly classified examples for phase 2 or final phase classification. However, we have included it here to stress that the Quantum-based learning captures the discriminating features of data better when compared to the classical i.,e, we believe that multivariate probability density functions (PDFs) of the two different class data are better captured when using the Quantum approach. We observed that the classical approach takes almost 100 epochs to reach training accuracy of 97% in phase 1 indicating that our hierarchical Quantum approach converges quickly.

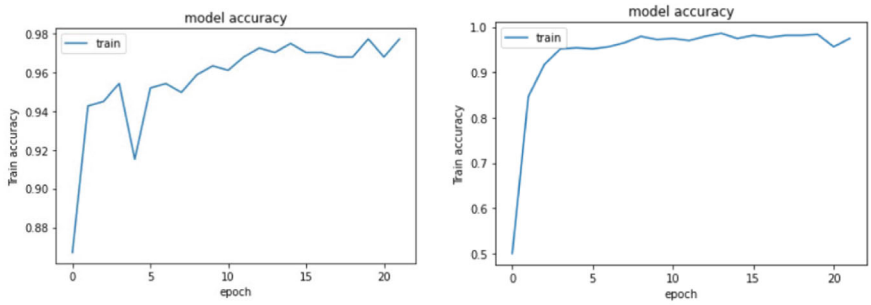
The accuracy comparisons for phase 2 are given in Table 2. As shown in this table, for Chest versus Hand classification (phase 2), proposed method gives us a training accuracy of 0.9771 where as the classical has only 0.9233. The corresponding test accuracy in this case are 0.9637 and 0.9167, respectively.

Similarly, Ankle versus Knee classification also performs better when compared to classical approach. This is also given in in Table 2, where we see that the training accuracy for our approach is 0.9038 while for classical it gives 0.8746. We see



(a) Phase 1 Classical accuracy (b) Phase 1 Quantum accuracy

Fig. 5 Phase 1 training accuracy comparison on UNIFESP dataset



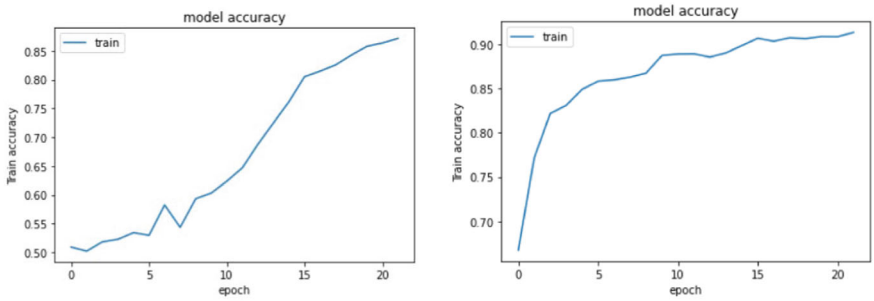
(a) Phase 2 Classical accuracy (b) Phase 2 Quantum accuracy

Fig. 6 Phase 2 training accuracy comparison of (Chest vs. Hand) on UNIFESP dataset

that the test accuracy is almost 5% better for both the upper and lower body part image classifications. As already mentioned, this happens due to better discriminating capability of the proposed hierarchical Quantum approach.

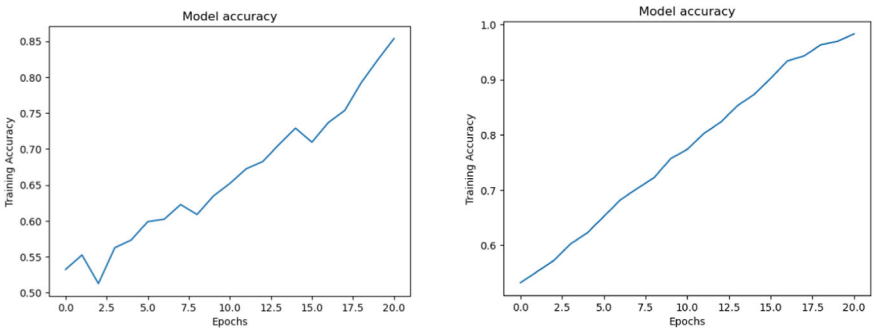
Let us now discuss in brief to see how the Quantum approach performs when compared to the classical if we do not consider the phase wise classification, i.e., when we directly use multi class classification which corresponds to non-hierarchical method. In Table 3, we show these results. This experiment is also conducted for 22 epochs only. Looking at the table, we see that the classical approach gives us a test accuracy of 0.8703 whereas the Quantum approach has an accuracy of only 0.9243 which is once again 5% more than the classical. This reiterates that the Quantum Machine Learning (QML)-based approach gives improved performance even when we do not consider hierarchical way of classification.

Phase 1 training accuracy plots of classical and Quantum approaches are depicted in Fig. 5. Similarly, training accuracy plots of Chest versus Hand and Ankle versus Knee classifications in phase 2 are depicted in Figs. 6 and 7, respectively.



(a) Phase 2 Classical accuracy (b) Phase 2 Quantum accuracy

Fig. 7 Phase 2 training accuracy comparison (Ankle vs. Knee) on UNIFESP dataset



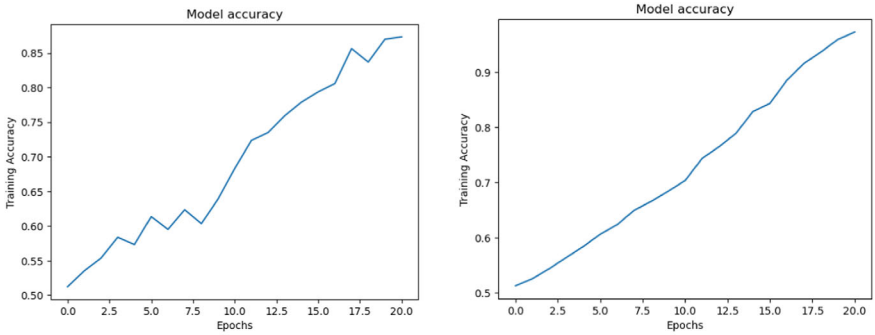
(a) Phase 1 Classical accuracy (b) Phase 1 Quantum accuracy

Fig. 8 Phase 1 training accuracy comparison (Animals vs. Automobiles) on CIFAR10 dataset

We now discuss briefly the results of 2nd experiment where we use CIFAR10 dataset. Training accuracy plots of CIFAR-10 classification of phase 1 for Animals versus Automobiles is shown in Fig. 8 and the same for for phase 2 for Airplane versus Ship and Deer versus Horse classifications are given in Figs. 9 and 10, respectively. One can see from these experiments that the accuracy stabilizes in few epochs in our proposed approach, whereas it exhibits an oscillatory behavior when we use classical method.

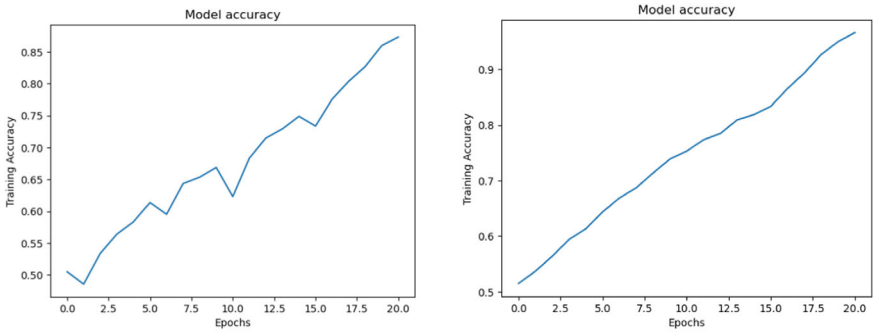
6 Conclusions

In this work, we have proposed an hierarchical way of image classification using hybrid Quantum convolution neural network (HQCNN). Our results show that that the speed of convergence as well as accuracy are improved by a significant margin when compared to the classical approach. Key factors behind this improved per-



(a) Phase 2 Classical accuracy (b) Phase 2 Quantum accuracy

Fig. 9 Phase 2 training accuracy comparison (Airplane vs. Ship) on CIFAR10 dataset



(a) Phase 2 Classical accuracy (b) Phase 2 Quantum accuracy

Fig. 10 Phase 2 training accuracy comparison (Deer vs. Horse) on CIFAR10 dataset

formance are due to the Quantum computing properties such as superposition and entanglement and the use of hierarchical approach. Our future work involves using the Quantum methods to come up with the solutions to secured machine learning algorithms that are computationally expensive.

References

1. Ozturk S (2022) Convolutional neural networks for medical image processing applications
2. Chowdhury M, Rahman T, Khandakar A, Mazhar R, Kadir M, Mahbub Z, Islam K, Khan MS, Iqbal A, Al-Emadi N, Reaz MBI, Islam M (2020) Can AI help in screening viral and Covid-19 pneumonia? IEEE Access 8:132665 –132676
3. Shor Peter W (1997) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput 26(5):1484–1509

4. Hochba Dorit S (1997) Approximation algorithms for np-hard problems. SIGACT News 28:40–52
5. Schuld M, Killoran N (2018) Quantum machine learning in feature hilbert spaces. Phys Rev Lett 122(4):040504
6. Henderson Maxwell P, Shakya Samriddhi, Pradhan Shashindra, Cook Tristan (2019) Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Mach Intell 2:1–9
7. Hidary Jack D (2019) Quantum computing: An applied approach. An Applied Approach, Quantum Computing
8. LoweDavid G (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision
9. Bay Herbert, Ess Andreas, Tuytelaars Tinne, Van Gool Luc (2008) Speeded-up robust features (surf). Comput Vision and Image Understanding 110:346–359
10. Rosten Edward, Porter Reid B, Drummond Tom (2008) Faster and better: a machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell 32:105–119
11. Han S, Liu X, Mao H, Pu J, Pedram A, Horowitz M, Dally WJ (2016) Eie: efficient inference engine on compressed deep neural network. In: 2016 ACM/IEEE 43rd annual international symposium on computer architecture (ISCA), pp 243–254
12. Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E (2012) Imagenet classification with deep convolutional neural networks. Commun of the ACM 60:84–90
13. Sengupta A, Ye Y, Wang RY, Liu C, Roy K (2018) Going deeper in spiking neural networks: Vgg and residual architectures. Front Neurosci 13
14. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020
15. Fujii K, Nakajima K (2021) In: Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices, Springer, Singapore, pp 423–450
16. Mitarai K, Negoro M, Kitagawa M, Fujii K (2018) Quantum circuit learning. Phys Rev A
17. Wilson CM, Otterbach J, Tezak NA, Smith RS, Polloreno AM, Karalekas PJ, Heidel S, Sohaib Alam M, Crooks GE, da Silva MP (2018) Quantum kitchen sinks: an algorithm for machine learning on near-term quantum computers. Hyperimage http://arxiv.org/abs/QuantumarXiv:QuantumPhysics
18. Chen SY-C, Yang C-HH, Qi J, Chen P-Y, Ma X, Goan H-S (2019) Variational quantum circuits for deep reinforcement learning. IEEE Access 8:141007–141024
19. Farina FKE (2012) Unifesp x-ray body part classifier competition

Emulation of Recommendation System Using QSVT Framework of QiML on CPU

Akshay Patil, Abhishek Tiwari, Vivek Khaneja, and Nagendra Singh

Abstract Classical Computing (CC) takes place in the binary form of 0 and 1, whereas Quantum Computing (QC) takes place adhering to fundamental principles of quantum mechanics, i.e., superposition, entanglement, and tunneling. Humanity is in the race to show that QC is better than CC in terms of time, accuracy, and memory requirements. But since Fault-Tolerant QC hardware with solving problems computationally exhaustive for today's CC are out of reach, we in this paper emulate Use case of the Recommendation System on Quantum Singular Value Transform Framework (QSVT) of Quantum-Inspired Machine Learning (QiML) on CPU using the Qiskit Library, pyqsp library and have shown a quantum advantage in terms of accuracy, in terms of time and memory requirement classical implementation is a winner as computations in the Quantum Emulations are larger owing to unitary matrix conversions of input matrix and considerable depths, but these depths can be reduced to approximate the target polynomial function in which case the Quantum Implementation is comparable to/better than its counterpart at cost of accuracy.The entanglement principle of QM is used in the QSVT Circuit for entangling the QSVT Qubit with unitary Qubits by using the CNOT gates. Also, the entanglement principle is inherently present in the Input User Preference Rating Unitary Matrix as well, if the exact tensor decomposition at a break-off qubit length of the input user preference unitary matrix is not possible.The input quantum state of our QSVT circuit can be initialized to the superposition of input user states to access all the items corresponding to all the users in one run of the circuit. Hence by using the fundamental

A. Patil (✉) · A. Tiwari · V. Khaneja · N. Singh
Embedded Systems R&D, Centre for Development of Advanced Computing, Noida, India
e-mail: akshayrpatil@cdac.in

A. Tiwari
e-mail: abhishek@cdac.in

V. Khaneja
e-mail: vivekkhaneja@cdac.in

N. Singh
e-mail: nagendras@cdac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_25

principles of quantum mechanics, we have shown quantum advantage over classical implementation in terms of accuracy.

Keywords QSVT · QML · QiML · Classical recommendation system · Quantum recommendation system · Quantum AI

1 Introduction

Before delving into the Quantum Recommendation System and its intricacies, let's establish the basics of the Recommendation System [1–4]. Consider a scenario with users numbered from 1 to m and items from 1 to n. Each user rates items based on personal preferences, suitability, ease of use, etc., resulting in a user rating matrix of m columns andn rows. The recommendation system's task is to fill blank values in this matrix, minimizing the difference between known and newly reconstructed elements. Quantizing errors between known and reconstructed elements involves methods like RMSE, MAE, R-squared error, etc. Several techniques for reconstructing a user rating matrix include Global Average, User Average, Movie Average, Matrix Factorization (Collaborative Filtering, Latent Factors with Biases), etc. The Netflix Prize-winning approach used a novel combination of these techniques, resulting in an RSME of 0.8563. A matrix polynomial is then designed to reconstruct a user rating matrix showing minimum error. The primary goal is to benchmark the accuracy, time and memory of Quantum Algorithms in the QSVT framework against Classical ML Algorithms. A movie recommendation dataset (610 by 610) is used, scaled to match qubit requirements (512 users and 512 items). The dataset is block-encoded into a unitary matrix which involves normalization and block encoding of the input matrix, requiring 10 qubits (9 Qubits + 1 Ancilla Qubit).One more qubit is required for QSVT Emulation called as QSVT Qubit. The Quantum Emulation demonstrates lower RMSE than classical Recommendation Algorithms, showing Quantum Advantage in accuracy.The QSVT framework involves applying polynomial transformations to singular values/eigenvalues of an input unitary matrix using rotation gates with parameters (ϕ). Rotation gates applied on QSVT Qubits, alongside CNOT gates with QSVT Qubit as control qubit and ancilla qubit as target qubit [5–7] perform required polynomial transformation on singular values. The values of ϕ are determined by fitting coefficients using classical ML frameworks [6, 8, 9]. Once parametrized ϕ values are obtained, the QSVT Framework of QiML is implemented using 11 qubits on a classical emulator, i.e., CPU with Qiskit Library. Classical post-processing involves removing probability amplitudes associated with Ancilla qubit and QSVT qubit. Absolute values of complex amplitudes are taken, and amplitudes are re-scaled to obtain user x's column ratings for all items in the reconstructed user-rated matrix. The user x can be changed by altering the input state of the 9 Recommendation Qubits using the Pauli X gate.

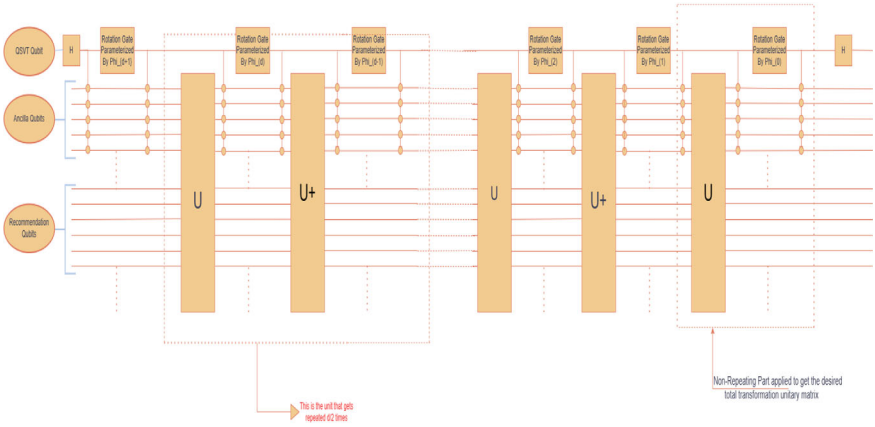


Fig. 1 QSVT circuit for implementation of recommendation system

Note: Quantum Machine Learning involves implementing machine learning algorithms on real quantum hardware. In contrast, Quantum-Inspired Machine Learning is the emulation of these Quantum Machine Learning Algorithms on classical hardware by using Qiskit Library Simulators [10].

Figure 1 shows the Quantum QSVT Circuit designed and tailored for the specific application of the Recommendation System, which consists of three Qubit sections-1. QSVT Qubit, 2. Ancilla Qubit, and 3. Recommendation Qubits. The QSVT Qubit is initialized to a superposition state by using the Hadamard Gate, followed by CNOT gates to entangle the QSVT Qubit (control qubit) with the Ancilla Qubit (target Qubit), so the evolution of the state vector due to the rotation gates on the QSVT Qubit is reflected on the ancilla and recommendation qubits state vector of the circuit. The Unitary (U) and the Unitary Dagger (U^{+}) matrices repeated d/2 +1 times (where d is the degree of the classical polynomial transformation), which multiplies to identity, followed by Unitary Matrix at the end of the circuit to deploy all the evolution changes made by the rotation gates on the last Unitary matrix, again followed by rotation gate to deploy last constant term of addition in the polynomial to the Evolved Unitary matrix as last layer of fine-tuning the transformation of singular values of unitary matrix as per the classical polynomial equation tailored to recommendation system.

2 Pre-Processing and Algorithms In Classical Recommendation System

The data set [11], has 100,000 Ratings given by 612 Users for 9000 Movies. Classical pre-processing mainly involves removing the timestamp column, and converting the data frame into a user input rating matrix with user number as columns and item

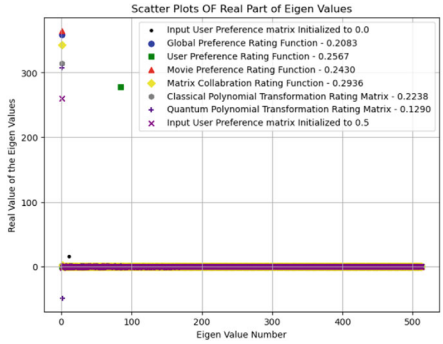


Fig. 2 All classical and quantum methods, real part plots of eigen values versus eigen number

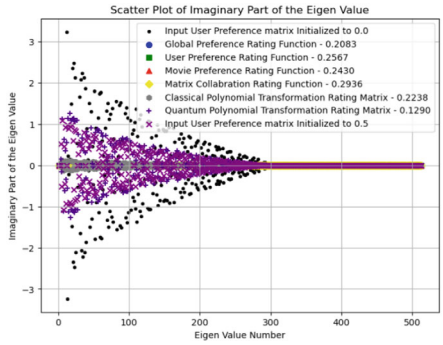


Fig. 3 All classical and quantum methods, imaginary part plots of eigen values versus eigen number

number as row and rating as elements of the matrix, with non-rated values initialized to 0.0 & 0.5. As the user rating matrix is to be implemented on Quantum Emulators (Gate Based Quantum Computing), the dimensions of the input user rating matrix should be divisible by $2^n * 2^n$, excess columns and rows of the user rating matrix are removed to make it a square 512*512 matrix.

Any square matrix can be characterized by its eigenvalues and eigenvectors owing to eigenvalue equation $A|\lambda\rangle = \lambda|\lambda\rangle$. Eigenvalues of reconstructed matrices have been linked to RMSE and other errors. If the reconstructed user rating matrix's eigenvalues follow the trend of the input user rating matrix initialized to zero's eigenvalues then the reconstructed matrix has the lowest RMSE and other errors of all the other reconstructed user rating matrices.

Figures 2 and 3 plot the real parts and imaginary parts of the eigenvalues of all the classical methods and quantum method used, and infers a clear conclusion that if the eigenvalues of the result matrix for the method used closely follow the eigenvalues trend of the input polynomial transformation rating matrix initialized to 0 and 0.5 then that corresponding method will be more accurate.

2.1 Classical Polynomial Transformation Rating Method

To implement the Recommendation system on the QSVT Framework of QiML, we aim to design a polynomial transformation equation for the input user rating matrix by minimizing RMSE and other errors in the reconstructed matrix. Input user rating matrix initialized to zero can be thought of as the user giving the 0 rating to all his unrated items or unrated movies. To prevent bias toward 0 initialization, unrated elements are set to 0.5, i.e., average of min and max rating. When exploring matrix powers, resulting matrices often exceed the maximum limit of 1. To create a valid reconstructed rating matrix, coefficients for polynomial transformation must be assigned, ensuring a maximum value of 1. Thus, the equation for the polynomial transformation of the input user rating matrix initialized to 0.5 (B) can be expressed as follows,

$$\text{Final } B = f(B) = \sum_{i=1}^{d} \left(\frac{\frac{1}{\max(\text{element of } B^d)} \cdot B^d}{\max(\text{element of Final } B)} \right) \tag{1}$$

The Polynomially transformed matrix (Final B), shows minimum errors, with all errors asymptotically approaching zero as the degree of transformation is increased as shown in the Figs. 4 and 5.

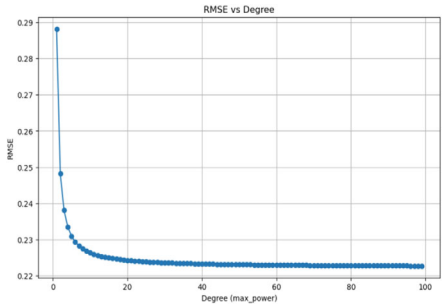


Fig. 4 RSME versus degree of the polynomial transformation plot

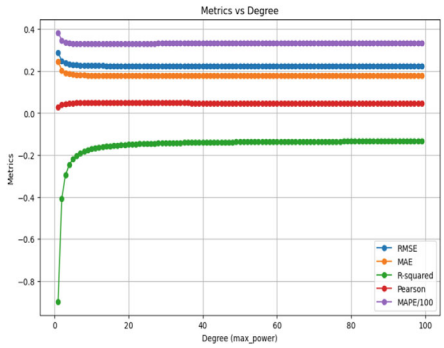


Fig. 5 All metrics versus degree of the polynomial transformation plot

Fig. 6 Implemented QSVT circuit using Qiskit library

3 Quantum Singular Value Transformation (QSVT)

Figure 6 of the QSVT Circuit is designed and tailored to the recommendation system after reading [5–8] for emulating the polynomial transformation of the recommendation system as discussed earlier.

To develop a QSVT Circuit for any use case emulation, let us understand the implemented circuit in detail. The first requirement is the conversion of the input matrix into a unitary matrix to preserve the norm of the input state vector as 1 (not considering the negative probability discussions of Wigner Interpretation). After every unitary gate operation on input state vector,the new probability amplitudes preserves norm of old probability amplitudes, also unitary matrix preserves the reversibility condition of quantum gates, i.e., $UU^{+} = \mathrm{I}$.

The main function of the CNOT gates before the Rotation gates is to entangle QSVT Qubit with Unitary Qubits (Recommendation Qubits + Ancilla Qubit), the Rotation gate operation can be implemented on Unitary Qubits via QSVT Qubit, then all the Rotation gates are again followed by a CNOT gate, to equivalently disentangle the QSVT Qubit from the Unitary Qubits.The main function of the Hadamard gate at the start of QSVT Qubit is to initialize the input zero state of the QSVT qubit into equal superposition of $|0\rangle$ and $|1\rangle$,later entangled with the Unitary Qubits via the CNOT gate,to deploy the distinguishing Quantum power of superposition and entanglement compared to classical in QSVT circuit , followed by the Hadamard gate at the end of the circuit on QSVT Qubit to remove the superposition and reverse the QSVT qubit back to the input state of $|0\rangle$.QSVT circuit performs unitary rotations in unitary and unitary dagger matrix (1024×1024) elements, i.e., required changes in the magnitude of input user rating unitary matrix as per the polynomial transformation

equation, the number of coefficients of which govern the number of rotation gates of QSVT circuit deployed on QSVT Qubit as shown in the figure and ϕ parameterization magnitudes of rotation gates are governed by the magnitude of coefficients of these polynomials.

So each iteration of rotation gate parameterized by ϕ angles make necessary equivalent changes in elements of input unitary user rating matrix in quantum setting as will be done by coefficient and the highest power of the input unitary user rating matrix in classical polynomial transformation setting as we proceed from left to right in the QSVT Circuit,owing to which for accurate implementation of the matrix polynomial,number of the rotation gates parameterized by ϕ angles (r) are equal to (d) + 1 , where one extra rotation gate on degree of the polynomial transformation is required to account for the last constant of the polynomial and also to make QSVT Qubit state $|0\rangle$ same as its input initialization state after the Hadamard gate.As the element changes are to be done on the input unitary user rating matrix as per matrix polynomial,a unitary gate is deployed after every rotation gate, but to get back the identity transformation on the unitary qubits and keep only the changes of rotation , a unitary gate matrix is followed by a unitary dagger gate matrix, this combination of unitary and unitary dagger gate matrix is repeated $\frac{d}{2} - 1$ times on unitary qubits (Recommendation + Ancilla Qubits), to make necessary quantum equivalent element changes in input unitary user rating matrix up-to [degree–2] of matrix polynomial, i.e., excluding equivalent quantum changes of last two terms of classical polynomial transformation equation.Now comes last non-repeating part of QSVT circuit, i.e., last U and U^+ combination repeating part is followed by a unitary gate matrix, which is preceded and succeeded by rotation gate combinations parameterized by ϕ_1 & ϕ_2 angles as shown in above figure. The non-repeating part is concatenated with remaining QSVT circuit to deploy all required changes done in elements until repeating part of the QSVT circuit on last input unitary user rating matrix to obtain required reconstructed user rating matrix in unitary form.

After run of QSVT circuit, we are not measuring qubits of QSVT Circuit, if we measure QSVT Qubit it will be $|0\rangle$, and Ancilla Qubit will be again $|0\rangle$ (same as its input state) and measurement of Recommendation Qubit will most probably be item state corresponding to user x (decided by input state initialization of Recommendation Qubits of QSVT Circuit),which has highest probability amplitude magnitude in the column of final unitary transformation matrix corresponding to user x (Copenhagen interpretation of QM). The probability amplitude for a complete column of final unitary transformation matrix corresponding to user x is required, hence to obtain values of the square of probability amplitude and reconstructing whole item state for user x, we have to run and measure the QSVT circuit with same input state on recommendation qubits corresponding to user x, a finite number of times, or item state for user x can also be reconstructed by using quantum tomography approaches.

Figure 7 shows a schematic that a QSVT Circuit can be constructed approximately for any universal application by using the Taylor, Maclaurin, Laurent, and other infinite series by first designing a mathematical function of the application (within -1 to 1 mapping constraint) and then converting the function into an infinite series and designing a QSVT circuit for it.

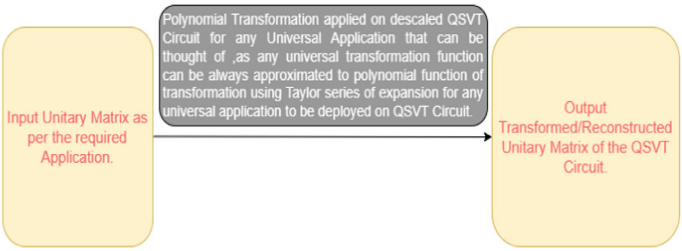


Fig. 7 QSVT algorithm approximation for universal application

Here, we are emulating QSVT Circuit on CPU, i.e., we are in QiML Domain, we have access to item column corresponding to user x, or whole unitary transformed matrix of QSVT Circuit and due to which there is no need to measure Recommendation Qubits and directly use 'state vector simulator' or 'unitary matrix simulator' of Qiskit library to obtain reconstructed User rating matrix in unitary form. On actual Quantum Hardware more number of shots to circuit will be required ,as the measurement will be a necessity to get the information of unitary transformation that goes on in QSVT circuit, as we will not be privy to final reconstructed user matrix as in case of Quantum emulation of the QSVT Circuit on CPU.

Theorem 1 ***Cayley Hamilton Theorem, Spectral Transformation of Matrices and Matrix Polynomial***

An invertible square matrix satisfies its characteristic equation. A polynomial transformation of matrix can be replaced by matrix's eigenvalues and vice versa, i.e., all eigenvalues of the matrix satisfy its corresponding polynomial transformations to obtain eigenvalues of a matrix obtained from the polynomial transformations of input matrix, also known as spectral transformation of matrix. (Cayley Hamilton theorem is a polynomial transformation of matrix that gives zero matrix as output, but spectral transformation is general polynomial transformation of matrix,for future purpose in this paper both are used interchangeably). Let A be the input square matrix having N by N dimensions, i.e., it has N eigenvalues, and $P(A)$, i.e., matrix H be its corresponding polynomial transformation of degree d.H = $P(A) = \mathrm{a}A^{d} + \mathrm{b}A^{d-1} + \mathrm{c}A^{d-2} + \mathrm{d}A^{d-3} + \mathrm{e}A^{d-4} + \ldots + \zeta A^{d-d}$, then the corresponding polynomial transformation is also satisfied by its eigen values (λ) as, $\lambda_H = \mathrm{P}(\lambda) = \mathrm{a}\lambda^{d} + \mathrm{b}\lambda^{d-1} + \mathrm{c}\lambda^{d-2} + \mathrm{d}\lambda^{d-3} + \mathrm{e}\lambda^{d-4} + \ldots + \zeta\lambda^{d-d}$, where λ_H are the eigen values of the matrix H.Hence Cayley Hamilton theorem descales a matrix from the higher dimension of N x N to the lower dimension of the scalar, i.e., its eigenvalues in the polynomial transformation equation.

Theorem 2 ***Quantum Signal Processing Theorem (QSP)*** *[5–8, 10, 12]*

Q.S.P Theorem is an extended version of Cayley Hamilton Theorem/spectral transformation for 2×2 matrices, in which eigenvalues/singular values/values of x over a range on which the polynomial transformation preserves the mapping [−1,1]

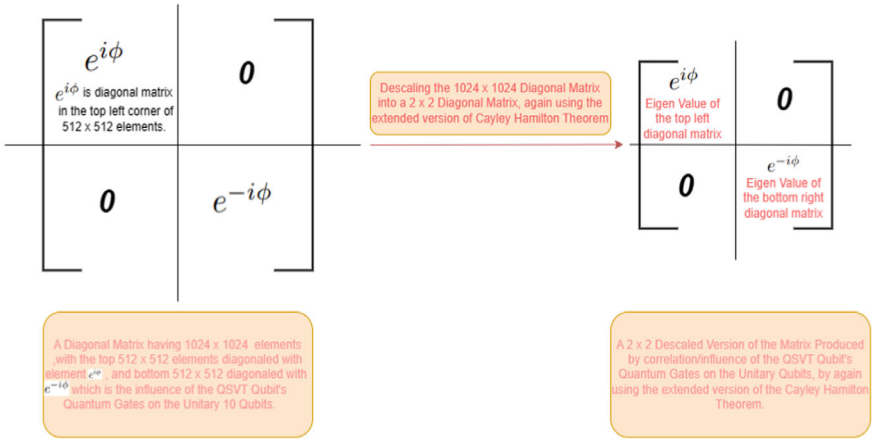


Fig. 8 Descaling the influence of rotation of the QSVT qubit on the unitary qubits from 1024×1024 matrix to 2×2 Matrix

=>$[-1,1]$ are top left corner elements of 2 by 2 matrix.The input unitary user rating matrix then it consists of four blocks , the input user rating matrix in normalized form (B) in the top left corner (512×512 elements), $\sqrt{I - BB^{+}}$ in the top right corner , $\sqrt{I - B^{+}B}$ in the bottom left corner and finally $-B^{+}$ in the bottom right corner, the role of last three blocks is to make input normalized matrix block encoded in the top left corner as unitary.The four blocks of the input unitary matrix are replaced by corresponding eigen values/singular values/values of x over a range on which the polynomial transformation, preserves the mapping $[-1,1]$ => $[-1, 1]$, to get a 2×2 matrix,a descaled matrix version of the 1024×1024 input unitary matrix,a schematic of which is shown in the figure below. Now if we look at the matrix multiplications in the QSVT Circuit, then role of QSVT Qubit Gate is to multiply the $e^{i\phi}$, where ϕ is the rotation gate parameter to all the top left 512×512 elements, & multiply the $e^{-i\phi}$ quantity with bottom right 512×512 elements block encoded in our Input Unitary matrix of 1024×1024 elements. So the descaled form of the Quantum Gates on QSVT Qubit's influence on the Unitary Qubits, i.e., input unitary matrix is a 2×2 matrix, with $e^{i\phi}$ element in the top left corner and $e^{-i\phi}$ element in the bottom right corner, & the remaining two elements on top right & bottom left corners as null/zero elements, which can be deduced from the spectral transformation of the matrix.

Figure 8 shows the schematic descaling of the rotation gates on QSVT Qubit entangled to Ancilla and Recommendation Qubits to 2×2 matrix and Fig. 9 shows the descaling of the Unitary Matrix to 2×2 matrix using the corresponding singular values as discussed above. Figure 10 shows the equation of transformation of the singular values of the descaled input unitary matrix by use of the descaled 2×2 rotation gates, which is used to calculate the rotation angles $[\phi_{10} - \phi_1]$.

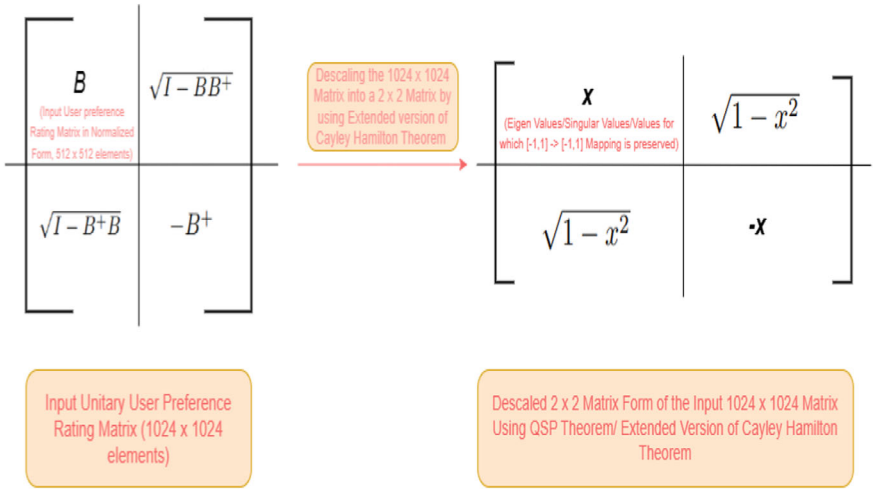


Fig. 9 Descaling of 1024 × 1024 matrix into a 2 × 2 matrix by using extended version of Cayley hamilton theorem

$$\begin{pmatrix} e^{i\phi_{10}} & 0 \\ 0 & e^{-i\phi_{10}} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_9} & 0 \\ 0 & e^{-i\phi_9} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_8} & 0 \\ 0 & e^{-i\phi_8} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_7} & 0 \\ 0 & e^{-i\phi_7} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_6} & 0 \\ 0 & e^{-i\phi_6} \end{pmatrix}$$

$$\begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_5} & 0 \\ 0 & e^{-i\phi_5} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_4} & 0 \\ 0 & e^{-i\phi_4} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_3} & 0 \\ 0 & e^{-i\phi_3} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{i\phi_2} & 0 \\ 0 & e^{-i\phi_2} \end{pmatrix} = \begin{pmatrix} P(x) & \cdot \\ \cdot & \cdot \end{pmatrix}$$

Fig. 10 Matrix multiplications of 2 × 2 descaled matrices from QSVT circuit of unitary qubits which equals 2 × 2 matrix with top left corner element as $P(x)$ (RHS)

Theorem 3 ***QSVT Theorem*** *[4–7, 10, 12]*

It states that it is possible to perform mathematical operation equivalent/analogue to the polynomial transform of the matrix, on a quantum setting using quantum rotation gates parameterized by ϕ on the QSVT Qubit (which inherently transform the singular values of the matrix as per polynomial transform without actually finding singular values of the matrix), provided the mapping of the polynomial transformation is $[-1,1]$ -> $[-1,1]$, i.e., the polynomial is bounded. (even if it's not, approximate quantum mapping is always possible $[-\infty, \infty]$ -> $[-1,1]$). Now using the descaled form of 2 × 2 matrix as discussed above, our Unitary Matrices of 1024 × 1024 dimensions on our 10 Unitary Qubits and the influence of the QSVT Qubit's Rotation Gates on these 10 Unitary Qubits (Again 1024 × 1024 matrix dimension) can be written in 2 × 2 descaled matrix form as (which is done to find the rotation gate parameters phi by curve fitting the target polynomial of classical setting using the unitary rotation gates of the quantum setting)

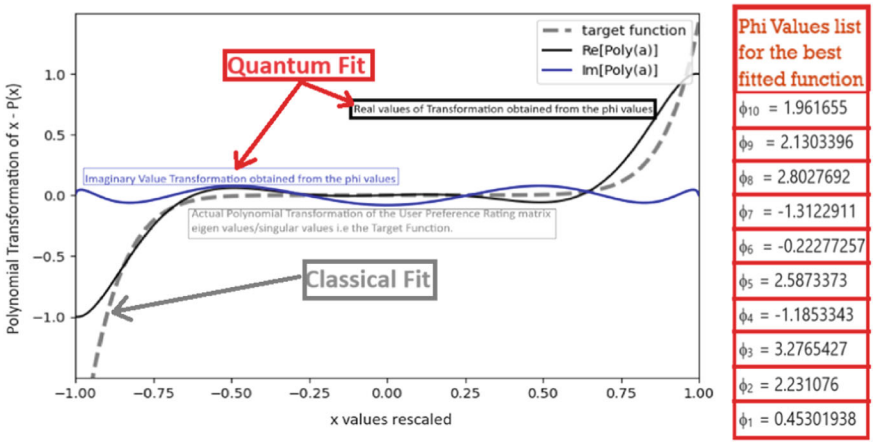


Fig. 11 Best fitted function from a set of ϕ values, i.e., quantum rotation gate setting, actual classical polynomial transform, and set of ϕ values are provided by using the pyqsp library

The values of x in which the polynomial transformation of $[-1,1]$ =>$[-1,1]$ is preserved, can be found by trial and error by putting x in the $P(x)$ equation, the initial guess for x can be taken as eigenvalues of input user rating unitary matrix which is block encoded in the top left corner (512×512 elements) of the 1024×1024 element matrix. Once we get x_{min} and x_{max} values by trial and error, we have infinite values of x in the range of x_{min} and x_{max} that satisfies above descaled 2×2 matrix equation. Hence we have infinite equations for finding the set of $[\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, \phi_8, \phi_9, \phi_{10}]$, $|\phi$ values$|$ = (d) + 1. The number of ϕ values can be reduced to reduce the depth of the circuit to in turn reduce memory and time required to run the QSVT circuit at the cost of the accuracy of the Emulated Recommendation Circuit on the QSVT Circuit.We have target polynomial function P(x) of classical setting that needs to be curve fitted by Polynomial function $P_Q(x)$ obtained from QSVT circuit setting.

Figure 11 shows the actual target polynomial function $P(x)$ in the classical setting for the Recommendation system and obtained Real and Imaginary Polynomial Functions obtained from the Quantum Setting from 2×2 matrix multiplication discussed above, by using the 'pyqsp' library by finding the set of ϕ values.It is this obtained Real and Imaginary Polynomial Functions that govern the accuracy of implemented Quantum QSVT Circuit for Recommendation System (Fig. 12).

Note, since ϕ are rotation angles it has values in limits $[-6.28,6.28]$ in radians. And set of ϕ values vary depending on set limits of x_{min} and x_{max},which depends on, 1. $[-1,1]$ => $[-1,1]$ mapping in the polynomial transformation and, 2. The visual accuracy of fit provided by Quantum Rotation gate setting curve of classical Polynomial Transformation Curve, we can try running the code in the loop each time varying the x_{min} and x_{max} by a small margin and obtaining the new set of ϕ values and selecting the sets of ϕ values that provide the best visual fits to classical

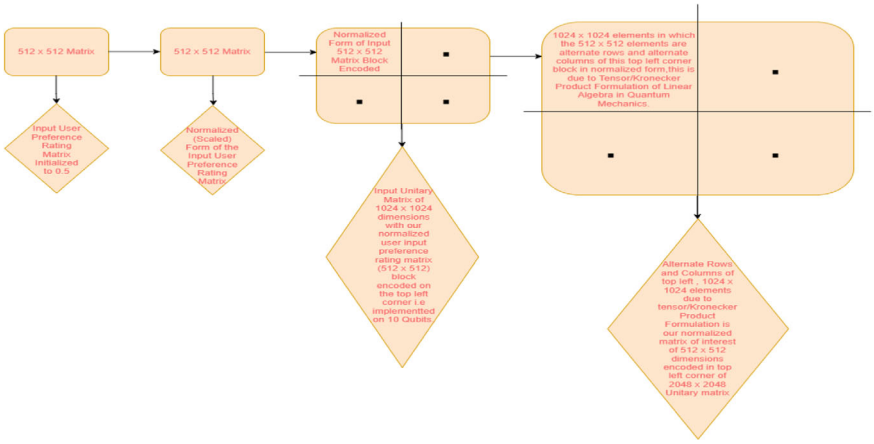


Fig. 12 Schematic explanation of the 512 × 512 input user preference rating matrix conversion to 2048 × 2048 matrix due to Kronecker product of one more qubit (QSVT qubit) with unitary matrix (1024 × 1024) when it is implemented on QSVT circuit

polynomial curve and parameterizing the rotation gates with these sets of ϕ values on QSVT circuit and getting accuracy of implementation, if it is better than or equal to classical implementation stop, if it is not, run the QSVT Circuit for new set of ϕ values and repeat the process. Following this loop iterating, we have discovered the set of ϕ values that provide accuracy better than the classical setting for emulation on Quantum Setting using rotation gates.

4 Classical Post Processing

The Polynomial Transformation of the unitary input user rating matrix implemented on QSVT is a 2048 × 2048 Matrix.QSVT emulation requires a minimum of 2 Qubits more than the original input user rating matrix of 512 × 512 dimensions, requiring 9 Qubits termed as Recommendation Qubits. One Qubit is required for the input user preference rating matrix initialized to 0.5 to a unitary matrix conversion, hence our normalized input user preference matrix of 512 × 512 dimensions is block encoded in top left corner of 1024 × 1024 dimensions unitary form of the input user preference rating matrix. Now for performing quantum gate unitary rotation operations on this unitary matrix, one more qubit is needed known as the QSVT Qubit. So in total for an implementation of a 512 × 512 matrix on a QSVT, a minimum of 11 Qubits are needed (9 Recommendation Qubits + 1 Ancilla Qubit + 1 QSVT Qubit.), so our scaled 512 × 512 matrix is encoded in a 2048 × 2048 matrix. Also after Quantum Fit, mapping is [real] to [real + imaginary] and hence classical post-processing is required 1. To take the absolute value of the reconstructed unitary matrix, 2. To

Fig. 13 Benchmarking time, memory, and accuracy for quantum implementation

Benchmarking	Classical Implementation	Quantum Implementation
Execution time of Single State-Vector Simulator	-	2.0794739723205566 seconds
Execution time of Numpy library for Classical /Execution time of Unitary Simulator	0.06593823432922363 seconds	62.07271218299866 seconds
Execution time of State-Vector ,Unitary , QASM Simulator	-	82.2 seconds
Memory Requirement in terms of the Number of Total Computations (Multiplications)	48,13,18,38,208.	3,52,18,73,18,272.
Accuracy (RMSE Value)	0.2238	0.1290

Fig. 14 Benchmarking accuracies for quantum implementation/emulations with respect to all the classical algorithms used

Techniques/ Error Parameters	Global Average Rating Error	User Average Rating Error	Movie Average Rating Error	Matrix Collaborative Filtering Rating Error	Polynomial Transformation Rating (My Method) Error Classically	Polynomial Transformation Rating Implemented on QSVT
R.M.S.E.	0.2093	0.3890	0.2598	0.2961	0.2238	0.1290
M.A.E.	0.1701	0.2970	0.2020	0.2375	0.1778	0.0842
R - Squared	-0.0005	-2.4568	-0.5419	-1.0030	-0.1444	0.6197
Pearson Corelation Coefficient	0.0000	-0.0068	0.0285	0.0164	0.0486	0.8863
P - Value	0.0000	0.4647	0.0023	0.0805	0.0000	0.0000
M.A.P.E	36.4277 %	50.5610%	41.9237%	43.4729%	33.1973%	13.6970 %

extract out the 512×512 elements from the 2048×2048 matrix, which are alternate rows and columns of the top left 1024×1024 elements of 2048×2048 matrix, 3. Dividing the normalized matrix by a scaling factor to obtain the reconstructed user preference rating matrix, which can be done on a unitary matrix if 'unitary simulator' is used for emulation or state vector if 'state vector simulator' is used for emulation.

5 Benchmarking, Conclusion, and Future Scope

Figures 13 and 14 benchmarks the implementation of the Recommendation System on the Quantum Emulator is **57.64% more accurate** than the best achieved by classical implementation, marking the Quantum Advantage in terms of Accuracy. However, time-wise and memory-wise, the Quantum Implementation is **941.5 times slower** and **72.88 times less efficient**, respectively, compared to the classical implementation.

We have shown a successful emulation of the Recommendation System on the QSVT Framework of QiML by novel polynomial transformation equation and shown a quantum advantage in terms of accuracy when compared to classical recommendation systems and algorithms. This marks the successful real-world problem-solving emulation of Quantum Systems on CPU. The future Scope of this work is benchmarking the Accuracy, Time, and Memory of all other Recommendation data sets

with respect to Classical Implementations. Emulation of QSVT Framework of QiML on GPU, FPGA, and VECTORs (HPC) to show acceleration in terms of time and memory of Quantum Emulations.

References

1. Vidiyala R (2020) Building a movie recommendation system. 2 Oct 2020. https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109
2. Chen D (2020) Recommender system—matrix factorization. 8 July 2020. https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b
3. Larry H, Jin H, Ajmal M, Hajime S, Yanqiu W, Seyit C (2023) Quantum-inspired machine learning: a survey
4. Camps D, Lin L, Van Beeumen R, Yang C (2022) Explicit quantum circuits for block encodings of certain sparse matrices
5. Gilyén A, Su Y, Low GH, Wiebe N (2019) Quantum singular value transformation and beyond exponential improvements for quantum matrix arithmetics. [arXiv:1806.01838 [quant-ph]]
6. Chao R, Ding D, Gilyen A, Huang C, Szegedy M (2003) Finding angles for quantum signal processing with machine precision. [arXiv:2003.02831 [quant-ph]]
7. Martyn JM, Rossi ZM, Tan AK, Chuang IL (2021) Grand unification of quantum algorithms. PRX Quantum 2(4):040203. https://doi.org/10.1103/PRXQuantum.2.040203 [arXiv:2105.02859 [quant-ph]]
8. Qiskit Q contributors (2023) Qiskit: an open-source framework for quantum computing. 10.5281/zenodo.2573505, GitHub repository, https://github.com/Qiskit/qiskit
9. iChuang AT, Martyn J (2022) Python quantum signal processing (pyqsp). (8 Jul 2022), GitHub repository. https://github.com/ichuang/pyqsp
10. Dong Y, Meng X, Whaley K, Lin L (2020) Efficient phase factor evaluation in quantum signal processing
11. MovieLens Dataset, Recommended for Education and Development (2023) "MovieLens Latest Datasets". https://grouplens.org/datasets/movielens/latest/
12. Jeongwan H (2019) Product decomposition of periodic functions in quantum signal processing. Quantum 3:190. 10.22331/q-2019-10-07-190

A Machine Learning-Based Approach for Soil Chemical Property Estimation Using Multispectral Data in Hungary

Sourav Seal and Divyashikha Sethia

Abstract Soil is essential to the environmental process and growing crops. An accurate assessment of soil chemical characteristics is essential for agricultural and land management decisions to be well-informed. Precision agriculture has advanced significantly because of machine learning. It uses automation and data-driven insights to improve several facets of agricultural methods, fostering effectiveness, sustainability, and productivity. The "Land Use/Cover Area Frame Statistical Survey Soil" (LUCAS) is a comprehensive and frequent topsoil survey conducted throughout the European Union to get data pertinent to policy about how land management affects soil properties. The data covers 28 European Union States. This paper analyzes and predicts the chemical properties of the soil of Hungary based on the LUCAS 2015 dataset that includes calcium carbonates ($CaCO_3$), nitrogen (N), phosphorus (P), potassium (K), electrical conductivity (EC), pH. It estimates them using LUCAS and Landsat 8 satellite images by using different regression-based algorithms like partial least square regression (PLSR), Gaussian process regression (GPR), support vector regression (SVR), multilayer perceptron (MLP), AdaBoost, ridge and compares them. The Lucas survey data points and Landsat 8 satellite images (multispectral) are integrated for forecasting different soil nutrients, expanding the model's relevance to diverse agricultural and environmental applications. The algorithms undergo evaluation, producing acceptable Root Mean Squared Error (RMSE) and Ratio of Performance to Prediction (RPD) values. The proposed approach can estimate the chemical properties of the soil samples in Hungary replacing the orthodox method of laboratory analysis with the soil samples' spectral image.

Keywords Multispectral · Regression · Soil chemical properties · machine learning · Precision agriculture · LUCAS

S. Seal (✉) · D. Sethia
Department of Software Engineering, Delhi Technological University, Delhi, India
e-mail: sourav_2k22dsc16@dtu.ac.in

D. Sethia
e-mail: divyashikha@dtu.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_26

1 Introduction

Soil properties estimation in precision agriculture allow for the precise application of water, fertilizers, and pesticides. By tailoring inputs to the specific needs of each area within a field, farmers can optimize resource use, reducing waste, and costs. The soil properties' estimation lessens the adverse environmental effects of farming practices and helps to slow down the depletion of natural resources [1]. They are using techniques for remote sensing, such as satellite imagery. Farmers may monitor crop conditions and spot pests using drones and a field's hot spots for stress or illness. This knowledge allows for prompt measures like targeted irrigation or pest control, which may increase crop yields and decrease losses.

Digital soil mapping gained prominence over the past decade as a valuable tool for acquiring and analyzing spatial soil data [2]. Multiple applications drive this demand, including agriculture, land use planning, environmental management, and infrastructure development. Ballabio et al. [3, 4] proposed a physical-chemical properties estimation of the Lucas 2012 dataset. In addition to it, Riese et al. [5] proposed a soil texture classification based on the LUCAS 2012 dataset [6].

Researchers collect data using different methods like satellite images or drones. The images collected by satellite are primarily to cover huge areas such as a country. The term "band" refers to the layers in an image, such as a satellite image or an image from a digital photograph. Different open-source satellite data include Modis Landsat, which captures multispectral data from unique spectral bands. Multispectral data provides distinct bands to capture different soil properties based on their distinct spectral footprints.

Owing to its ability to capture data from significant datasets and produce inferences, machine learning, a branch of artificial intelligence, has become a potent method for assessing soil nutrients. Bing Lu et al. [7] reviewed different machine learning deep learning-based approaches adopted by researchers to estimate and classify different soil and plant properties. Shagun et al. [8] reviewed various machine learning approaches for soil properties estimation.

Ballabio et al. [3] focused their research on estimating the chemical properties of all 25 countries in the LUCAS dataset using Modis Satellite using LUCAS data points of 2012 survey. In contrary, the current study directs its focus to capture the chemical soil properties of a particular country, Hungary, from the LUCAS 2015 dataset using Landsat 8 satellite. The LUCAS 2018 survey is just an extended version of the 2015 dataset, and some of the land classification is added without any modification or addition of data points. This study focuses on a particular region for better estimation of the chemical properties in Hungary and also captures enhanced information introduced in the LUCAS 2015 dataset [6].

This paper utilizes multispectral data from the Landsat 8 satellite to estimate the chemical soil nutrients for Hungary. It focuses on remote sensing and machine learning to estimate chemical properties, helping agricultural professionals and researchers make well-informed decisions without physically testing the soil in the laboratory. It estimates $CaCO_3$, N, P, K, EC, pH from LUCAS topsoil data 2015 [6].

It consists of data from 28 EU countries consisting of ground truth values of soil chemical and physical properties, coordinates of the points, and a shape file of the points collected. It is crucial for definative prediction models.

Contributions: The major contributions of this paper are :

- It presents a comparative study of different machine learning algorithms using open-source satellite image data from Landsat 8 and ground truth values from the LUCAS 2015 dataset for Hungary.
- Estimation of the chemical soil properties in Hungary Region with limited data with the chemical properties of full 28 European countries for a region-specific novel comparison.

The corresponding sections deep dive about each step. Section 2 explores the previous work on the LUCAS dataset. Section 3 enlists the research methodology by exploring the data description, preprocessing, feature extraction, model building, and evaluation technique. Section 4 examines the result by exploring models like GPR, MLP, PLSR, AdaBoost, and ridge and compares the result with previous work on this dataset. The paper ends with a conclusion in Sect. 5.

2 Previous Work on LUCAS Dataset

Many researchers have attempted to explore multispectral data from satellites and used it for soil analysis and prediction of soil features and crop type classification. The LUCAS dataset is a significant leap due to its vast coverage of 28 countries and actual ground truth values. This section attempts to capture all the significant work previously done in this area. Table 1 refers to the researchers' previous work using the LUCAS dataset and satellite images combined to produce significant predictions about the soil properties. Ballabio et al. [3] provided a GPR regression technique to estimate the chemical properties of the soil based on Modis satellite data and LUCAS 2012 topsoil data.

The chemical properties include $CaCO_3$, N, P, K, EC, and pH. Ballabio et al. (2016) also provided Multivariate Adaptive Regression Splines (MARS) to predict the physical soil properties by capturing data from Modis satellite and LUCAS data points from 2012 database. The physical properties include sand, clay, silt, and gravel. In both the above papers, the authors achieved good accuracies. Pflugmacher et al. [9] used Landsat 8 satellite data and LUCAS 2015 data to classify land/cover classes across Europe using random forest. The overall accuracy achieved was 75.1%. Riese et al. [5] also classified the soil texture using the LUCAS (2012) dataset using 1-D convolution neural network (CNN) with an accuracy of 74%. Ghassemi et al. [10] classified crops using Sentinel 2 satellite and LUCAS 2018 dataset using SVM, achieving an accuracy of 76.8 %. All the researchers utilized all the countries of the European Lucas dataset for their estimation or prediction of different soil or crop types.

Table 1 Comparison of machine learning and deep learning methods on LUCAS using multispectral data

Previous study	Lucas dataset year	Research focuses	Accuracy
Pflugmacher	2015	Mapping pan European data using Landsat8 data and LUCAS dataset	Random forest:75.1%
Ballabio	2012	Predicting LUCAS topsoil physical properties using Modis Sattelite	Multiplicative adaptive Regression splines(MARS): R^2 = 0.47–0.5
Ballabio	2012	Predicting LUCAS topsoil chemical properties using Modis Sattelite	Gaussian process regression(GPR): RMSE = 0.78–121.89
Ghassemi	2018	Crop type classification using Sentinel 2 and LUCAS dataset	SVM : 76.8%
Riese	2012	Soil texture classification based on hyperspectral data using LUCAS dataset	CNN : 74%
Castaldi	2012	SOC prediction using XDS Rapid content analyzerspectrora-diometer	PLSR : RMSE–3.7
Castaldi	2012	SOC prediction using normalized burn ratio (NBR)for North-Eastern Germany	NBR : RMSE—6.8–26.6 RPD— 0.9–4.4

Many researchers have estimated the soil properties of specific countries for better estimation accuracy. Castaldi et al. [11] predicted Soil Organic Carbon (SOC) using the XDS Rapid content analyzer spectrometer for Belgium and Luxembourg using the LUCAS 2012 dataset. The study divided the soil samples into 12 distinct classes. SOC was estimated using a regression-based approach with an acceptable RMSE. Castaldi et al. [12] also used Sentinel 2 satellite data along with the LUCAS 2012 dataset to estimate SOC for North East Germany. The study also used a regression-based technique to estimate SOC and achieved an excellent RMSE score.

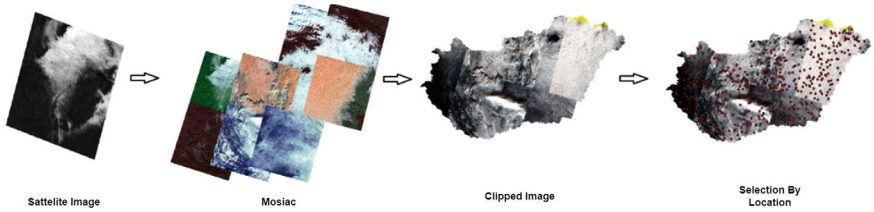


Fig. 1 Image processing step

3 Methodology

This section covers the methodology of the models to estimate soil nutrients. Figure 2 describes the methodology adopted in this paper.

3.1 Dataset Description

Land Use/Cover Area frame Statistical Survey (LUCAS) 2015 dataset [6] is used in this survey. Also Landsat 8–9 OLI/TIRS C2 L2 satellite captures images in the multispectral frequency. This study uses the dataset for Hungary. It estimates the chemical properties $CaCO_3$, N, P, K, EC, and pH from the LUCAS dataset.

LUCAS 2015 topsoil dataset covers 28 European Union countries with ground truth data. The LUCAS dataset consists of a ground truth CSV file and a shp file comprising all the collected data points. Approximately 22000 data samples were collected from all of the 28 European countries. In this study, we only use a subset of the data of Hungary that consists of 412 points. It consists of all the ground truth values of organic carbon content, soil texture, soil type, nutrient accumulation, and pH levels. It offers extensive information about the soil features in Europe.

The Landsat 8 satellite captures image from Hungary. The data used is a satellite image 2015 (same as LUCAS). It is a multispectral dataset that captures different electromagnetic spectrum bands [13].

Figure 3 represents the key map of Hungary using the Inverse Distance Weighting (IDW) interpolation technique, which shows the chemical property of the soil EC. The maps show the distribution of the nutrient levels from low to high. The lighter tone represents a lower range, while a brighter tone represents a high range. A specific color denotes the scope of the nutrient value present in that region.

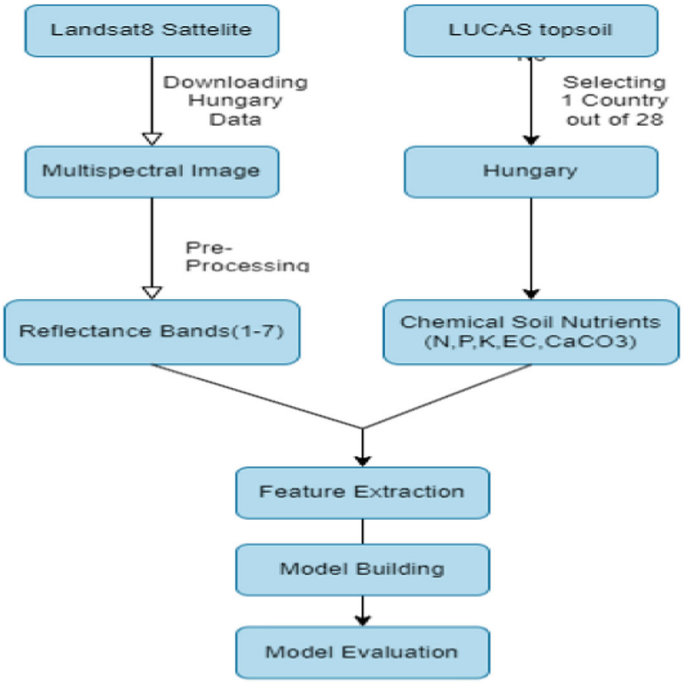
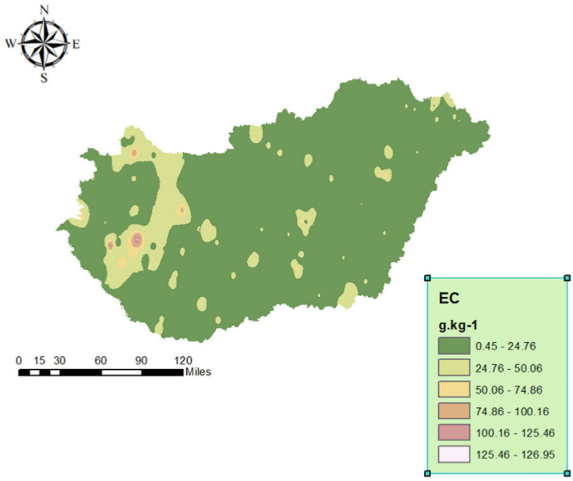


Fig. 2 Flow diagram for research methodology

Fig. 3 Topsoil map EC



3.2 Satellite Data Processing

Data processing for Landsat 8 satellite image is critical to accurately predict soil properties. This section discusses the essential steps to process satellite images: band composition, mosaicing, clipping, selection by location, and extracting multi-value to points. Figure 1 describes Landsat 8 multispectral data processing steps for a broad area and combining it for a target area.

- *Band Composite*: The multispectral Landsat 8 data consists of 7 bands. For each band, a tiff image is present for a particular tile. The seven different images are combined into a single image using this step.
- *Mosaicing*: Multiple tiles are needed for a broad coverage area. Each tile has data that has seven images that are combined using the previous step. The total tiff images are the same as the number of tiles. This step combines all the tiff images into one single tiff file.
- *Clipping*: The data obtained from the previous step is raw data from the specified region with outliers. The shape file of Hungary is superimposed on the raw data to trim out the target region (Hungary). The excess portion is clipped out to get only the area of interest.
- *Selection By Location*: The clipped data outlines the the shape file from LUCAS topsoil 2015 data. The collected data points are marked on the image and spatially aligned. It ensures optimal comparison and integration of the satellite image with the ground data.
- *Extracting MultiValue to points*: The final tiff image now consists of the relevant data and features of the LUCAS data and the band information of Landsat 8 satellite data. This paper converts the LUCAS data and the bands into point values, where each tuple consists of the soil features and the reflectance values of each of the seven bands.

3.3 Materials and Methods

3.3.1 Reflectance Calculation and Normalization

This paper calculates the reflectance index from Landsat 8 data. The OLI sensor in Landsat 8 captures the images for 7 bands (Band 1–7). The data is downloaded from the United States Geological Service (USGS). The data is converted into 16 bit Digital Numbers (DN) from 0 to 65535. The dataset normalization is done using Standard Scalar, which normalizes bands in the range of 0–1. This method facilitates the data to be converged and free from outliers for the model to be applied.

3.3.2 Feature Selection

Landsat 8 sattelite has 7 bands and a spatial resolution of 30 m. The representation is as follows Band 2: Blue , Band 3: Green, and Band 4: Red , Band 1: indigo for Aerosols/Coastal and Band 5: Near Infrared Rays(NIR). Water in the leaves of healthy plants reflects this portion of the spectrum. Bands 6 and 7 represent Short Wave Infrared Rays (SWIR)-I and II for identifying dry and moist regions. This study employs all the 7 bands from the satellite.

3.3.3 LUCAS Data Extraction

The principal nutrients of this paper are the chemical properties $CaCO_3$, N, P, K, EC, and pH. The Lucas topsoil dataset from 2015 [6] served as this study's ground truth data source. It estimates the soil fertility and other important aspects of the ground. The novelty of the paper focus is comparing how the region-specific estimation with less data points is conducted and how it performs in contrast with the overall LUCAS dataset by different algorithms and comparing the accuracy. So out of 22,000 soil sample data, 412 data are extracted and combined with the reflectance values of bands extracted from Landsat data.

3.4 Model Description

This paper proposes and compares different machine learning algorithms such as partial least square regression (PLSR), Gaussian process regression (GPR), support vector regression (SVR), multilayer perceptron (MLP), AdaBoost, and ridge. The regression algorithm performs specific steps, including data cleaning, feature selection, data preprocessing, normalization, and splitting data into train and test sets in 7:3 ratio. Further steps include an algorithm selection following training and scoring the model by evaluation metrics (Root Mean Squared Error and Ratio of Performance to Prediction). The methodology adopted in this paper is to repeat the steps for all the tested algorithms. Figure 4 shows the regression methodology adopted in this paper. All the learning algorithms are supervised and implemented using the sklearn python library [14].

3.5 Hyperparameter Tuning and Cross-Validation

Hyperparameters and model parameters make up a regression model's parameters. Hyperparameters are predetermined, whereas the training step involves fine-tuning the model parameters. The hyperparameters' optimal setting varies depending on the soil nutrient and its variation with the bands' reflectance value. The method used

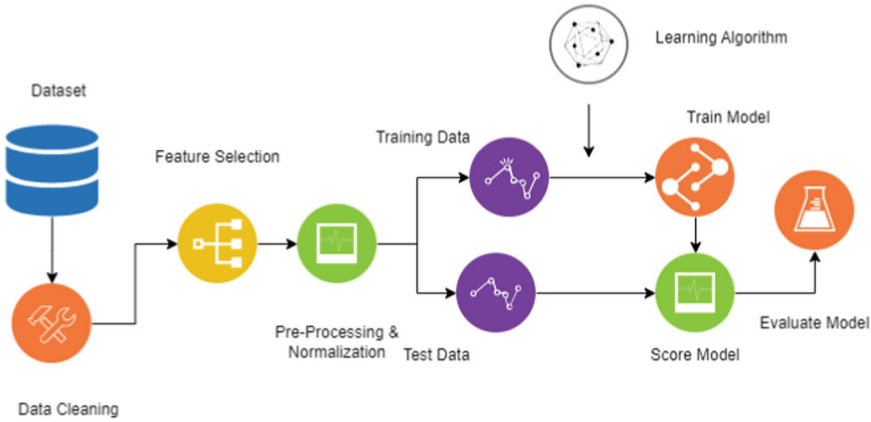


Fig. 4 Regression algorithm flow

in this paper obtains the optimal hyperparameter setting by a primary grid search method and extracts the best parameter. Table 2 points out all hyperparameters used in this study for the different machine learning and deep learning models including every chemical properties.

The paper also uses a fivefold cross-validation to ensure no bias is present while predicting the results.

3.6 Performance Metrics

The evaluation metrics chosen in the model are the Root Mean Squared Error (RMSE) and Ratio of Performance to Deviation for evaluating the model performance (RPD). The RMSE is a quadratic scoring rule gauges the average error's magnitude. RPD is a relationship between range and standard deviation. A good performance indicates a low value of RMSE while a high value of RPD. RMSE ranges from 0—any finite range [15], while RPD values usually range above 2 [15].

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(predicted - actual\right)^{2}}$$

$$RPD = RangeReference/StandardDeviation$$

Table 2 Results of chemical property estimation by regression models

Electrical conductivity(EC)			Calcium carbonate (CaCO3)		
Model	RMSE	RPD	Model	RMSE	RPD
PLSR	16.86	7.25	PLSR	75.37	4.93
SVR	19.15	6.37	SVR	73.91	5.11
MLP	19.36	7.17	MLP	78.37	4.94
GPR	17.07	7.16	GPR	72.48	4.91
AdaBoost	16.92	7.19	AdaBoost	72.45	4.99
Ridge	16.73	7.27	Ridge	73.9	4.89
Ballabio et al. [3] GPR	11.02	–	Ballabio et al. [3] GPR	78.29	–
Pottasium (K)			Phosphorus (P)		
Model	RMSE	RPD	Model	RMSE	RPD
PLSR	201.41	7.32	PLSR	23.77	4.88
SVR	207.37	7.19	SVR	24.37	4.83
MLP	208.56	7.37	MLP	23.75	3.56
GPR	199.42	7.3	GPR	23.97	4.86
AdaBoost	199.81	7.24	AdaBoost	23.85	4.88
Ridge	197.16	7.33	Ridge	23.69	4.89
Ballabio et al. [3] GPR	199.17	–	Ballabio et al. [3] GPR	17.52	–
Nitrogen (N)			pH H2O		
Model	RMSE	RPD	Model	RMSE	RPD
PLSR	1.21	7.3	PLSR	0.93	4.54
SVR	1.31	5.91	SVR	1.28	3.49
MLP	1.20	7.45	MLP	0.98	4.69
GPR	1.05	7.44	GPR	0.99	4.52
AdaBoost	1.08	7.22	AdaBoost	0.95	4.59
Ridge	1.12	7.07	Ridge	0.93	4.69
Ballabio et al. [3] GPR	0.78	–	Ballabio et al. [3] GPR	0.78	–
pH CaCl2					
Model	RMSE	RPD			
PLSR	0.88	4.22			
SVR	1.22	3.1			
MLP	0.89	4.15			
GPR	0.91	4.06			
AdaBoost	0.87	4.24			
Ridge	0.88	4.22			
Ballabio et al. [3] GPR	0.68	–			

4 Results and Discussion

This paper consists of three techniques:

- Processing Landsat 8 data for Hungary.
- Combining it with LUCAS data for Hungary.
- Develop and compare the different machine learning models.

The regression models, namely PLSR, SVR, GPR, MLP, AdaBoost, and ridge, predict the chemical properties of soil, EC, $CaCO_3$, K, N, P, pH_h_2O and pH_$CaCl_2$. RMSE and RPD are the methods used to evaluate and validate the models. This study reveals the prediction of the chemical properties of soil in Hungary.

Table 2 contains all the regression results for different soil nutrient predictions. PLSR, GPR, AdaBoost, and ridge Regressors perform comparably. The regression results show that the soil nutrients EC, P, N, pH_H_2O, pH_$CaCl_2$ can be predicted with reasonable accuracy as the RMSE value is low and the RPD value is high. In comparison, SVR and MLP do not perform consistently for all the properties. The R^2 for the nutrients is of the range of 0.2–0.3, due to the less number of points available for multispectral data(400 points). Whereas Ballabio et al. [4] used approximately 22,000 points to predict the chemical properties of the overall LUCAS dataset. EC (RMSE = 16.73 and RPD = 7.19), P(RMSE = 23.69 and RPD = 4.89), N(RMSE = 0.93 and RPD = 7.07),

K(RMSE = 197.16 and RPD = 7.33), pH_H_2O(RMSE = 0.93 and RPD = 4.69) achieved best results using ridge Regressor. While $CaCO_3$(RMSE = 72.45 and RPD= 4.89) and pH_$CaCl_2$(RMSE = 0.87 and RPD = 4.24) achieved best results using AdaBoost regressor. GPR is computationally more complex than AdaBoost with a cubic linear time complexity. GPR models are less interpretable in this study due to their nonparametric nature and complex function representations.

The study fully explains the various regression models' propensities to predict soil properties. Ballabio et al. [4] accuracied mentioned in Table 2 are of the full LUCAS dataset consisting of approximately 22,000 points. At the same time, this paper focuses on predicting soil chemical properties in Hungary only with only 400 points. This study proposes an AdaBoost algorithm to predict the chemical properties when working with a smaller dataset. The GPR algorithm, due to its complex function, is unable to perform well on a smaller dataset. AdaBoost performs better and more efficiently than GPR on a region-specific survey with minimal data points. This work enhances the science of soil analysis by creating prediction tools with substantial implications for agricultural and environmental research.

5 Conclusion

This paper discusses estimating soil chemical properties by regression using Landsat 8 multispectral data. Different machine learning algorithms perform the regression on multispectral data combined with LUCAS. It accurately predicts the chemical properties of the soil of Hungary. Such region-specific analyses can yield valuable insights and recommendations tailored to that area's unique soil and environmental conditions. Ridge and AdaBoost outperform all the machine learning models and previous studies [3]. The models' predictive capacity may be useful for monitoring land, the environment, and precision agriculture. This work presents the thorough processing that goes into merging and analyzing satellite photos with soil characteristics.

This study can address the specific challenges and opportunities related to soil and land management of the Hungary region along with digital soil mapping and spatial analysis of the region. Future Research may examine long-term trends of soil properties, the impact of climate change on Hungarian soils, or the development of site-specific management practices.

References

1. Weiss M, Jacob F, Duveiller G (2020) Remote sensing for agricultural applications: a meta-review. Remote Sensing of Environ 236:111402
2. McBratney AB, Santos MM, Minasny B (2003) On digital soil mapping. Geoderma 117(1–2):3–52
3. Ballabio C, Lugato E, Fernández-Ugalde O, Orgiazzi A, Jones A, Borrelli P, Montanarella L, Panagos P (2019) Mapping LUCAS topsoil chemical properties at European scale using Gaussian process regression. Geoderma 355:113912
4. Ballabio C, Panagos P, Monatanarella L (2016) Mapping topsoil physical properties at European scale using the LUCAS database. Geoderma 261:110–123
5. Riese FM, Keller S (2019) Soil texture classification with 1d convolutional neural networks based on hyperspectral data. arXiv preprint arXiv:1901.04846
6. Tóth G, Jones A, Montanarella L (2013) The LUCAS topsoil database and derived information on the regional variability of cropland topsoil properties in the European Union. Environ Monitoring and Assessment 185:7409–7425
7. Lu B, Dao PD, Liu J, He Y, Shang J (2020) Recent advances of hyperspectral imaging technology and applications in agriculture. Remote Sens 12(16):2659
8. Jain S, Sethia D (2023) A review on applications of artificial intelligence for identifying soil nutrients. pp 71–86
9. Pflugmacher D, Rabe A, Peters M, Hostert P (2019) Mapping pan-European land cover using Landsat spectral-temporal metrics and the European LUCAS survey. Remote Sens Environ 221:583–595
10. Ghassemi B, Dujakovic A, Żółtak M, Immitzer M, Atzberger C, Vuolo F (2022) Designing a European-wide crop type mapping approach based on machine learning algorithms using LUCAS field survey and Sentinel-2 data. Remote Sensing 14(3):541
11. Castaldi F, Chabrillat S, Chartin C, Genot V, Jones A, van Wesemael B (2018) Estimation of soil organic carbon in arable soil in Belgium and Luxembourg with the LUCAS topsoil database. Europ J Soil Sci 69(4):592–603

12. Castaldi F, Chabrillat S, Don A, van Wesemael B (2019) Soil organic carbon mapping using LUCAS topsoil database and Sentinel-2 data: an approach to reduce soil moisture and crop residue effects. Remote Sens 11(18):2121
13. Acharya TD, Yang I (2015) Exploring landsat 8. Int J IT Eng Appl Sci Res (IJIEASR) 4(4):4–10
14. Oliphant TE (2007) Python for scientific computing, computing in science and engineering
15. Gholizadeh A, Žižala D, Saberioon M, Borvka L (2018) Soil organic carbon and texture retrieving and mapping using proximal, airborne and sentinel-2 spectral imaging. Remote Sens Environ 218:89–103

Machine Learning-Based Network Attacks Detection

Muthukumarapandian Chandrasekaran and Chetana Tailor

Abstract This paper focuses on detecting network attacks using anomaly-based identification systems. The research introduces robust machine learning techniques to identify network intrusion detection including its type using the dataset, namely, UNSW-NB15. It covers nine distinct attacks with the help of 49 features. The decision tree classifier has given the best result and noted accuracy is 99.05% among other techniques, namely, random forest, Adaboost, XGboost, KNN, and SVM. The study emphasized the relevance and strength of all features in network attack detection, obviating the need for feature selection.

Keywords Network security · NIDS · Ensemble learning · Machine learning

1 Overview

The substantial surge in the domain of network-related data can be ascribed to several factors, including the Internet of Things (IoT), adoption of cloud-based services, increase the use of video and social media, and the widespread proliferation of networked devices [1]. The surge in network-related data is a double-edged sword. On the one hand, it provides us with a wealth of information that can be used to improve our understanding of the world and to make better decisions. On the other hand, it also creates new opportunities for cybercriminals to exploit. While adding more security devices can enhance network security, it cannot assure complete safety. Existing Network Intrusion Detection Systems (NIDS) offer multilayered security protections for applications, systems, networks, and data transmission [2]. This layered approach ensures that if one layer of security is penetrated, subsequent layers will prevent attackers. Modern NIDS systems encounter numerous obstacles, which encompass

M. Chandrasekaran
Ascendion Inc., Bernards, NJ, USA

C. Tailor (✉)
Ascendion Engineering Private Limited, Bangalore, India
e-mail: chetana.tailor@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_27

issues like inadequate accuracy, the dynamic patterns of network traffic, infrequent network attacks, adaptability to software-defined networks, the immense quantities of stored and transmitted data, as well as the wide array of network access devices. Modern NIDS trend is either anomaly or signature-based identification systems. Signature-based NIDS exclusively identifies established threats and the telltale signs associated with them [3]. While it excels in processing speed and accuracy when dealing with known attacks, it encounters difficulties in recognizing zero-day attacks. Its effectiveness is contingent on the applications, operating system, and version [4]. In contrast, anomaly-based NIDS can identify abnormal behaviors that deviate from normal patterns. This approach excels at detecting zero-day attacks but generates more false-positives, necessitating additional efforts to investigate alerts [1].

NIDS systems built using machine learning (ML) and deep learning (DL) techniques utilize classification models trained on a varied collection of network data samples [5]. This training enhances their capacity to accurately categorize attacks [6]. DL-based models outperform at identifying attacks with a reduced rate of false alerts [7, 8]. The cyber attackers operate advanced methods to take advantage of vulnerabilities in computing resources. So, in this article, authors proposed a robust method to prevent network security attacks using AIML. The remaining paper is organized as Literature review, Experiment, and Conclusion of the research work.

2 Literature Study

This section focuses on types of network attacks, and its detection techniques covering signature-based detection and anomaly-based NIDS.

2.1 Network Attacks

Cyberattacks on computer networks can have critical consequences, impacting the entire system's operation. These attacks can result in unauthorized access to data, damage to system integrity, and theft of valuable information [9]. Preintrusion activities like IP snoofing and port scanning take place before the attacks. The main role of NIDS is to monitor network traffic, detect known attack patterns, recognize, and flag potential security breaches, and provide detailed reports on detected security incidents. Detection of attacks relies on the analysis of attributes including sender and receiver's IP addresses, port numbers, header information, and protocol details. These attributes are used to categorize attacks into two main types: passive and active [2]. Passive attacks involve the monitoring of networks with the intention of extracting confidential data, while active attacks exploit security vulnerabilities by circumventing established security measures. Passive attacks are often considered more challenging to detect than active attacks due to their covert nature, making them

harder to identify and mitigate. In the following section, attack-related literature with ML literature is discussed:

Fuzzers launch attacks by injecting a large volume of random or unexpected data into a system, with the goal of inducing failures and discovering software or system vulnerabilities. This technique is effective at uncovering weaknesses in networks and operating systems, as it can reveal security loopholes and potential points of exploitation [10].

ML models are instrumental in identifying and countering various web application penetration techniques like port scanning, spam emails, malicious HTML code penetration, helping to enhance cybersecurity, and protect against these types of attacks [11, 12].

Attack, namely, Backdoor deals with security mechanisms, enabling unauthorized entry into computer systems and access to their data [13]. Their primary objective is to endanger user privacy and disrupt the computing resources availability.

DoS (Denial of Service) attacks disrupt network resources, rendering them unavailable to users by interrupting services [12]. Verisign has reported a significant surge in the frequency and sophistication of DoS attacks, highlighting the need for robust NIDS systems that leverage ML and DL models to defend against these threats.

Cyber adversaries exploit software vulnerabilities to seize control of resources such as computers and network data, frequently leading toward system collapses or operational malfunctions. In the case of zero-day exploits, these attacks target vulnerabilities in software that vendors are unaware of or have not yet released patches for.

Generic attacks target block ciphers without depending on an understanding of the internal structure of these ciphers [12]. Since block ciphers have limitations on the length of keys and blocks, they are susceptible to generic attacks. Identifying generic attacks requires the careful selection of appropriate external parameters. A few instances of generic attacks on block ciphers encompass dictionary attacks, exhaustive key searches, rainbow table attacks, and similar techniques [14].

Reconnaissance attacks entail the comprehensive gathering of the target system information in advance of initiating the attack, serving as a preparatory phase for the subsequent offensive actions. These reconnaissance attacks can be broadly classified into three primary types: social, public, and software. In these types of attacks, information is gathered through various techniques, including port scanning, packet sniffing, ping sweeps, and queries for internet information [15].

Shellcode serves as the payload, comprising a concise code segment designed to exploit software vulnerabilities. It facilitates the execution of a command interpreter, enabling interactive command input and the retrieval of output on vulnerable systems [16]. Detecting shellcode attacks can be achieved through runtime statistics that capture machine-level operations.

A worm detection system aims to provide early warnings and minimize reaction time for countermeasures. It achieves this by analyzing important features such as content and format of payload, header information of a packet, traffic patterns in a network, and observing the behavior of hosts in the network [17].

2.2 Signature-Based Detection

Four key components of NIDS are complimentary signature database, frequent signature database, intrusion detection system (IDS), and updating agent [18]. IDS operates by extracting signatures from network packets, which are then compared against signature databases. If a match is found, the system triggers an alert. These four components are designed to achieve early and precise detection of attacks, reducing false alarm. Intrusions characterized by atypical signatures are still detected by utilizing stored signatures of the complementary database. The key obstacle in signature-based detection lies in the mitigation of false alarms, a challenge that can be tackled through strategies like signature refinement, stateful signatures, and vulnerability-specific signatures [19].

2.3 Anomaly Based NIDS

The authors conducted an extensive statistical analysis of features and observation in their research [20]. They employed various statistical tests, including the Kolmogorov–Smirnov test, multivariate kurtosis, and multivariate skewness, to assess data characteristics. Additionally, they assessed feature relevance through Gain Ratio and Pearson's correlation coefficient. In their evaluation, they used the UNSW-NB15 dataset to test the performance of existing classifiers. In their assessment, they employed the UNSW-NB15 dataset to gauge the effectiveness of established classifiers. Among these classifiers, the decision tree (DT) classifier excelled, achieving a noteworthy accuracy of 85.56% by reducing the false alarm by 15.78%.

The most influential features for reducing the impurity in the dataset include ct_srv_src, ct_srv_dst, ct_dst_src_ltm, ct_dst_sport_ltm, and ct_src_dport_ltm. In binary classification for attack detection, support vector machine (SVM) gives an accuracy of 82%, which is better as compared to Gradient Boost and Logistic Regression. For attack classification, the decision tree C5.0 has given good accuracy as compared to Naive Bayes and SVM [21].

The deep neural network (DNN) exhibited acceptable results in classifying three different categories, namely, DoS, Normal, and Probe. SVM excelled in classifying Normal and four different types of attacks [22].

3 Experiment and Result Discussion

The publicly available dataset with name UNSW-NB15 having network traffic-related labeled data covering normal and nine network attack labels [12]. The training and testing dataset contain 175,341 and 82,332 records, respectively. It contains packet-based features, flow-based features, and statistical features which are used

Fig. 1 Architectural design

to detect and classify the network attacks. Figure 1 shows the overall architectural design of network attack detection.

To identify the network attacks using ML model, authors focus on Adaboost, DT, K-nearest neighbors (KNN), multilayer perceptron (MLP), random forest, and SVM as these techniques are widely used [23]. SVM excels at distinguishing between normal network traffic and attack patterns [24]. It can handle a broad range of features and conduct training swiftly. On the other hand, KNN is categorized as a lazy classifier that predicts class labels by considering the nearest class labels [25]. In contrast, the DT is a multistep decision-making model well-suited for both categorical and numerical data and delivers rapid decisions due to its nested conditional statements [16]. The random forest classifier trains a collection of CART trees for making classification by randomly choosing features to make node decisions and utilizes out-of-bag error estimation. The final label is decided by averaging the separate class label probabilities of each tree [19]. Adaboost combines multiple weak learners and constructs a strong predictive model requiring minimal parameter tuning and reducing the susceptibility to overfitting. XGBoost creates a strong predictive model by combining weak learners modeled through parallel implementation. Additionally, it offers support for pruning, management of missing values, prevention of overfitting, and hardware optimization [26].

Train test split ratio for each model is 80:20. Each model is trained and tested on the same train-test dataset. In the table below, results for top five attack detection models are shown. Accuracy, F1 score, Precision, and Recall are considered as evaluation metrics (Table 1).

As shown in the above table, DT model outperforms all the above-mentioned models with reported accuracy of 99% and F1 score 0.99. The rest of the techniques' accuracy ranges from over 95–98%.

Table 1 Attack detection result analysis

Model	Accuracy (%)	F1 score	Precision	Recall
Adaboost	97.86	0.98	0.98	0.97
Decision tree	99.04	0.99	0.99	0.99
Random forest	98.95	0.99	0.99	0.99
SVM	95.16	0.94	0.96	0.93
XGBoost	98.06	0.97	0.98	0.97

4 Limitations and Future Directions

This study leverages the UNSW-NB15 dataset for network attack detection, employing a combination of packet-based, flow-based, and statistical features. While valuable insights are revealed, it is crucial to acknowledge potential limitations and future research directions for further development.

Dataset Representativeness: The UNSW-NB15 dataset provides a valuable resource for training and evaluating NIDS. However, the inherent characteristics of the dataset may restrict the applicability of the results. Specifically, concerns arise regarding its capability to accurately represent the wide range of real-world network traffic. The potential discrepancy might result in decreased model performance when utilized with datasets featuring notably distinct attack types, traffic patterns, or protocol distributions.

Feature Engineering Exploration: The effectiveness of feature engineering significantly impacts the performance of NIDS models. This study utilizes a combination of packet-based, flow-based, and statistical features for classification. While informative, it is essential to acknowledge that the optimal feature set might vary depending on the specific attack type being targeted. Further exploration of alternative feature engineering techniques and feature selection methods could potentially enhance the model's ability to identify diverse attack scenarios.

To address these limitations and further advance the research in this domain, below points are consider for future exploration:

4.1 Leveraging Diverse Datasets

Expanding the training and testing datasets to encompass a wider range of real-world network traffic scenarios could enhance the model's generalizability and robustness to unseen attack types.

4.2 Advanced Feature Engineering

Investigating alternative feature engineering techniques, such as DL-based feature extraction, could potentially capture more complex relationships within the data and improve model performance.

4.3 Dynamic Feature Selection

Implementing dynamic feature selection methods that adapt to specific attack scenarios could further enhance the model's effectiveness in targeted attack detection.

4.4 Explainable AI (XAI)

Integrating XAI techniques into the model development process can provide valuable insights into the decision-making process of the model, fostering trust and interpretability in its predictions.

By addressing these limitations and exploring the suggested future directions, researchers can contribute to the development of more robust and generalizable NIDS models capable of effectively safeguarding networks against evolving cyber threats.

5 Conclusion

In conclusion, this paper has delved into the application of ML techniques for NIDSs using the UNSW-NB15 dataset. Notably, the DT emerged as the top-performing model, boasting an impressive accuracy of 99.05%, surpassing counterparts like SVM, KNN, random forest, Ada-boost, and XGBoost. This underscores the efficacy of DTs in the context of network attack detection with the specified dataset.

Despite the promising results, it is imperative to acknowledge potential limitations. The UNSW-NB15 dataset's potential inability to fully capture the diversity of real-world network traffic could impact the generalizability of our findings. Furthermore, the selection of features plays a pivotal role in influencing model performance.

To address these limitations, future research should venture into exploring diverse datasets that encompass a broader range of network scenarios. The investigation of advanced feature engineering techniques, such as DL-based feature extraction, holds promise in enhancing the model's capacity to identify various attack types. Additionally, the integration of Explainable AI techniques can contribute to improved model interpretability, fostering trust in its predictions.

In essence, by systematically addressing these considerations, researchers can pave the way for the development of more robust and adaptable NIDSs, ensuring effective protection against the ever-evolving landscape of cyber threats.

References

1. Almutairi A, Abdelmajeed N (2017) Innovative signature based intrusion detection system: parallel processing and minimized database. In: 2017 international conference on the frontiers and advances in data science (FADS). Xi'an, China, pp 114–119. https://doi.org/10.1109/FADS.2017.8253208
2. Ammar A (2015) A decision tree classifier for intrusion detection priority tagging. J Comput Commun 3(4):52–58
3. Belgiu M, Drăguţ L (2016) Random forest in remote sensing: a review of applications and future directions. ISPRS J Photogramm Remote Sens 114:24–31
4. Cunningham P, Delany S (2021) K-nearest neighbour classifiers-a tutorial. ACM Comput Surv 54(6):25
5. de Canniere C, Biryukov A, Preneel B (2006) An introduction to block cipher cryptanalysis. Proc IEEE 94(2):346–356. https://doi.org/10.1109/JPROC.2005.862300
6. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Statist 29(5):1189–1232
7. Garuba M, Liu C, Fraites D (2008) Intrusion techniques: comparative study of network intrusion detection systems. In: Fifth international conference on information technology: new generations. IEEE, Las Vegas, NV, USA, pp 592–598. https://doi.org/10.1109/ITNG.2008.231
8. Gascon H, Orfila A, Blasco J (2011) Analysis of update delays in signature-based network intrusion detection systems. Comput Secur 30(8):613–624
9. Gbenga Dada E, Stephen Bassi J, Chiroma H, Muhammad Abdulhamid S, Olusola Adetunmbi A, Emmanuel Ajibuwa O (2019) Machine learning for email spam filtering: review, approaches and open research problems. Heliyon 5(6):e01802
10. Ghandhi M, Srivatsa S (2008) Detecting and preventing attacks using network intrusion detection systems. Int J Comput Sci Secur 2(1):49–60
11. Lee C-H, Lin Y-Y, Lee Y-C, Lee S-J (2017) Machine learning based network intrusion detection. In: 2017 2nd IEEE international conference on computational intelligence and applications (ICCIA). IEEE, Beijing, China, pp 79–83. https://doi.org/10.1109/CIAPP.2017.8167184
12. Li P, Salour M, Su X (2008) A survey of internet worm detection and containment. Commun Surv Tutorials IEEE 10(1):20–35. https://doi.org/10.1109/COMST.2008.4483668
13. Meftah S, Rachidi T, Assem N (2019) Network based intrusion detection using the unsw-nb15 dataset. Int J Comput Digital Syst 8(5):477–487
14. Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military communications and information systems conference (MilCIS). IEEE, Canberra, ACT, Australia, pp 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
15. Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Secur J Glob Perspect 25(1–3):18–31. https://doi.org/10.1080/19393555.2015.1125974
16. Peng Y, Su J, Shi X, Zhao B (2019) Evaluating deep learning based network intrusion detection system in adversarial environment. In: 2019 IEEE 9th international conference on electronics information and emergency communication (ICEIEC). IEEE, Beijing, China, pp 61–66. https://doi.org/10.1109/ICEIEC.2019.8784514
17. Rieger P, Duc Nguyen T, Miettinen M, Reza Sadeghi A (2022) DeepSight: mitigating backdoor attacks in federated learning through deep model inspection. In: Network and distributed systems security (NDSS) symposium 2022. San Diego, CA, USA. Retrieved from https://arxiv.org/pdf/2201.00763.pdf
18. Shah S, Singh P (2012) Signature-based network intrusion detection system using SNORT and WINPCAP. Int J Eng Res Technol 1(12):1–7
19. Sugunan K, Gireesh Kumar T, Dhanya K (2018) Static and dynamic analysis for android malware detection. Adv Big Data Cloud Comput Adv Intell Syst Comput 645:147–155
20. Thanh H, Van Lang T (2020) Evaluating effectiveness of ensemble classifiers when detecting fuzzers attacks on the unsw-nb15 dataset. J Comput Sci Cybern 36(2):173–185

21. Uma M, Padmavathi G (1013) A survey on various cyber attacks and their classification. Int J Netw Secur 15(5):390–396
22. Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S (2019) Deep learning approach for intelligent intrusion detection system. IEEE Access 7:41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334
23. Vinayakumar R, Soman K, Poornachandran P (2017) Applying convolutional neural network for network intrusion detection. In: 2017 International conference on advances in computing, communications and informatics (ICACCI). Udupi, India, pp 1222–1228. https://doi.org/10.1109/ICACCI.2017.8126009
24. Wang W, Jian S, Tan Y, Wu Q, Huang C (2022) Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions. Comput Secur 112:102537. https://doi.org/10.1016/j.cose.2021.102537
25. Yang H, Cheng L, Chuah M (2019) Deep-learning-based network intrusion detection for scada systems. In: 2019 IEEE conference on. Washington, DC, USA, pp 1–7. https://doi.org/10.1109/CNS.2019.8802785
26. Yang Y, Li J, Yang Y (2015) The research of the fast SVM classifier method. In: 2015 12th international computer conference on wavelet active media technology and information processing (ICCWAMTIP). IEEE, Chengdu, China, pp 121–124. https://doi.org/10.1109/ICCWAMTIP.2015.7493959

Attention-Based Neural Architecture Search for Effective Semantic Segmentation of Satellite Images

Avinash Chouhan, Dibyajyoti Chutia, Biswarup Deb, and Shiv Prasad Aggarwal

Abstract Semantic segmentation is an important activity in satellite image analysis. The manual design and development of neural architectures for semantic segmentation is very tedious and can result in computationally heavy architectures with redundant computation. Neural architecture search (NAS) produces automated network architectures for a given task considering computational cost and other parameters. In this work, we proposed an attention-based neural architecture search (ANAS), which uses attention layers at cell levels for effective and efficient architecture design for semantic segmentation. The proposed ANAS has achieved better results than previous NAS-based work on two benchmark datasets.

Keywords Neural architecture search · Semantic segmentation · Attention

1 Introduction

Deep learning has produced good results for image analysis tasks of remotely sensed data such as semantic segmentation [1, 2], change detection [3], and image translation [4]. Semantic segmentation is an important image analysis task that requires dense pixel annotations. Neural architecture design for semantic segmentation is challenging as network components need to be selected in such a way that it can extract dense pixel relations and requires minimum computational cost. Neural architecture search (NAS) is the technique to find optimal architecture from the network search space for a specific task. NAS-based methods are used in some remote sensing image classification [5–9] and multi-label classification [5] tasks. For semantic segmentation of satellite images, several NAS-based approaches [10–15] have been proposed. Some works used NAS for network pruning [14] to generate efficient architecture. Few works used combined approach [10, 15] where NAS-generated architecture

A. Chouhan (✉) · D. Chutia · B. Deb · S. P. Aggarwal
North Eastern Space Applications Centre, Umiam, Meghalaya 793103, India
e-mail: avinash.chouhan@nesac.gov.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_28

is mixed with handcrafted architecture. Channel attention has been used in NAS-based techniques [13, 15, 16]. NAS-based classification and segmentation techniques are also proposed for hyperspectral [15] and synthetic aperture radar (SAR) [7] data. A comprehensive review of NAS-based techniques is presented in [11, 17, 18]. NAS-based methods can generate optimal architectures, but the computation cost requirement is significant and grows exponentially with an increase in search space and depth of search. In this work, we proposed attention-based NAS (ANAS), which consists of attentive search space for efficient architecture search. ANAS could produce better results than other NAS-based methods with fewer FLOPS requirements.

2 Literature Survey

Zhang et al. [19] proposed neural architecture search with directed acyclic graph and Gumble Max trick for satellite image segmentation. Hurt et al. [20] created a meta dataset using four benchmark datasets and presented the NAS-based comparative analysis. Peng et al. [21] proposed a gradient optimization-based greedy and progressive automatic architecture search framework with incremental dataset and number of epochs. They also propose a combined dataset strategy. Salinas et al. [22] utilized the NAS system for the classification of remotely sensed data, and the NAS system is further extended to take advantage of transfer learning using the pretrained models. Wei et al. [23] proposed an efficient architecture search that can reduce the number of channels in a pruned model variant using differential NAS with a computational cost-sensitive loss function. Zheng et al. [24] proposed a combined approach of manual and automatic architecture search where an encoder network is created using gradient-based NAS with two-level search space. The adaptive decoder network is manually selected based on the searched encoder network. Zhang et al. [25] proposed an automated architecture search that can effectively model spatiotemporal relations using inter- and intra-module level search space. Broni-Bediako et al. [5] utilized symbolic linear generative encoding-based NAS to produce a multipath network with skip connections for single and multi-label classification of remotely sensed images with early stopping. Xie et al. [10] proposed a combined approach with an automatically searched encoder and attention decoder network with multiscale fusion and matching attention for real-time semantic segmentation. Elksen [11] presented the detailed review on NAS-based works for semantic segmentation and object detection tasks. Gudzius et al. [12] proposed an automatic architecture search approach to produce a multiscale and skip connection-based network for satellite semantic segmentation in a limited annotation scenario. Lu et al. [26] proposed energy-based filter pruning that used maximum and average parameter distribution to generate pruned networks, which are performance-wise comparable and require lesser computational cost than NAS methods. Guo et al. [13] proposed a dynamic high-resolution network that is produced using NAS with accelerated prox-

imal gradient and channel attention. Xiao et al. [6] proposed differential NAS to search automatic plug-and-play module for multiscale architecture using new search space. Ang et al. [14] proposed NAS-based method to search effective decoders for predefined encoder modules and architecture for hyperseptral image segmentation. A parameter-free pruning strategy using perturbation is also proposed. Wang et al. [15] proposed a multiscale attentive network where NAS is used to produce a lightweight feature extraction module. Radosavovic et al. [27] proposed designing of network design space with parametrization of possible network instances to produce an efficient model. Dong et al. [7] used differential NAS-based model generation for PolSAR image classification, and NAS extension was developed to handle complex values. Zhang et al. [8] used differential NAS with regularization to produce efficient architectures and to overcome the mode collapse problem. Li et al. [9] used NAS with recurrent neural network controller to produce network architecture with newly designed search space. Wang et al. [28] proposed gradient-based NAS, which utilized two stages with hierarchical search to generate task-specific architectures. Wei et al. [29] proposed automatic mixed precision network generation using bit quantization information for each layer produced using NAS. Liu et al. [30] proposed a combined cell and network-level architecture search for semantic segmentation tasks. It reduced the search time significantly compared to previous methods. Wang et al. [31] proposed decoupled supernet-based three-level architecture search for memory efficient architecture search. Zhang et al. [32] proposed dense search space with proxyless architecture search with path and feature level sampling for memory efficient search. Nakai et al. [16] used attention blocks after each operation of the searched architecture to avoid redundant information. NAS-based semantic segmentation methods face the following challenges.

1. Full search space searching is computationally very expensive.
2. The selection of cell search space is essential and can affect the overall results significantly.

3 Proposed Scheme

The following are the contributions of this work:

1. We proposed the use of attention construct in the base cell level search space in multi-level supernet-based NAS.
2. ANAS achieved state-of-the-art results with lesser computational requirements.
3. Extensive qualitative and quantitative results are shown to highlight the superior performance of ANAS.

Figure 1 shows the workflow of ANAS. ANAS followed three levels (Path level, Connection level, and Cell level) architecture search as used in DNAS [31]. ANAS used four-resolution search space in path level search. The path network that is frozen

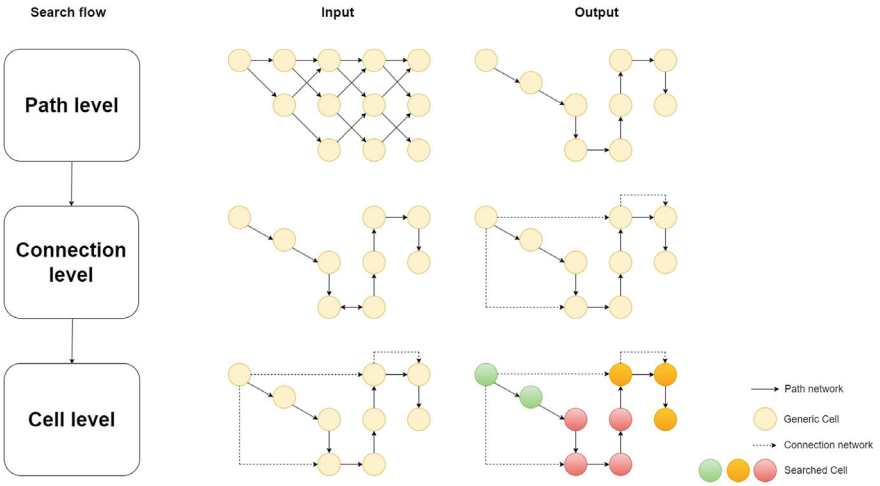


Fig. 1 Workflow of attention-based neural architecture search. Here proposed attentive search space is used in cell level search

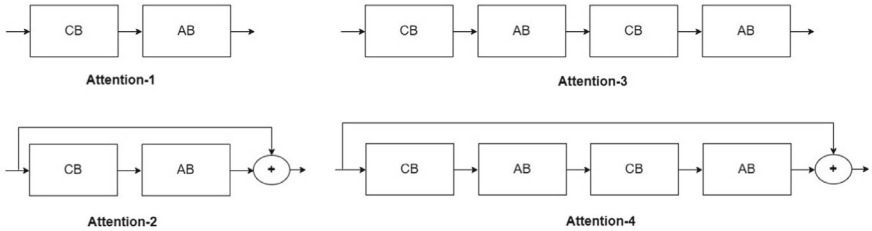


Fig. 2 Block diagram of attentive constructs used in cell level search. Here, CB is the convolutional block and AB is the attention block

in path level search is further refined in connection level search where the interconnection between frozen paths is searched. In cell level search, ANAS used attentive search space, which consists of four attention constructs (Attention-1, Attention-2, Attention-3, Attention-4) for better cell level design. Attention at the cell level enables focusing on important spatial and channel-level features, which results in better segmentation output. The details of attentive constructs are shown in Fig. 2. Convolutional block consists of convolution with ReLU layer, and attention block contains spatial and channel attention based on CBAM [33]. Attention-1 and Attention-3 are used when input and output have different numbers of features. Attention-2 and Attention-4 have residual connections and are used when input and output have the same number of features. Attention-1 and Attention-2 are simple attentive constructs that are further extended in Attention-3 and Attention-4 with stacking of layers. The details of all operations in cell level search space are given in Table 1.

Table 1 Search space for cell level search in proposed ANAS

Operator	Size	Operation
Convolution	3 × 3	Simple convolution
DilConv	3 × 3 d	Convolution with dilation rate d
SepConv	3 × 3 5 × 5,7 × 7	Depth seprable convolution
Pooling	3 × 3	Max, avg
Global pooling		
Attention-1	3 × 3	Attention Block with the different number of input and output features
Attention-2	3 × 3	Attention Block with same numbers of input and output features
Attention-3	3 × 3	Attention Block with the different number of input and output features
Attention-4	3 × 3	Attention Block with same numbers of input and output features

4 Experiments and Results

4.1 Dataset Used

In this study, two benchmark datasets (GID [34], ISPRS Vaihingen) are used to evaluate the proposed scheme. The color code used to represent label information for both datasets is shown in Fig. 3. The details of these datasets are as follows:

4.1.1 GID

GID dataset contains 31500 satellite images with the size of 512 × 512 pixels. It has 18900, 6300, and 6300 images in train, validation, and test sets. The ground truth images have semantic information of 5 land cover classes (Built-up, Farmland, Forest, Meadow, Water).

4.1.2 ISPRS Vaihingen

This dataset consists of 33 aerial scenes, which are divided into training and testing sets of 17 and 16 images. The images are of size around 3000 × 3000 pixels and are cropped to make patches of size 512 × 512. The label image contains semantic information for six land cover classes (Clutter, Tree, Low vegetation, Car, Building, Impervious surface).

Fig. 3 Color code details used in GID and VEH (ISPRS Vaihingen) datasets

Color	GID	VEH
	Built-up	Clutter
	Farm land	Tree
	Forest	Low vegetation
	Meadow	Car
	Water	Building
		Impervious surface

4.2 Experimentation Setup

All experiments are done on a system with 80 GB of graphics memory (Nvidia A100 card) and 128 GB of system memory. PyTorch framework is used to implement the experiments.

4.3 Training and Evaluation

The hyperparameters details which are used in model search and retrain phases are presented in Table 4. We followed optimization steps as used in Auto-Deeplab [30] and DNAS [31]. Train dataset is divided into two parts (train_1, train_2) and network weight parameter n_w are updated using train_1 data and network architecture parameters n_{a_i} are updated using train_2 data after calculating cross entropy losses $Lw_{train_1}(n_w, n_{a_1}, n_{a_2}, n_{a_3})$ and $La_{train_2}(n_w, n_{a_1}, n_{a_2}, n_{a_3})$. The model is searched in three stages. In each stage, the model is trained using train_1 data for alpha epochs and train_1 and train_2 datasets for remaining epochs. Finally, the search architecture is retained with the complete training dataset.

Numerical comparisons are made using mean intersection over union (mI), frequency weighted intersection over union (fI), and overall accuracy (OA), which are calculated using the following equation using values of true positive (tp), true negative (tn), false positive (fp), and false negative (fn).

$$\begin{aligned} \text{mI} &= \text{mean}\ (\frac{\text{tp}}{\text{tp}+\text{fp}+\text{fn}}) \\ \text{fI} &= \text{mean}\ (\frac{\text{tp}.(\text{tp}+\text{fn})}{(\text{tp}+\text{fp}+\text{fn})(\text{tp}+\text{fp}+\text{tn}+\text{fn})}) \\ \text{OA} &= \frac{\text{tp}+\text{tn}}{\text{tp}+\text{fp}+\text{tn}+\text{fn}} \end{aligned} \tag{1}$$

Table 2 Numerical results on GID dataset

Method used	mI	OA	fI	Param	Mem	gFLOPS
UNet [1]	0.87	0.92	0.91	48.44	1482	131.72
PSPNet [2]	0.90	0.97	0.95	72.58	2820.78	279.64
AutoDeeplab [30]	0.87	0.97	0.95	13.35	925.15	16.19
DNAS [31]	0.91	0.98	0.96	7.14	1195.38	54.06
ANAS	0.92	0.98	0.96	8.9	1160	27.59

Here, Params is the number of parameters in million, Mem is memory in MB, and gFLOPS is Giga floating-point operations

Table 3 Numerical results on ISPRS Vaihingen dataset

Method used	mI	OA	fI	Param	Mem	gFLOPS
UNet [1]	0.51	0.77	0.64	48.44	1482	131.72
PSPNet [2]	0.51	0.79	0.67	72.58	2820.78	279.64
AutoDeeplab [30]	0.52	0.81	0.68	4.5	488.46	28.16
DNAS [31]	0.50	0.80	0.67	2.1	430.86	24.7
ANAS	0.55	0.85	0.71	3.5	427.63	17.16

Here, Params is the number of parameters in a million, Mem is memory in MB, and gFLOPS is Giga floating point operations

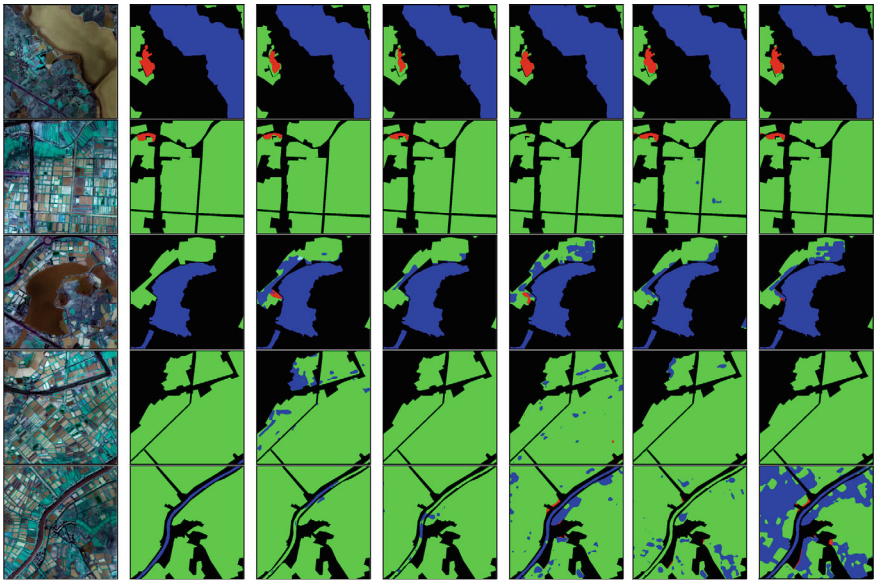
5 Analysis

The performance comparative evaluation is done using numerical and visual analysis. For both datasets, results are generated using UNet [1] and PSPNet [2]. The comparative NAS results are produced for AutoDeeplab [30], DNAS [31], and ANAS.

5.1 Numerical Analysis

In Table 2, numerical results are shown for the GID dataset. UNet achieves the baseline result with values of .87, .92, .91 for mI,OA, and fI. DNAS got .91, .98, .96 values for mI, OA, and fI which is also previous best results. ANAS achieved mI, OA, and fI results of .92, .98, and .96 and required 27.59 GFLOPS. ANAS got better mI and similar OA and fI with half GFLOPS requirements.

Table 3 presented numerical results for the ISPRS Vaihingen dataset. UNet achieves the baseline result with values of .51, .77, .64 for mI, OA, and fI. DNAS got .50, .80, .67 values for mI, OA, and fI. ANAS achieved mI, OA, and fI results of .55, .85, and .71 and required 17.16 GFLOPS. ANAS got better mI OA, fI, and lesser GFLOPS compared to DNAS [31].



(a) Inp (b) Lbl (c) PSPNet (d) ADL (e) DNAS (f) ANAS (g) UNet

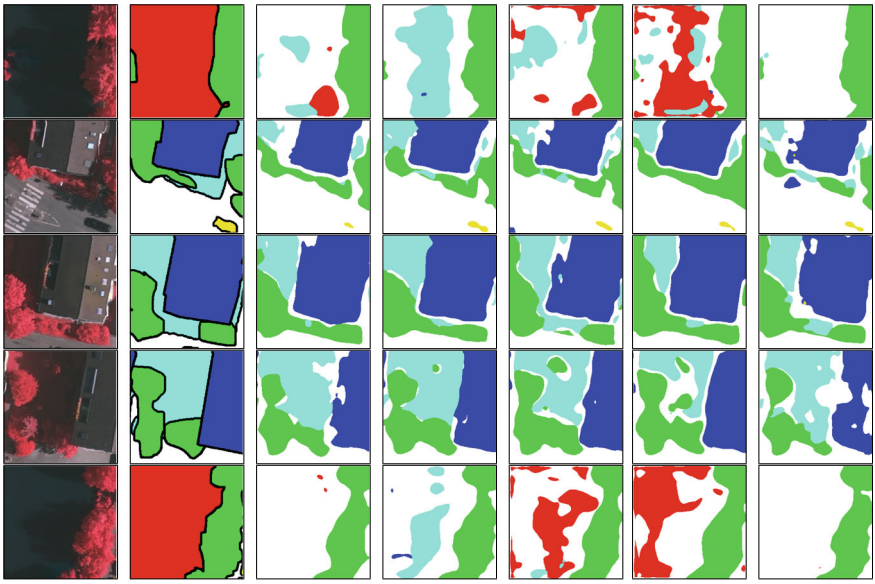
Fig. 4 Output comparison results for Unet, PSPNet, AutoDeeplab, DNAS, ANAS for GID dataset. Here, Inp is the input image, Lbl is the label image, and ADL is AutoDeeplab

5.2 Visual Analysis

Figure 4 shows visual results for the GID dataset. DNAS produced the closest result to ground truth compared to other methods. Figure 5 presented visual results for the ISPRS Vaihingen dataset. ANAS is able to generate good results for clutter class, which other methods missed completely (Table 4).

6 Conclusion

We proposed the attention base neural architecture search approach for efficient architecture design for semantic segmentation of satellite images. ANAS contains attention-based cell level search space, which enables efficient network creation. ANAS is evaluated on two benchmark datasets and shows the qualitative and computational advantage over the previous NAS-based approach. The results are also validated using visual comparisons with results from other methods.



(a) Inp (b) Lbl (c) PSPNet (d) ADL (e) Unet (f) ANAS (g) DNAS

Fig. 5 Output comparison results for Unet, PSPNet, AutoDeeplab, DNAS, ANAS for VEH dataset. Here, Inp is the input image, Lbl is the label image, and ADL is AutoDeeplab

Table 4 Details of hyperparameters used in the experimentation

Stage	Batch	Epoch	Alpha
Search_1	12	60	20
search_2	12	60	20
Search_3	12	60	20
Retrain	24	300	–

Here, search_i is ith search stage, and retrain is the retraining stage after the supernet search completion

References

1. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241, Cham. Springer International Publishing
2. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In CVPR
3. Chouhan A, Sur A, Chutia D (2022) Drmnet: difference image reconstruction enhanced multiresolution network for optical change detection. IEEE J Sel Top Appl Earth Obs Remote Sens 15:4014–4026
4. Chouhan A, Jindal N, Sur A, Chutia D, Aggarwal SP (2023) Edcgan: encoder decoder based conditional gan for sar to optical image translation. In: Proceedings of the thirteenth indian conference on computer vision, graphics and image processing, ICVGIP '22, New York, NY,

USA. Association for Computing Machinery
5. Broni-Bediako C, Murata Y, Mormille LH, Atsumi M (2022) Searching for cnn architectures for remote sensing scene classification. IEEE Trans Geosci Remote Sens 60:1–13
6. Xiao A, Shen B, Tian J, Hu Z (2023) Pp-nas: searching for plug-and-play blocks on convolutional neural networks. IEEE Trans Neural Netw Learn Syst
7. Dong H, Zou B, Zhang L, Zhang S (2020) Automatic design of cnns via differentiable neural architecture search for polsar image classification. IEEE Trans Geosci Remote Sens 58(9):6362–6375
8. Zhang Z, Liu S, Zhang Y, Chen W (2022) Rs-darts: a convolutional neural architecture search for remote sensing image scene classification. Remote Sens 14(1)
9. Li L, Tian T, Li H (2019) Classification of remote sensing scenes based on neural architecture search network. In: 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), pp 176–180
10. Xie B, Yang Z, Yang L, Luo R, Wei A, Weng X, Li B (2022) Multi-scale fusion with matching attention model: a novel decoding network cooperated with nas for real-time semantic segmentation. IEEE Trans Intell Transp Syst 23(8):12622–12632
11. Mohan R, Elsken T, Zela A, Metzen JH, Staffler B, Brox T, Valada A, Hutter F (2022) Neural architecture search for dense prediction tasks in computer vision. arXiv preprint arXiv:2202.07242
12. udzius P, Kurasova O, Darulis V, Filatovas E (2023) Automl-based neural architecture search for object recognition in satellite imagery. Remote Sens 15(1)
13. Guo S, Yang Q, Xiang S, Wang P, Wang X (2023) Dynamic high-resolution network for semantic segmentation in remote-sensing images. Remote Sens 15(9):2293
14. Ang SP, Phung SL, Bui L, Bouzerdoum A (2023) Adaptornas: a new perturbation-based neural architecture search for hyperspectral image segmentation. IEEE Trans Circ Syst Video Technol
15. Wang J, Huang R, Guo S, Li L, Zhu M, Yang S, Jiao L (2021) Nas-guided lightweight multiscale attention fusion network for hyperspectral image classification. IEEE Trans Geosci Remote Sens 59(10):8754–8767
16. Nakai K, Matsubara T, Uehara K (2020) Att-darts: differentiable neural architecture search for attention. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp 1–8
17. Borba P, de Souza Bias E, da Silva NC, Roig HL (2021) A review of remote sensing applications on very high-resolution imagery using deep learning-based semantic segmentation techniques. Int J Adv Eng Res Sci 8(8)
18. Osco LP, Junior JM, Ramos AP, de Castro Jorge LA, Fatholahi SN, de Andrade SJ, Matsubara ET, Pistori H, Gonçalves WN, Li J (2021) A review on deep learning in uav remote sensing. Int J Appl Earth Obs Geoinf 102:102456
19. Zhang M, Jing W, Lin J, Fang N, Wei W, Woźniak M, Damaševičius R (2020) Nas-hris: automatic design and architecture search of neural network for semantic segmentation in remote sensing images. Sensors 20(18):5292
20. Hurt JA, Huangal D, Davis CH, Scott GJ (2020) Enabling machine-assisted visual analytics for high-resolution remote sensing imagery with enhanced benchmark meta-dataset training of nas neural networks. In: 2020 IEEE International Conference on Big Data (Big Data), pp 3914–3923
21. Peng C, Li Y, Jiao L, Shang R (2020) Efficient convolutional neural architecture search for remote sensing image scene classification. IEEE Trans Geosci Remote Sens 59(7):6092–6105
22. Palacios Salinas NR, Baratchi M, van Rijn JN, Vollrath A (2021) Automated machine learning for satellite data: integrating remote sensing pre-trained models into automl systems. In: Joint european conference on machine learning and knowledge discovery in databases, pp 447–462. Springer
23. Wei X, Zhang N, Liu W, Chen H (2022) Nas-based cnn channel pruning for remote sensing scene classification. IEEE Geosci Remote Sens Lett 19:1–5
24. Zheng C, Wang J, Ma A, Zhong Y (2022) Autolc: Search lightweight and top-performing architecture for remote sensing image land-cover classification. In: 2022 26th International conference on pattern recognition (ICPR) pp 324–330

25. Zhang X, Jin Q, Xiang S, Pan C (2022) Mfnet: the spatio-temporal network for meteorological forecasting with architecture search. IEEE Geosci Remote Sens Lett 19:1–5
26. Yiheng L, Gong M, Zhuping H, Zhao W, Guan Z, Zhang M (2023) Energy-based cnn pruning for remote sensing scene classification. IEEE Trans Geosci Remote Sens 61:1–14
27. Radosavovic I, Kosaraju R, Girshick R, He K, Dollar P (2020) Designing network design spaces. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pages 10425–10433, Los Alamitos, CA, USA. IEEE Computer Society
28. Wang J, Zhong Y, Zheng Z, Ma A, Zhang L (2021) Rsnet: the search for remote sensing deep neural networks in recognition tasks. IEEE Trans Geosci Remote Sens 59(3):2520–2534
29. Wei X, Chen H, Liu W, Xie Y (2021) Mixed-precision quantization for cnn-based remote sensing scene classification. IEEE Geosci Remote Sens Lett 18(10):1721–1725
30. Liu C, Chen L, Schroff F, Adam H, Hua W, Yuille AL, Fei-Fei L (2019) Auto-deeplab: hierarchical neural architecture search for semantic image segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 82–92, Los Alamitos, CA, USA. IEEE Computer Society
31. Wang Y, Li Y, Chen W, Li Y, Dang B (2022) Dnas: Decoupling neural architecture search for high-resolution remote sensing image semantic segmentation. Remote Sens 14(16)
32. Zhang X, Xu H, Mo H, Tan J, Yang C, Wang L, Ren W (2021) Dcnas: densely connected neural architecture search for semantic image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 13956–13967
33. Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: Convolutional block attention module. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision – ECCV 2018, pages 3–19, Cham. Springer International Publishing
34. Tong X-Y, Xia G-S, Qikai L, Shen H, Li S, You S, Zhang L (2020) Land-cover classification with high-resolution remote sensing images using transferable deep models. Remote Sens Environ 237:111322

A Location-Based Application for Recording the Video Stream of Urban Traffic Cameras

Karl Aleksander Leoste, Johan Valdemar Leoste, Janika Leoste, and Jose San Martin

Abstract Dashboard cameras, or dash cams, have become essential in enhancing road safety, providing accident evidence, deterring fraud, and promoting responsible driving. This paper explores the compatibility of Tallinn's traffic cameras for traffic incident documentation. Over 50% of city center intersections are covered, and reliability issues are minimal. The developed mobile application augments traditional dash cams using these cameras, effectively tracking vehicles and documenting incidents. However, it has limitations in capturing pedestrians and cyclists. This integration represents a promising step toward comprehensive traffic incident documentation, combining onboard and intersection camera strengths. Future work should focus on user interface, real-time traffic data integration, and advanced detection methods.

Keywords Urban traffic cameras · Dashboard cameras · Traffic safety

1 Introduction

In the realm of automotive technology, dashboard cameras, or dash cams, have emerged as indispensable tools. Their necessity stems from their multifaceted role in enhancing road safety, providing irrefutable evidence in accidents, preventing fraudulent claims, promoting responsible driving behavior, and safeguarding vehicles both in motion and at rest. As defined by Merriam-Webster [1], a dashboard camera or a dash cam is a "*video camera mounted usually on the dashboard of a vehicle and used to continuously record activity through the vehicle's windshield.*"

K. A. Leoste · J. Leoste (✉)
Tallinn University, Tallinn, Estonia
e-mail: leoste@tlu.ee

J. V. Leoste · J. Leoste
Tallinn University of Technology, Tallinn, Estonia

J. S. Martin
Universidad Rey Juan Carlos, Madrid, Spain

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_29

Dash cams are designed to record continuously video footage, primarily capturing the events transpiring in close proximity to the vehicle. Dash cams come in two primary categories: single-channel (1-channel) and dual-channel (2-channel) models. Single-channel dash cams, which capture a forward-facing view of the road, constitute the majority of the market. In contrast, dual-channel dash cams encompass two distinct configurations: one focused on recording the front and interior of the vehicle, and the other capturing both the front and rear perspectives. As of 2021, single-channel dash cams accounted for over 72.5% of the total dash cam sales revenue globally, reflecting their popularity and affordability compared to multichannel alternatives [2]. As described by Kim et al. [3], dash cam feeds can be enhanced with blockchain technology to ensure video privacy and prevent tampering. Together with the automated accident-initiated cloud-upload of the last minute footage, such video feed can be used to verify the accident details. However, while single-channel dash cams offer a clear view of the road ahead, they lack coverage of events occurring behind the vehicle. Even dual-channel dash cams, despite their added perspective, do not provide comprehensive coverage of the vehicle's surroundings. As an alternative or supplementary approach to dash cams, urban traffic surveillance cameras can be used.

In modern urban environments, low-cost video surveillance cameras are used for automated traffic surveillance. These cameras generate enormous quantities of data that can be submitted to Big Data processing methods [4]. However, currently this potential is usually not realized, as the data from the cameras are provided, processed, and accessed, based on individual cameras [5]. In our paper, we describe a solution that uses the data feed from the traffic surveillance cameras in Tallinn, Estonia, in order to provide more relevant details in case of traffic incidents. Tallinn Transport Department maintains a webpage displaying real-time traffic video feeds from various intersections throughout Tallinn [6]. To protect individual privacy, the displayed video feeds are intentionally of low resolution and frame rate, rendering individuals unidentifiable by direct or indirect means. The primary objective of this webpage is to provide up-to-date information on current traffic and road conditions, adhering to data protection regulations [7]. As of April 2023, the public can access feeds from 211 cameras, predominantly at intersections, but also in two parking areas. While these feeds offer real-time insights, individually they lack the functionality required for replacing a dash cam, as they do not support video recording or real-time tracking of vehicle location.

Our goal was to develop a functional mobile and server application that utilizes GPS technology to track its user's car through urban environments. The purpose of this application would be providing relevant additional details in case of traffic incidents. This application's core functionality revolves around capturing video frames from traffic cameras that include the user's car within the frame. Essentially, our aim was to provide users with a tool similar to an on-board camera, facilitating the ability to document and ascertain the details surrounding a traffic incident. While the development process of the application is described in more detail in the work by Author and Author [8], this paper presents a shorter overview of the challenges, processes, and results of developing the application.

2 Methods

2.1 Compatibility Analysis of Tallinn Traffic Cameras

The first part of the analysis aims to examine the overlap of Tallinn traffic cameras with locations where the most accidents occur. Next, the reliability of the cameras is evaluated, and finally, the image quality and frame rate of the cameras are discussed to assess their impact on their utility.

2.1.1 Coverage Analysis

In Tallinn's publicly available web application provided by the Transport Department, the camera locations are typically identified by the names of the two intersecting streets and a camera identification number. However, precise GPS coordinates for each camera are missing, making it necessary to add each camera's location to a map for later export in machine-readable format. We selected Google My Maps as the mapping application as it allows for rapid data entry and provides sufficient options to record camera ID, location, direction, and even attach images to make identification easier. Data can be exported in.csv format for easy integration into their application. To determine the location of each camera, we conducted the following steps.

First, Google Street View was used to examine the target intersections, identifying easily recognizable buildings visible in Street View and matching them to buildings potentially captured by the camera. This process helped establish both the road on which the camera was located and its approximate direction. Then we compared the cars visible in the camera view with those in the Street View to determine precisely the camera's direction. This process was applied to all cameras, and in cases where visibility might have been temporarily reduced due to snow in winter, we relied on images from other cameras at the same intersection to make accurate determinations. Although the current application does not utilize camera directions, these can be potentially useful for future development or other purposes, such as assessing whether the sun shines on the camera lens at specific times of the year and day based on camera direction and nearby building locations.

The coverage analysis was performed using two datasets. The first part involved quantitative picture analysis using the Estonian Insurance Association (EIA) and the Traffic Insurance Fund (TIF) joint dataset of Estonian insurance cases, focusing on Tallinn. The goal was to demonstrate that a significant portion of insurance cases in the city center occurs at intersections rather than on straight road segments. Due to the unavailability of downloadable raw data, we limited the analysis to a specific central parking zone in Tallinn. The second part of the analysis involved the use of the Transport Department's dataset of traffic accidents involving human casualties, covering a 12-year period from 2011 to 2022. This dataset was filtered to include only Tallinn accidents with known GPS coordinates and excluded accidents outside of the city. This dataset was then compared with the camera locations dataset, calculating the

distance between each accident and the nearest camera to determine if it potentially fell within the camera's field of view.

The coverage analysis results showed that the majority of accidents occurred at intersections, confirming the utility of intersection cameras. We also discovered a few areas in Tallinn, where camera coverage was insufficient, despite a higher number of accidents. We suggest that these areas would benefit from additional cameras.

2.1.2 Camera Reliability Assessment

The assessment of camera reliability was conducted in three distinct phases. In the first phase, spanning three months, images from all the cameras located at Tallinn intersections were recorded at least once a week. This allowed for calculating the average number of cameras that were unused each month. In the second phase, images from all the cameras were continuously recorded for 48 h at one-hour intervals to investigate if there were specific times when a camera was not in use. The final phase involved recording 22 camera streams for over two hours continuously at their maximum frame rate. This aimed to assess their reliability at their highest frame rate, including identifying issues like video feed freezing or the occurrence of repeated frames. For additional analysis, we developed a program that established an HTTPS connection with the webpage and retrieved images from the selected cameras in order to identify problems such as missing frames and image corruption.

There are other common reasons for unreliability of traffic camera feed:

- Detecting *nonfunctioning cameras* was facilitated by Tallinn Transport Authority's practice of displaying a static image instead of a live camera feed when a camera was not working. This static image was consistent and could be easily automated for detection.
- Identifying *blurry images* was a more subjective task, as it required determining the point at which an image became too blurry to be useful. This threshold was set based on factors like the distortion of vehicle shapes or a significant drop in image contrast. Blurriness in images could not only be caused by various factors, including weather conditions (e.g., rain or snow) but also issues like dusty or wet camera lenses exposed to direct sunlight.

The camera reliability assessment showed variations in camera performance influenced by factors like weather conditions and sunlight exposure. Some cameras experienced partial sunlight interference during the sunset times. However, these findings were limited to the observation period and may vary seasonally and by location.

Out of 193 tested cameras, 35 exhibited issues during the test, excluding cameras temporarily out of service due to construction. These issues ranged from data corruption to cameras being temporarily blinded by wet road surfaces. The latter was particularly prevalent at night, during periods of rain, and was influenced by the changing trajectory of the sun throughout the year. At least one camera had a consistently slightly blurry image due to a dusty lens but also experienced unpredictable image

corruption at irregular intervals. Both of these problems might have been caused by moisture inside the camera.

2.1.3 Camera Image Quality

While the cameras themselves are able to provide higher quality videos, the video feeds available on the Tallinn intersection camera website have low resolution. Most cameras have a frame size of 1280 × 720 pixels and a frame rate of 1 frame per second (fps). This lower resolution is intentional to comply with the requirements of the Data Protection Inspectorate. In addition, the images are heavily compressed using the JPEG format, which divides the image into 8 × 8 pixel blocks and reduces color and detail information within each block aggressively. In comparison, many dashboard cameras sold in 2021 had resolutions of 1280 × 720p or lower, leading to difficulties in identifying license plates. The frame rate of 1 fps is adequate for urban traffic where speeds are typically around 50 km/h or lower, as it provides sufficient information about vehicle trajectories and speeds—although known reference points in the image such as pedestrian crossings are required.

Despite its limitations in resolution and frame rate, the camera image quality is sufficient for identifying vehicles' colors and shapes. The low frame rate does not hinder the calculation of vehicle speeds or the prediction of trajectories. However, further testing might be needed, especially in rainy conditions.

2.1.4 Results of the Compatibility Analysis

The outcome of the compatibility analysis was generally positive. In the city center, over 50% of intersections were already covered by cameras, and more cameras are continuously being added. There were no significant issues with reliability, with the most common disruptions being related to weather conditions and construction-related camera outages. The image quality, while not sufficient for identifying license plates or individuals, met the goal of enhancing the precision of accident circumstances. Despite the cameras being located only at intersections, accidents tend to occur most frequently in their vicinity. During the reliability testing occasional variations in image quality were observed, potentially requiring more investigation. Nevertheless, for most of the time, the visibility of cameras at intersections was adequate.

2.2 Software Development

2.2.1 General Choices

In the development decisions for the project, there were various options to consider, especially in terms of technology and project structure. The choices made had to accommodate both the planned work during development and potential future functionality.

For server-side applications, two key components were the API application and the rendering application. The API application's role was to communicate with the client application, while the rendering application was responsible for combining camera images into videos.

To select the appropriate API framework, several criteria were considered. These included the speed of code development, the framework's performance, and ease of working with data models. ASP.NET Core was chosen because it excelled in these aspects, making coding efficient and promoting component reuse. Other frameworks, such as Django, Spring Boot, Express, and Laravel, were considered but excluded based on various criteria.

For rendering the application, the decision was made to use, for convenience, the same language as the API application, which was C# in this case.

The client application's role was to enable users to interact with the project, such as initiating or ending tracking, updating their location, and downloading videos. To decide between a web and mobile application, criteria like background GPS tracking and ease of development were assessed. Despite the ease of web application development, the requirement for background GPS tracking made it necessary to choose a mobile application.

Three mobile application frameworks were considered: React Native, Flutter, and .NET Maui. After evaluating their capabilities and compatibility with the server-side applications, .NET Maui was chosen due to its logical alignment with the project's needs and ease of setup.

Security requirements for the project included authorization for requests, user-specific data management, and seamless data access even in cases of session termination or device change. To fulfill these requirements, an authentication and authorization system was needed. The choice was between creating a custom system and using an external service.

Using an external service, such as Auth0, was selected because it simplified user authentication and authorization while reducing the need to manage user passwords. Auth0 allowed users to log in with existing social media accounts, aligning with the project's goal of providing a seamless experience.

2.2.2 Architecture

The comprehensive tracking service consists of six major components: client application, API application, rendering application, database, traffic camera service, and identity service.

The client application is used for tracking services. Users can initiate and terminate tracking through this application, which periodically updates GPS locations. The client application communicates with an API application and requires authorization for its requests. To achieve this authorization, the client application interacts with an identity service for user login. The client application's structure includes two views: a login view with a login button redirecting users to Auth0 login services, and a logged-in user view with three buttons for actions: log out, start tracking, and stop tracking. The start-tracking button triggers API requests to initiate tracking and activate a timer for periodic location updates, while the stop-tracking button halts the timer and sends a corresponding request. Testing was conducted on an Android smartphone with an active data plan.

The API application was used as an intermediary between the database and the client application. It exposes specific endpoints for communication with the client application and ensures that incoming requests are authorized by checking the user's login status through an identity service. Authorized requests lead to database updates, including adding or removing users from the list of tracked users and updating location information. The API application's structure involves three endpoints: start tracking, stop tracking, and location update, each requiring authorized requests from logged-in users.

The render application plays a crucial role in creating videos from database information. This application operates by fetching data from the database every second, including user locations and camera details. It then matches users with nearby cameras, retrieves real-time images from these cameras, and combines them into video frames. Each user's video is divided into 30 s segments. To ensure efficient processing, the application employs .NET's built-in task framework and channels. This approach allows tasks to be managed without causing delays. Furthermore, the Render Application runs on the author's personal computer, taking advantage of its powerful graphics card to expedite video rendering. The videos are stored directly on the local machine, simplifying development and debugging while eliminating the need for separate server downloads.

The database serves as a link between the API and rendering applications, storing data on currently tracked individuals and their locations. It was deployed on Azure servers using free student credits, following Microsoft's documentation. Initially, it used Azure Active Directory for authorization but later switched to a username-password combination for flexibility. A firewall was configured to allow network traffic from the home network and access for Azure services, which were initially restricted. However, it is important to note that the selected Azure database service can be relatively expensive, with a monthly cost of €353.73 according to the Azure calculator. This cost may be a concern for long-term implementations, but it was suitable for testing and experimentation using free credits.

The Traffic Camera Service is responsible for retrieving real-time images from traffic cameras, specifically the Tallinn traffic camera website. In this project, images are requested individually from these cameras. Since there is no official API for obtaining camera images from the Tallinn website, the images must be directly extracted from the web page. Due to already significant traffic on the Tallinn traffic camera website, the application must optimize its interactions with the website and minimize the additional load. The application should only store images from cameras near its users, and it should start capturing images slightly before users enter the camera's field of view to ensure users are included in the shots. This is achieved by comparing the GPS locations of cameras and users, considering an adjustable radius. As requested by the Tallinn Transport Department's website, the application's code is kept confidential. For access to the code, interested parties are advised to contact the authors directly.

The Identity Service in this project serves a crucial role in managing user authentication and authorization. It facilitates user logins for the client application and authorizes requests in the API, ensuring that only authenticated users can access its functionalities. Auth0, a third-party service, is employed to implement the Identity Service, simplifying the authentication process significantly. Configuring Auth0 for this project involved following comprehensive documentation available on Auth0's website, which includes detailed guides for various use cases and technologies [9]. Auth0's quick-start guides provide step-by-step instructions and even offer sample source code for building applications from scratch. However, some challenges emerged during the configuration process, particularly when developing a .NET Maui mobile application on Windows. Auth0 had an issue with post-login redirection on Windows, which was addressed by creating an in-app page for redirection. Additionally, adapting Auth0's documentation to meet project-specific requirements, like restricting API access to authenticated users, necessitated custom code modifications. Auth0 offers both free and paid versions, accommodating various user needs, with the free version suitable for up to 7000 active users, while paid options provide more advanced features and customization capabilities, primarily targeted at larger enterprises.

The application processes two main types of user data: precise location and images from traffic cameras. Users have full control over location tracking. The app respects user privacy by ensuring that tracking is initiated by the user and by locating cameras only in high-traffic areas, avoiding residential and private spaces. Regarding camera images, the app collects them only when users are present in the frame. These images are integrated into users' videos and are not stored separately or accessible to users or third parties. The app complies with data protection guidelines by minimizing data collection and using these images for personal purposes, exempting it from certain data protection regulations (see also [10]). In cases requiring sharing with third parties, like traffic accident investigations, the app aligns with its primary function, but it should consider limiting the duration of saved recordings to enhance data protection.

3 Results

The testing of the developed application involved a short test drive through Tallinn to evaluate how well the application tracked the vehicle. The route covered various areas of the city, where the presence of cameras on specific intersections was considered. The test lasted a total of 34 min, during which the car passed through 16 different intersections with cameras. The indicated black car in Fig. 1 was chosen for the test to represent a typical everyday vehicle, and it was observed that even among similar cars, the distinct design features of this model made it easily identifiable in the pictures (Fig. 1).

The results indicated that the application could effectively track the vehicle, even when multiple similar cars were present. The quality and resolution of the camera proved sufficient for this purpose. The application successfully tracked the test vehicle for most of the route, with only brief periods when it was out of the camera's view. On average, the vehicle was outside the camera's view for only 1–2 min, with the longest period without camera coverage being around 10 min in areas with fewer cameras. The analysis included 36 different cameras on 16 distinct intersections, with an average of 2 cameras per intersection and a few having 3–4 cameras. A couple of intersections with cameras were excluded from the analysis as the video feeds were temporarily down during testing.

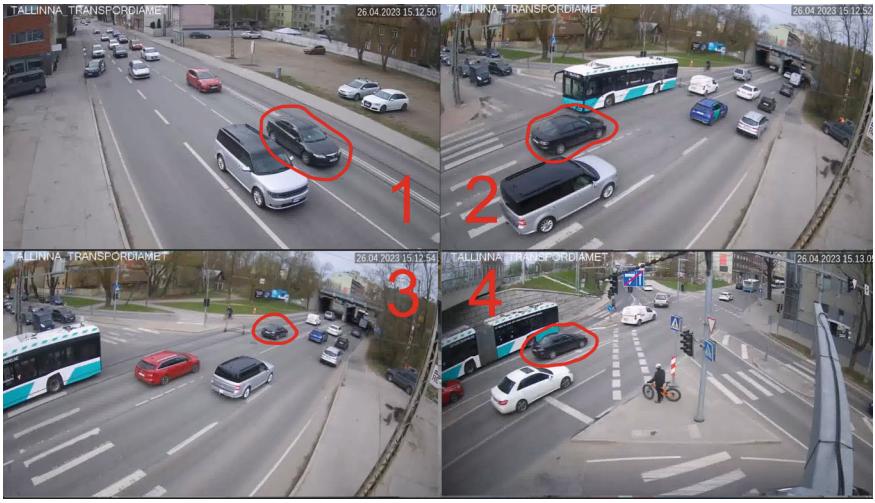


Fig. 1 Test car as seen via multiple cameras in the app

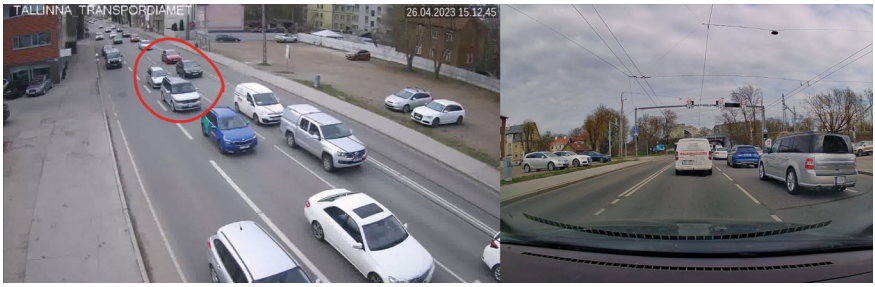


Fig. 2 Different views from dash cam and the app

3.1 Comparison with a Dash Camera

When comparing the dash camera with intersection cameras, notable differences emerge. The dash camera fails to capture the red car trailing behind the monitored vehicle and the white car adjacent to it. In contrast, intersection cameras clearly depict these vehicles, making them distinguishable by shape and color (Fig. 2). Conversely, intersection cameras continue to provide additional information that the dash camera does not. One significant advantage of intersection cameras is their capacity to capture a broader perspective of the vehicle's surroundings, even before an accident occurs. For example, the dash camera only records an intervening vehicle when it reaches the front of the recording car, while the intersection camera documents the vehicle's trajectory over a more extended period, including its presence on the road before entering the intersection. Therefore, intersection cameras offer a much wider view of the car's surroundings. This extended perspective can be invaluable in comprehensively assessing the progression of a traffic accident, allowing for observations from all sides and even providing an aerial view when necessary.

3.2 Usability for Pedestrians or Cyclists

The Tallinn traffic cameras primarily monitor the central portion of intersections, often leaving pedestrian crosswalks at a considerable distance from their field of view. Due to the small size of bicycles compared to cars and the distance from the cameras, bicycles are barely distinguishable. The image compression algorithm also tends to overlook bicycle details, rendering the bicycle almost invisible upon closer inspection. In addition, while the cameras cover main routes for cars, pedestrians have more alternative paths, which might not fall within the view of any intersection camera, as illustrated in the example where the nearest camera is too far from the pedestrian.

In summary, intersection cameras offer a better overview of the events around the monitored vehicle compared to onboard cameras, especially concerning what occurs

behind and beside the vehicle. However, since the number of intersections equipped with cameras is limited, the ideal approach would be to use both onboard cameras and the developed application to ensure coverage beyond intersections. Moreover, intersection camera recordings lack audio, while most onboard cameras record sound. However, this solution works well for vehicles but not for cyclists and pedestrians. The video has too low a resolution for this purpose, making it challenging to locate pedestrians in the images.

4 Discussion and Conclusions

Our analysis revealed that network of Tallinn's traffic cameras is usable for providing a video feed relevant to potential traffic incidents, useful for finding out the circumstances of the incident. Over 50% of the intersections in the city center are already covered by these cameras, providing a solid foundation for our application. Reliability issues were observed in a limited number of cameras, mainly related to weather conditions and construction-related outages. The approach needs to be extended in order to address these challenges, by automatically using the weather forecast for individual cameras (as this allows to predict the potential changes in camera visibility), and by taking into consideration the historical data on the behavior of individual cameras (failures due to the weather exposure, light pollution or vandalism). Despite these challenges, the visibility of cameras at intersections was generally adequate, suggesting that the existing infrastructure of Tallinn's traffic cameras aligns well with our project's goals.

When assessing the usability of intersection cameras in the context of documenting traffic incidents, it becomes evident that they offer several advantages over traditional dash cameras. Intersection cameras provide a broader perspective, capturing events from multiple angles and even before the cars reach the intersection. This extended view can be invaluable in understanding the progression of a traffic accident comprehensively. However, it is important to note that while intersection cameras excel in capturing events involving vehicles, they fall short when it comes to pedestrians and cyclists. The limited resolution and frame rate of these cameras make it challenging to identify individuals, especially in scenarios involving pedestrians and cyclists. Therefore, for these specific user groups, onboard cameras with higher resolution and audio recording capabilities remain more suitable.

4.1 Limitations and Future Directions

The application we developed under this study is just a prototype with the purpose of highlighting the possibilities of using traffic camera networks to uphold traffic security. The application features several critical and valuable functionalities. Firstly, the essential additional feature is the ability for users to download their videos within the

mobile app. This feature is crucial for ensuring user privacy, as it prevents administrators from accessing users' videos, aligning with data protection laws. Another essential functionality is limiting the video storage to the last 5–10 min, rather than saving the entire journey. This aligns with data protection principles and minimizes the amount of video data stored—although it also requires the phones to have a crash detection feature similar to iPhone 14 or later [11]. In addition, the client application should be available on multiple platforms, including iPhone devices and smartwatches, as it is built on platform-independent NET Maui. Improving the user interface, including indicators for GPS signal and functionality descriptions, is also important for enhancing the user experience. The application's functionality could be further enhanced by integrating it with traffic light statuses to provide real-time information. This could enhance the app's utility by providing insights into traffic flow and safety.

The application can be further improved by incorporating automated detection methods and machine learning in order to analyze large datasets effectively and to address changing environmental factors and seasonal variations.

References

1. Merriam-Webster (2023) Dashcam definition and meaning. [Online] Available at: https://www.merriam-webster.com/dictionary/dashcam
2. Grand View Research (2021) Dashboard camera market size and share analysis report. [Online] Available at: https://www.grandviewresearch.com/industry-analysis/dashboard-camera-market
3. Kim T, Jung IY, Hu Y-C (2020) Automatic, location-privacy preserving dashcam video sharing using blockchain and deep learning. Human-centric Comput Inf Sci 10(1):36. [Online] Available at: https://doi.org/10.1186/s13673-020-00244-8
4. Fedorov A, Nikolskaia K, Ivanov S et al (2019) Traffic flow estimation with data from a video surveillance camera. J Big Data 6:73. [Online] Available at: https://doi.org/10.1186/s40537-019-0234-z
5. Yang H, Cai J, Zhu M, Liu C, Wang Y (2022) Traffic-informed multi-camera sensing (TIMS) system based on vehicle re-identification. IEEE Trans Intell Transp Syst 23(10):17189–17200. [Online] Available at: https://doi.org/10.1109/TITS.2022.3154368
6. Tallinn Transport Department (2023) Cams Front/Tallinna Liikluskaamerad. [Online] Available at: https://ristmikud.tallinn.ee/index.php/cams
7. Nõugast R (2023) Personal communication, 8 Feb 2023
8. Leoste KA, Leoste JV (2023) Asukohapõhine linnaruumi liikluskaamerate videovoogu salvestav rakendus (Unpublished master's thesis). Tallinn University, Tallinn. [Online] Available at: https://www.etera.ee/zoom/200425/view?page=1&p=separate&tool=info
9. Okta (2023) Auth0 Docs. [Online] Available at: https://auth0.com/docs/
10. European Data Protection Board (2019) Guidelines 3/2019 on processing of personal data through video devices. [Online] Available at: https://edpb.europa.eu/sites/default/files/consultation/edpb_guidelines_201903_videosurveillance.pdf
11. Apple (2023) Use crash detection on iPhone or Apple Watch to call for help in an accident. Apple Support. [Online] Available at: https://support.apple.com/en-us/HT213225

A Multiband Sierpinski Arrowhead Fractal Antenna with 3 × 3 CSRR Array for Various Wireless Applications

Indraja Lekkala, Likitha Mandyam, Jayanthi Chittipothala, T. Manivannan, E. Sathiyanarayanan, and E. Aravindraj

Abstract In the proposed work, a multiband Sierpinski arrowhead fractal antenna with 3 × 3 Circular Split Ring Resonator (CSRR) Array integrated ground for various wireless applications is presented. The implementation of third-order Sierpinski arrowhead fractal structure makes the prototype occurs in a compact cross-sectional area with multiband resonances. The complete dimension of the antenna is 60 × 70 × 1.6 mm^3 which is made using a FR-4 dielectric material sandwiched by Sierpinski arrowhead radiator and 3 × 3 CSRR Array integrated ground. The CSRR generates magnetic susceptibility media to establish a robust magnetic connection with an applied electromagnetic field. The proposed fractal printed monopole antenna design exhibits resonances from 4.54 to 4.83 GHz, 7.14 to 7.96 GHz, and 11.71 to 12.26 GHz coupled with an average gain of 5.1 dB. The fabricated prototype obtains a strong alignment during both testing and measurement processes, experimented using a Microwave analyzer. The antenna resonates effectively at 5G (n79) band and Satellite Communication and RADAR applications.

Keywords Circular split ring resonators · Fractional bandwidth · FR-4 substrate · Gain · Sierpinski arrowhead fractal

I. Lekkala · L. Mandyam · J. Chittipothala · T. Manivannan · E. Sathiyanarayanan · E. Aravindraj (✉)
Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science, Angallu, Andhra Pradesh, India
e-mail: aravindraj1794@gmail.com

I. Lekkala
e-mail: indrajalekkala2003@gmail.com

L. Mandyam
e-mail: likithareddy310@gmail.com

J. Chittipothala
e-mail: jayanthichittipothala@gmail.com

T. Manivannan
e-mail: manivannant79@gmail.com

E. Sathiyanarayanan
e-mail: sathiyaame@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_30

1 Introduction

The wider range of microwave frequency coverage in antenna development has become more essential in our everyday activities. To meet the evolving requirements of advancements in radio communication, a viable approach involves employing a single-component radiator capable of spanning a broader spectrum of microwave frequencies [1]. In the realm of mobile communication and diverse modern settings, it is crucial to carefully consider both the size and profile of the antenna. Traditional antennas may not sufficiently meet this specific requirement [2]. Effective radiators are favored for their capacity to seamlessly align with different transceiver models and offer protection against multipath interference. Consequently, printed monopole antennas find application in specific scenarios where there is a need for size reduction and dynamic utilization across a broader range of frequencies [3]. Enhancing antenna design beyond traditional planar configurations includes integrating a fractal structure into the radiating element. This arrangement reduces the antenna's dimensions and enables it to operate efficiently over an extended frequency range, encompassing both lower and higher frequencies [4]. Fractal frameworks are self-replicating structures that exhibit progressively fractional configurations identical to previous levels in Euclidean space [5]. Integrating fractal patterns into printed monopole antennas results in a significant decrease in antenna size, an extension of the effective length of the substrate, and an enhancement in the propagation of electromagnetic radiation across the surface area [6, 7]. These effects facilitate the development of next-generation electronic communicators and mobile phones [8, 9].

Within the conventional techniques employed for fractal generation, the Iterated Function System (IFS) is specifically applied to attain accurate self-similar structures at designated vertices. The resonance of the antenna is fine-tuned and tailored by selecting a suitable fractal design [10]. The proposed design integrates the Sierpinski arrowhead fractal antenna, resulting in a reduced cross-sectional area for the radiator while achieving broad resonance with enhanced gain [11]. In the proposed design, an electric field is directed to the printed patch or radiator (Sierpinski arrowhead) through the utilization of a microstrip, and a lumped port is employed to facilitate the transfer of electric fields between the radiator and the ground [12]. Hence, the proposed design utilizes an FR-4 circuit board substrate as the dielectric material, possessing distinctive characteristics such as a dielectric constant (ε_r) of 4.4 and a loss tangent (δ) of 0.02 [13]. The integration of a 3×3 CSRR Array in ground enables the attainment of metamaterial characteristics to achieve negative permeability. Recent literature has highlighted the use of CSRRs for generating multiband operation in printed monopole antennas. The periodic structure, utilizing split rings, is capable of exhibiting negative permeability, while a configuration of parallel pins can demonstrate negative permittivity. The combined effect of these elements results in an effective frequency-dispersive permeability.

2 Antenna Design Methodologies

2.1 Antenna Configuration

The antenna configuration consists of a third-order Sierpinski arrowhead fractal-shaped radiator mounted on an FR-4 substrate, fed via a microstrip line. The design methodology encompasses the antenna's setup, integration of the fractal, and the underlying operational principles of a generic antenna layout [14]. The standard representation for indicating the dimensions of the radiator width and length is designated as W and L.

$$W = \frac{c}{2f_r\sqrt{\frac{(\varepsilon_r+1)}{2}}} \tag{1}$$

$$L = \left[\frac{C}{2f_r\sqrt{\varepsilon_{\text{eff}}}}\right] - 2\left[\frac{0.412\, h(\varepsilon_{\text{eff}}+0.3)\left(\frac{W_p}{t}+0.264\right)}{(\varepsilon_{\text{eff}}-0.258)\left(\frac{W_p}{t}+0.8\right)}\right] \tag{2}$$

$$\varepsilon_{\text{eff}} = \frac{\varepsilon_r+1}{2} + \frac{\varepsilon_r-1}{2}\left[1+12\frac{t}{w}\right]^{-1/2} \tag{3}$$

The dimensions of the substrate and ground plane are denoted as (W_s & L_s) and (W_g & L_g) respectively, while the scales of the radiator are represented as (W & L), and the feed line dimensions are expressed as (W_f & L_f). The effective dielectric constant is denoted by ε_{eff}. These values are determined through the utilization of various mathematical expressions.

$$\mu_r = 1 - \left[\frac{f_{\text{mp}}^2 - f_{\text{mo}}^2}{f^2 - f_{\text{mo}}^2 + i\gamma_m f/2\pi}\right] \tag{4}$$

The magnetic damping factor, denoted as γ_m, the magnetic resonant (f_{mo}), and plasma frequencies (f_{mp}) are defined, with f_{mo} being less than f_{mp}, i.e., ($f_{\text{mo}} < f_{\text{mp}}$). The permeability between these frequencies takes on negative values, and these specific frequencies directly depend on the circular ring's shape, size, metallization, radii, and other parameters.

The resonant frequency (f_r = 1.8 GHz), thickness of substrate (t = 1.6 mm), dielectric constant of FR4 (dielectric constant ε_r = 4.4; loss tangent δ = 0.02), and velocity of light ($c = 3 \times 10^8$ m/s) are some of the essential parameters for the calculations. In designing a conventional radiator, the factors that impact the measurement of the associated ground and substrate are the radiator width and length [15]. As shown in Table 1, the antenna occupies a volume of 60 × 70 × 1.6 mm^3 and is activated through a lumped port, while it is enclosed by an open radiation boundary. The specifications outlined in above table, the Sierpinski arrowhead fractal

Table 1 Parameters and dimensions of Sierpinski arrowhead antenna

Sections of antenna	Parameters	Dimensions (mm)
Substrate	Length (L_s)	70
	Width (W_s)	60
	Thickness (t)	1.6
Radiator	Side (a)	30.31
	Width (W)	34.64
	Side Notch (s)	5
Ground	Length (L_g)	70
	Width (W_g)	60
Feed line	Length (L_f)	20
	Width (W_f)	4.5
Split ring resonators (SRR)	Radii 1 (R_1)	8
	Radii 2 (R_2)	6.5
	Radii 3 (R_3)	5
	Radii 4 (R_4)	3.5
	Width of Split (W_1)	2
	Width of Split (W_2)	2

antenna was designed using ANSYS Electronic Desktop, and Fig. 1a, b denote the specifications of proposed antenna in front and rear view, respectively.

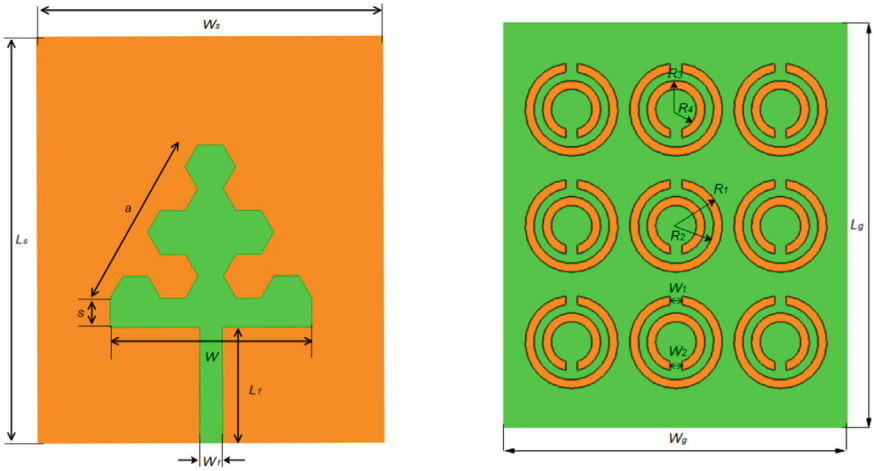
$$\mathrm{Zo} = \frac{60\ \ln\left[\frac{S_1 t}{w_f} + \sqrt{1 + \left(\frac{2t}{w_f}\right)^2}\right]}{\sqrt{\varepsilon_{\mathrm{eff}}}} \tag{5}$$

$$S = 6 + (2\pi - 6)e^{-\left(30.666h/w_f\right)^{0.7528}} \tag{6}$$

In this context, Zo represents the characteristic impedance of the antenna when using microstrip feeding, and $S1$ provides the corresponding expressions for Zo, wherein the width of the feed (W_f) is employed [16]. The proposed antenna yields an approximate calculated value of 50 Ω impedance between the coaxial transmission line and the radiating element.

2.2 *Implementation of Sierpinski Arrowhead Fractal Antenna*

The Sierpinski arrowhead geometry is built upon gasket fractal structure which begins with an equilateral triangle as its base shape. Initially, the first-order equilateral



(a) Front view (b) Rear view

Fig. 1 Specifications of Sierpinski arrowhead fractal antenna

triangle is subdivided into three equal parts on each side, resulting in new lengths that are arranged to form an additional triangle as a branch. To calculate the finite points and their positions within the vertices, the relative values of first-order affine transformation matrices are utilized. Figure 2 illustrates Sierpinski arrowhead fractal patterns with three progressively higher levels of iterations.

Sierpinski gasket fractals are self-replicating formations generated through the Iterated Function System (IFS) method coupled with a generic algorithm. This method entails depicting fractal patterns within a finite set of complete metric spaces, employing an affine transformation matrix represented as $v(x, y)$.

$$v(x, y) = \begin{bmatrix} p & q \\ r & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{7}$$

$$v(x, y) = (px + qy + a, rx + sy + b) \tag{8}$$

1 2 3

Fig. 2 Increasing levels of Sierpinski arrowhead fractal with respect to gasket fractal

When taking into account the rising level of the contraction factor, the fractal geometry is delineated by a succession of complex linear matrices, referred to as the Hutchinson operator (V). This operator determines the vertices at each successive iteration level.

$$V(A) = \bigcup_{n=1}^{3} v_n(x, y) \tag{9}$$

The utilization of the Hausdorff–Besicovitch formula is aimed at determining the positioning of a fractal structure within the Euclidean dimension (D). This is accomplished by counteracting its small-scale impact through the inversion of the scaling factor (S) while ensuring equilibrium in the count of self-similar branches (N). Hence, the representation of the Hausdorff–Besicovitch formula is as follows:

$$N = S^D \tag{10}$$

$$\log_{10}(N) = D \log_{10}(S) \tag{11}$$

The affine transformation matrices for the proposed design of the Sierpinski gasket fractal are represented below:

$$v(x, y) = \begin{bmatrix} a\cos\theta & c\sin\theta \\ b\sin\theta & d\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{12}$$

$$v_1(x, y) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix} \tag{13}$$

$$v_2(x, y) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{1}{4} \\ \frac{\sqrt{3}}{4} \end{bmatrix} \tag{14}$$

$$v_3(x, y) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{3}{4} \\ \frac{\sqrt{3}}{4} \end{bmatrix} \tag{15}$$

As depicted in Fig. 3, the Sierpinski arrowhead fractal pattern emerges by tracing the curve around the solid triangle, using the four edges of the initial triangle as the vertices. Concerning the three outer triangles, the middle boundary is eliminated. The first order is obtained by scaling three replicas of the 0th order, each by a factor of 1/2, and subsequently translating two of the smaller triangles to create the intended configuration. Therefore, the outcome of intersecting all sets within this sequence essentially represents the collection of points that persist after the repetition of this construction an infinite number of times.

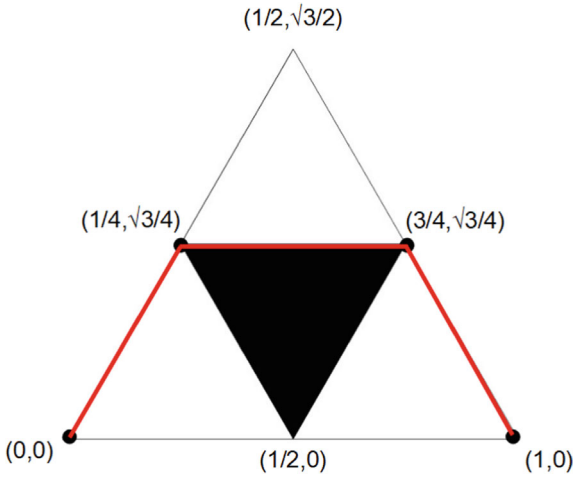
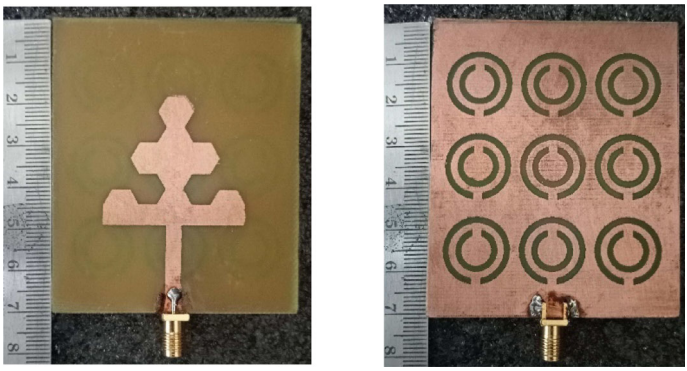


Fig. 3 Construction of first-order Sierpinski arrowhead in the vertices base of gasket fractal

2.3 Operating Principle

The antenna configuration features a Sierpinski arrowhead fractal pattern feed using a microstrip with 3 × 3 CSRR array integrated ground to enhance bandwidth. By introducing cutting truncation and cut edges in the Sierpinski triangles within the arrowhead, the antenna induces greater discontinuity in directed radiation. This design aims to progressively reduce reflections, leading to improved bandwidth. The edge-truncated fractal structure in the radiator of the antenna facilitates efficient reflection and current distribution at the surface boundary of the printed monopole. Incorporating CSRR into the antenna structure is to modify the propagation conditions across the structure, causing a downward shift in the resonant frequency. The shift can be realized by introducing the CSRR on the ground plane. The iteration began with a single resonator beneath the patch, and the number was systematically increased to strike a balance between size reduction and minimizing leakage through the ground plane. As anticipated, a higher number of CSRRs resulted in a smaller antenna size. Additionally, various configurations, such as the size and width of the CSRR, were explored for examination.

As outlined in the preceding section, the Sierpinski arrowhead pattern constructed and traced at a third-order iterative level using the Iterated Function System (IFS) technique. This integration aims to reduce the antenna's size and enhance its radiation characteristics. The decision to limit the iteration level to the third order is to avoid discrepancies in antenna measurements during the fabrication process, particularly considering the prototype's scale at a few millimeters. The antenna is designed and analyzed using the high-frequency structure simulator (HFSS) tool, and the fabrication employs the milling method. The proposed Sierpinski arrowhead fractal antenna has overall dimensions of 60 × 70 × 1.6 mm^3. Figure 4a, b depict the front and rear view of the physical antenna prototype manufactured through milling.



(a) Front view (b) Rear view

Fig. 4 Fabricated Sierpinski arrowhead fractal antenna

3 Results and Discussion

3.1 Bandwidth

Upon scrutinizing the S11 parameter of both designed and manufactured antennas, it becomes clear that both antennas offer an extended bandwidth and demonstrate a notable correlation. In this context, the − 10 dB range acts as the standard for identifying functional frequencies. Based on the return loss plot presented in Fig. 5, it is observed that the simulated antenna featuring the Sierpinski arrowhead fractal pattern functions over multiple frequency ranges, spanning from 4.54 to 4.83 GHz, 7.14 to 7.96 GHz, and 11.17 to 12.26 GHz.

The fabricated antenna incorporating the Sierpinski arrowhead fractal pattern exhibits operation across the frequency band of 4.16–5.06 GHz, 7.34–8.18 GHz, and 11.66–12.14 GHz.

Upon comparison, it becomes evident that the simulated and experimented antennas featuring the Sierpinski arrowhead fractal pattern outperform the 1st and 2nd iteration levels. Consequently, both simulated and experimented antennas showcase an expanded bandwidth and exhibit a consistent agreement with each other. The S11 comparison of simulated and fabricated Sierpinski arrowhead fractal antenna is shown in Fig. 6.

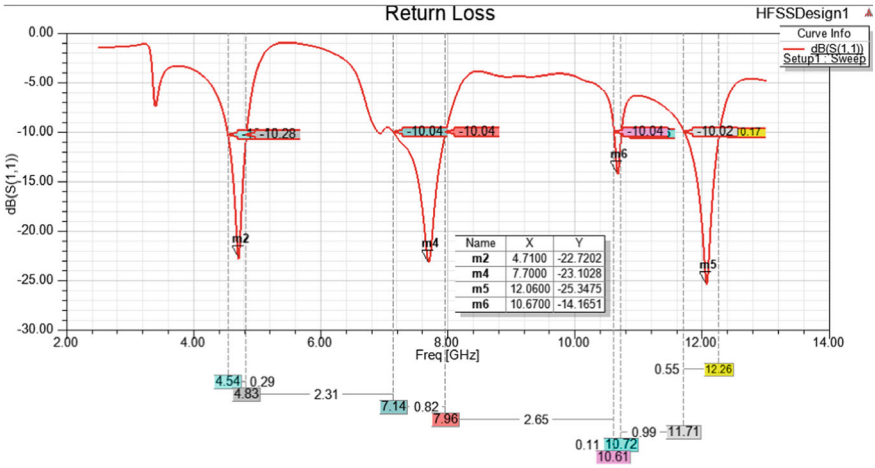


Fig. 5 Bandwidth of Sierpinski arrowhead fractal antenna

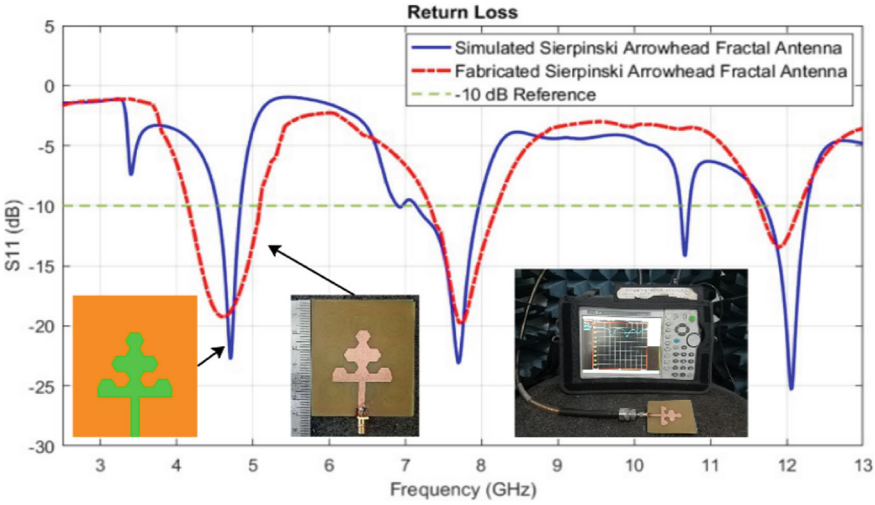
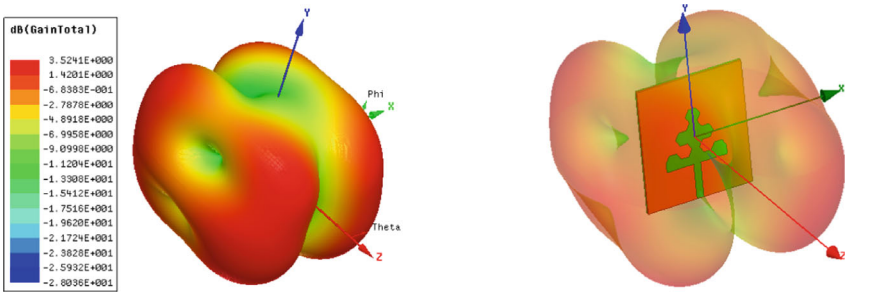


Fig. 6 S11 comparison of simulated and fabricated Sierpinski arrowhead fractal antenna

3.2 Gain

Figure 7a illustrates the gain of the simulated antenna, while Fig. 7b showcases the orientation and 3D perspective of the radiation pattern at 4.7 GHz. A thorough examination of the three-dimensional gain pattern reveals the antenna's omnidirectional radiation characteristic, allowing it to emit signals in all directions across a 360-degree span.



(a) 3D Gain (b) 3D Gain with reference to the Antenna

Fig. 7 Gain of simulated Sierpinski arrowhead fractal antenna

At a frequency range from 2.5 to 13 GHz, the simulated antenna achieves a peak gain of 6.52 dB and a peak directivity value of 6.91 dB at 10.5 GHz resonant frequency. Given the symmetrical structure of the proposed antenna, this study emphasizes the far-field results during feed excitation. From the observation, it is evident that the average gain and directivity achieved by the propose antenna are 5 dB and 5.25 dB, respectively. Figure 8 provides a visual representation of the gain and directivity values of the antenna in the y-direction.

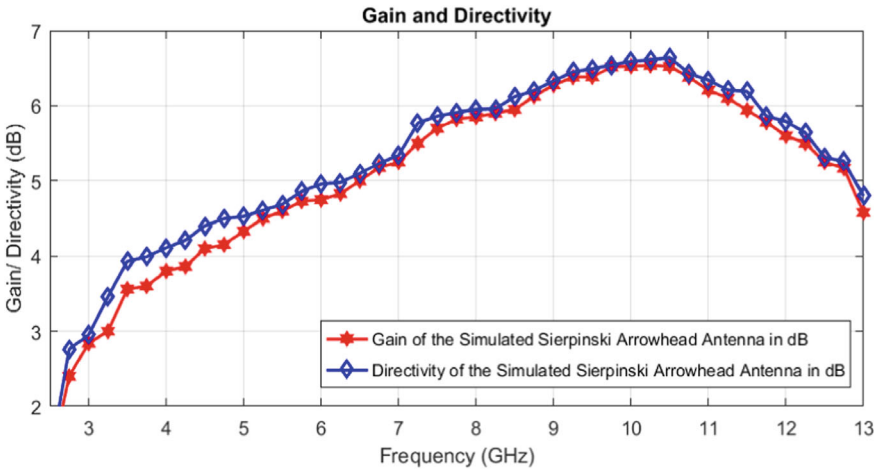
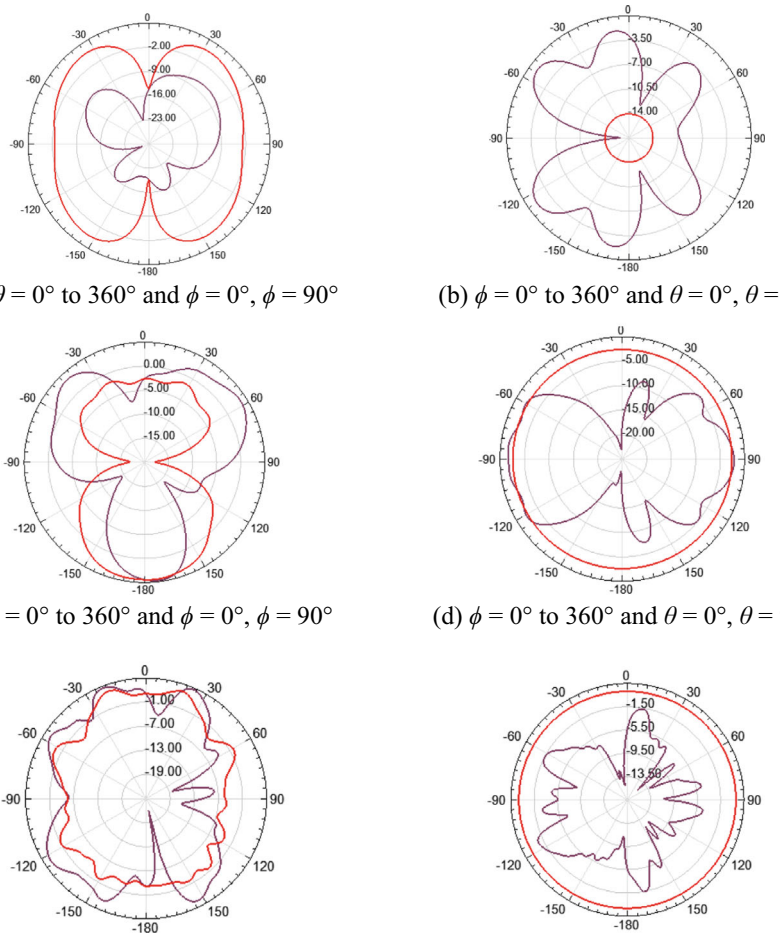


Fig. 8 Gain and directivity of simulated Sierpinski arrowhead fractal antenna

3.3 Radiation Pattern

The radiation patterns of the proposed antenna at resonant frequencies (4.7, 7.7, and 12.1 GHz) are examined in the xz-plane and yz-plane for total gain measurements, covering $\theta = 0°$–$360°$ and $\phi = 0°$, $\phi = 90°$; $\phi = 0°$–$360°$ and $\theta = 0°$, $\theta = 90°$, as illustrated in Fig. 9a–f.



(a) $\theta = 0°$ to $360°$ and $\phi = 0°$, $\phi = 90°$

(b) $\phi = 0°$ to $360°$ and $\theta = 0°$, $\theta = 90°$

(c) $\theta = 0°$ to $360°$ and $\phi = 0°$, $\phi = 90°$

(d) $\phi = 0°$ to $360°$ and $\theta = 0°$, $\theta = 90°$

(e) $\theta = 0°$ to $360°$ and $\phi = 0°$, $\phi = 90°$

(f) $\phi = 0°$ to $360°$ and $\theta = 0°$, $\theta = 90°$

Fig. 9 Radiation pattern of simulated Sierpinski arrowhead fractal antenna

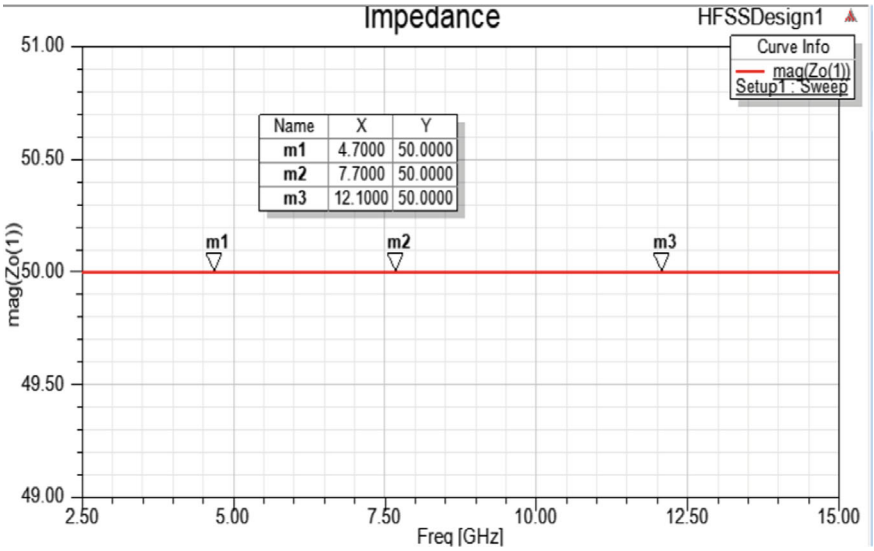


Fig. 10 Impedance

3.4 Impedance and Current Distribution

The impedance of the proposed antenna at resonant frequencies (4.7, 7.7, and 12.1 GHz) is examined and it is observed that all the resonating frequency provides impedance matching at 50 Ω. Figure 10 depicts the impedance plot of the proposed antenna.

Figure 11 illustrates the current distribution of the simulated antenna, focusing on the electric field distribution on the arrowhead patch's surface and the CSRR ground. This current distribution is measured at the resonance frequency of 4.7 GHz, providing insights into power levels at the antenna's corners and edges.

In Table 2, a comparative analysis is presented for the proposed work, including simulated results and experimented results, in relation to recent studies. Parameters considered in the comparison encompass antenna dimensions, radiated frequencies, bandwidth, and antenna gain. The table clearly indicates that the proposed antenna excels in terms of output parameters.

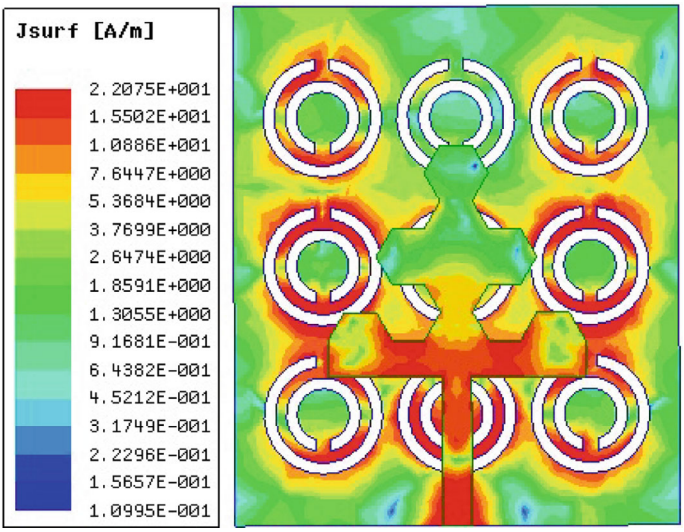


Fig. 11 Current distribution

Table 2 Comparison of proposed work with some recent existing works

Reference/ Proposed work	Dimensions (mm^3)	Frequency range/ Resonant frequency (GHz)	Bandwidth (GHz)	Average gain (dB)
[17]	52 × 71 × 5.1	2.3–2.56	0.26	4.56
[18]	60 × 70 × 4	0.73–0.98 /1.6–3.1	1.47	3.8
[19]	65 × 65 × 1.6	2.35–3.79	1.44	4.1
[20]	120 × 120 × 1.6	1.7/ 3.5/ 6.9	0.7	3.9
[21]	52 × 71 × 5.1	2.43–2.475	0.45	3.1
Sim. Results	**60 × 70 × 1.6**	**4.54–4.8/7.14–7.96/ 11.7–12.2**	**1.66**	**5.1**
Fab. Results	**60 × 70 × 1.6**	**4.1–5/ 7.3–8.1/ 11.6–12.14**	**1.8**	–

4 Conclusion

A multiband Sierpinski arrowhead fractal antenna with 3 × 3 Circular Split Ring Resonator (CSRR) Array integrated ground is designed, analyzed, fabricated, tested, and compared. The complete dimension of the antenna is 0.95 λ_0 × 1.1 λ_0 × 0.025 λ_0. The proposed printed monopole antenna design exhibits a fractional bandwidth (FBW) of 126%, which produced by the resonances from 4.54 to 4.83 GHz, 7.14 to 7.96 GHz, and 11.71 to 12.26 GHz coupled with an average gain of 5.1 dB.

The fabricated prototype obtains a strong alignment during both testing and measurement processes, experimented using a Microwave analyzer. The antenna resonates effectively at 5G (n79) band and Satellite Communication and RADAR applications.

References

1. Visser HJ (2012) Antenna theory and applications. Wiley
2. Balanis CA (2016) Antenna theory: analysis and design, 4th edn. Wiley
3. Milligan TA (2005) Modern antenna design. 2nd edn. Wiley
4. Barnsley MF (1993) Fractals everywhere, 2nd edn. Academic Press, Georgia
5. Allin Joe D, Rishi R, Yathin Kumar R, Prasanna Balaji E (2023) Fractal antenna design with slots for multiband applications. In: International conference on advancements in electrical, electronics, communication, and automation. IEEE, Coimbatore, India, pp 1–6
6. Falconer K (1997) Techniques in fractal geometry. Wiley
7. Mandelbrot B (1982) The fractal geometry of nature. W. H. Freeman & Co., New York, USA
8. Elkorany AS, Mousa AN, Ahmad S, Saleeb DA, Ghaffar A, Soruri M, Dalarsson M, Limiti E (2022) Implementation of a miniaturized planar tri-band microstrip patch antenna for wireless sensors in mobile applications. Sensors 22(2):1–13
9. Pandey U, Singh P, Singh R, Gupta NP, Arora SK, Nizeyimana E (2023) Miniaturized ultra-wideband microstrip antenna for IoT-based wireless body area network applications. Wirel Commun Mob Comput 23:1–19
10. Ezhumalai A, Ganesan N, Balasubramaniyan S (2021) An extensive survey on fractal structures using iterated function system in patch antennas. Int J Commun Syst 34(2):1–38
11. Raj A, Dhubkarya DC, Srivastava DK, Mandal D (2023) Design and analysis of square shape slot cut high gain sierpinski carpet fractal antenna for wireless applications. Microw Opt Technol Lett 65(8):2337–2343
12. Markkandan S, Malarvizhi C, Raja L, Kalloor J, Karthi J, Atla R (2021) Highly compact sized circular microstrip patch antenna with partial ground for biomedical applications. Mater Today Proc 318–320
13. Nejdi IH, Bri S, Marzouk M, Ahmad S, Rhazi Y, Lafkih MA, Sheikh YA, Ghaffar A, Hussein M (2023) UWB circular fractal antenna with high gain for telecommunication applications. Sensors 23(8):1–15
14. Sood M, Rai A (2023) Design of compact fractal patch antenna for X/Ku/K-band applications. Int J Commun Syst 36(5):1–13
15. Aravindraj E, Nagarajan G, Kumaran RS (2020) Design and analysis of recursive square fractal antenna for WLAN applications. In: International conference on emerging trends in information technology and engineering. IEEE, Vellore, India, pp 1–5
16. Sharma S, Tripathi C, Rishi R (2017) Impedance matching techniques for microstrip patch antenna. Indian J Sci Technol 10(28):1–16
17. Kalaiyarasan R, Nagarajan G, Seenuvasamurthi S (2023) Design and implementation of an efficient sierpinski carpet fractal antenna with defected ground structure for 2.45 GHz applications. SN Comput Sci 4(6):1–13
18. Nsir CB, Boussetta C, Ribero J-M, Gharsallah A (2021) Dual-band fractal antenna with bandwidth improvement for wireless applications. Int J Comput Sci Netw Secur 21(12):75–80
19. Yazid MYI, Baharuddin MH, Islam MS, Almutari AF (2023) A Sierpinski arrowhead curve slot Vivaldi antenna for microwave head imaging system. IEEE Access
20. Porcu I, Maccio C, Muntoni G, Fanti A (2023) Design of a Sierpinski fractal antenna for breast tumors diagnosis. In: Photonics & electromagnetics research symposium. IEEE, Prague, Czech Republic, pp 1698–1704
21. Kalaiyarasan R, Nagarajan G, Kumaran RS (2023) Design and implementation of an efficient Sierpinski carpet antenna for 2.45 GHz applications. SN Comput Sci 4(6):1–9

CKGLD: Curating Knowledge Graph for Linked Open Data Generation for Cyber Law and Ethics

A. Aravind Krishnan, Gerard Deepak, A. Santhanavijayan, and K. R. Venugopal

Abstract This paper introduces a knowledge graph curation framework for the generation of linked open data in the domain of cyber law and ethics called the CKGLD. This framework utilizes both static metadata and dynamically acquired knowledge from Web 3.0. With advanced methods of term and category extraction, classification using both lightweight machine learning and strong deep learning methods, and clustering, the framework can exceed all similar frameworks in terms of performance. Initial terms are processed from the documents using the TF-IDF strategy. Static metadata stack is established from e-books, query logs, and glossaries. Renyi entropy is used to guide informative term selection for a decision tree classification; in this manner, the scale of the generated knowledge is managed by lightweight computing requirements. Dynamic knowledge from Web 3.0 is classified using GANs and is subjected to DBSCAN clustering. A multi-agent setup with parallelization enhances the efficiency of the clustering process. The knowledge graph is finally created using the Co-SIM rank and is evaluated for specificity using Hellinger's distance and volume of intersection index. These paradigms contribute to this framework significantly outperforming the baseline frameworks.

Keywords Knowledge graph · Linked open data · Cyber law and ethics · GANs · DBSCAN

A. Aravind Krishnan · G. Deepak (✉)
Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru, India
e-mail: gerard.deepak.christuni@gmail.com

A. Santhanavijayan
National Institute of Technology, Tiruchirappalli, India

K. R. Venugopal
Bangalore University, Bengaluru, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_31

1 Introduction

The progression from Web 2.0 to Web 3.0 marks an important transformation in how information from the web is utilized and accessed. The transition to Web 3.0 is characterized by a more intelligent, decentralized, and semantic structure, making the need for a knowledge graph curation framework more apparent. In the context of cyber law and ethics, there is a large cognitive gap between the current abilities of the web and actual implementation, and a knowledge graph is an essential paradigm in addressing the organization of the vast data that accompanies this domain.

Cyber law and ethics constitute an important field at the intersection of technology and legal frameworks. Cyber law encompasses regulations and policies that govern online activities, digital transactions, and IP rights. Cyber ethics investigates the ethical implications of technological use and responsible use of technology. Together they serve as an important pillar in shaping a secure digital environment and societal online wellbeing.

Open-linked data is the decentralized approach to sharing structured information on the web, allowing different types of datasets to be more accessible. This has grown to be the need of the hour as it promoted the very interconnectedness that is required by the standards of Web 3.0. This interconnected data structure is crucial for leveraging the full potential of the web, promoting collaboration, and effective and efficient information retrieval.

Motivation: The primary motivation of the proposed framework is a lack of linked open data or knowledge graphs or ontologies or represented linked knowledge for cyber law and ethics as the domain. Digital knowledge focusing on cyber law and ethics curated for the web is absent, and automatic strategies that encompass semantics-oriented machine learning with semantics-driven reasoning learning for fact-driven knowledge graph generation for cyber law and ethics are absent.

Contributions: The framework's contributions lie in the creation of static and dynamic knowledge for cyber law and ethics. Key innovations include strategic term extraction using TF-IDF, Renyi entropy-guided feature selection for a lightweight decision tree classifier, and the use of GANs for dynamic knowledge classification. DBSCAN clustering over static and dynamic entities ensures a well-formed, domain-specific knowledge graph. The multi-agent approach with JADE enhances clustering efficiency, addressing the challenge of knowledge scale and specificity. The knowledge map, Co-SIM rank, Hellinger's distance, and volume of intersection index collectively contribute to a robust framework for linked open data generation.

Organization: Section 2 depicts the Related Works, Sect. 3 depicts Proposed System Architecture, Sect. 4 depicts Implementation, Performance Evaluation, and Results, and Sect. 5 depicts the Conclusion.

2 Related Works

Huaman et al. [1] have proposed a practical framework to enhance the quality of knowledge graphs by introducing defined metrics, cleaning tasks, and strategies for duplication detection and knowledge fusion. Lakshmi et al. [2] have introduced a knowledge-centric method of leveraging machine learning and semantic inference to curate large-scale knowledge graphs from text using ontologies. Hoyt et al. [3] have developed workflows for re-curation and rational enrichment that apply to NeuroMMSig's KGs using text mining from MEDLINE and PubMed Central.

Iglesias et al. [4] have introduced a declarative approach for efficiently generating RDF knowledge graphs from heterogeneous data using a partitioning and scheduling strategy. Ma et al. [5] have provided a comprehensive knowledge graph construction and implementation in geosciences, covering concepts, approaches, and applications in data life cycle, data analysis, and highlighting challenges and trends for future development. Kumar et al. [6] have introduced a novel graph-based structure called Groups that can enhance the efficiency of DBSCAN by 1.5–2.2 times on benchmark datasets. Qaiser et al. [7] have discussed the application of TF-IDF in assessing keyword relevance to documents, covering implementation steps, results, and strengths and weaknesses. Daradkeh et al. [8] have proposed the HC-LTM, an improved Lifelong Topic Model addressing biases and improving knowledge acquisition. Wu et al. [9] have addressed catastrophic forgetting in incremental classifier learning by introducing a new loss function, addressing class imbalance, and utilizing GANs to generate historical data, demonstrating the effectiveness on CIFAR-100, Flower-102, and MS-Celeb-1M-Base datasets. Oggier et al. [10] have presented a graph clustering method based on Renyi entropy, adopting the viewpoints of embedding the graph and directly clustering it.

3 Proposed System Architecture

Figure 1 illustrates the architecture diagram of the knowledge graph curation framework for the generation of linked open data for cyber law and ethics. Documents about cyber law and ethics are subjected to extraction of terms and categories. They are extracted by targeting the keywords across the documents at a random level and also getting the contexts for the rare terms and preprocessing using tokenization, lemmatization, and stop-word removal to yield the initial terms. Weighted strategic informative term extraction is achieved using TF-IDF where the rare term occurs in the document corpus and the frequent terms within the document corpus are strategically extracted as the informative terms. The informative terms extracted from the dataset further drive the model. A static metadata stack is formalized using the indices of e-books, query logs of cyber lawyers and professionals, glossaries on cyber law and ethics, and crawled metadata. The metadata is crawled using a customized crawler and it is retrieved from the current structure of Web 3.0. Query logs of cyber

lawyers and cyber professionals (domain experts) are subjected to preprocessing. The pre-processed terms from these query logs are all constituted as static metadata which is quite comprehensive and large owing to the number quantity and the volume of e-books that are used in the framework. The top 10% from each of the categories is extracted from the static metadata. The Renyi entropy is calculated for the informative terms discovered through the use of the TF-IDF. Renyi entropy is set to a step-deviance of 0.15 and it computes the most informative terms. This is because a large number of features are required which is sent as input to the decision tree classifier to classify the dataset. The decision tree classifier is preferred here to a strong deep learning model because a feature-controlling mechanism is required, so when features are extracted from static metadata, deviance from the actual domain of cyber law and ethics is preserved. A lightweight algorithm is required because while the framework can be demonstrated on the dataset, it is not as large as the knowledge that will be generated from Web 3.0. Therefore, to preserve the computational cycles and to preserve the deviance from the specific domain, we use a lightweight machine learning classifier rather than a strong deep learning classifier. The classified dataset terms are further used in the model. Dynamic knowledge is generated from Web 3.0 using these terms and categories and is passed into the data fountain metadata generator tool. The dynamic knowledge is generated, and as the metadata is extensively large, it requires classification by deep learning classifiers like the GANs as it is a strategic deep learning classifier that classifies based on implicit or automatic feature selection. Feature selection over very large-scale knowledge generated from the web is very difficult, and hence, implicit feature selection mechanisms with deep learning models like GAN are preferred. Subsequently, DBSCAN clustering is applied over the entities coming out of the GAN classifier of the classified dynamically generated knowledge and the statically generated metadata.

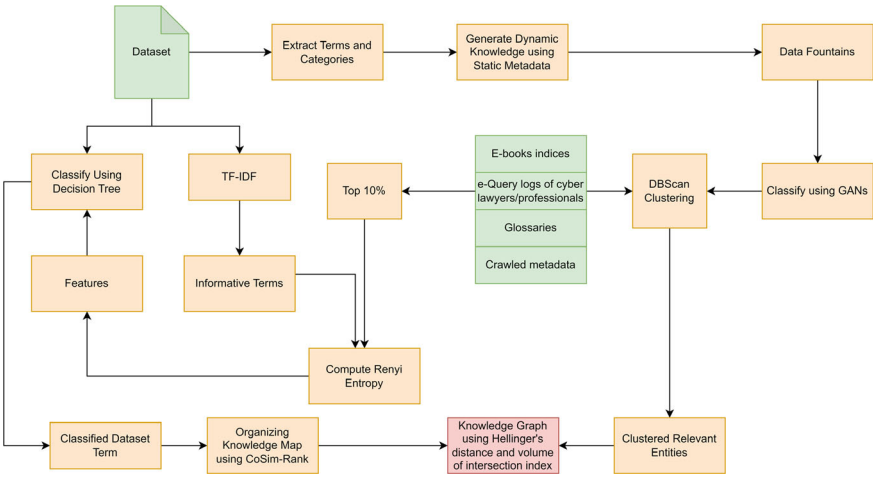


Fig. 1 Architecture of the proposed CKGLD framework

DBSCAN is the preferred clustering algorithm as it can easily cluster similar entities using overlaps between the entities themselves. Since both the static metadata and the dynamically generated knowledge are highly specific to the domain, the overlap is well-formed using DBSCAN clustering using a multi-agent setup where an agent using JADE and a distributed computing paradigm with parallelization of the agents (where the aim of the agent is to compute the semantic relevance and to yield the clustered relevant entities) is made use of. The clustered relevant entities are further used in the model. The classified dataset terms that come out of the decision tree classifier are organized as an initial knowledge map along with the informative terms obtained from the dataset that are discovered using the TF-IDF. The knowledge map is done using the Co-SIM rank measure. It is set to a threshold of 0.75. The clustered relevant entities that come out of the DBSCAN clustering are organized as a large knowledge graph where both these entities are evaluated using Hellinger's distance set to a threshold of 0.10 and volume of intersection index set to a threshold of 0.25 because diversification and a large number of entities are required. Hellinger's distance is not the strongest of measures while the volume of intersection index is considerably strong, there is a variation of the threshold/step-deviance measures. The knowledge graph obtained is highly specific to the underlying domain of cyber law and ethics.

The Renyi entropy is a measure of uncertainty/randomness in a probability distribution. It is a generalization of the popular Shannon's entropy measure, offering control over parameter α that is the order of entropy. It is represented by Eq. 1.

$$H_\alpha(\mathrm{X}) = \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n} P(x_i)^\alpha\right) \tag{1}$$

Here n is the number of possible outcomes in the sample space X, x_i are the individual outcomes, and $P(x_i)$ is the probability of outcome x_i.

As the parameter α approaches 1, the Renyi entropy converges into Shannon entropy. A decision tree classifier is a supervised machine learning approach that makes predictions by constructing a tree-like structure. It starts at the root node with the entire dataset, while the algorithm traverses recursively, selecting features and setting thresholds to partition data. This process continues until leaf nodes are created where predictions are made based on the majority class for classification.

To build a decision tree, the algorithm selects features and thresholds at each node to create homogeneous subsets. The results, which are obtained through recursive splitting, form the leaves. This is a lightweight machine learning approach, and while decision trees are easily interpretable, they can sometimes be susceptible to certain drawbacks such as overfitting and lack of depth constraints.

Gated Recurrent Units represent a type of Recurrent Neural Network (RNN) architecture introduced in 2014 by Cho et al., specifically designed for effective modeling of sequential data, particularly in the domain of NLP. The inherent simplicity of the

gating mechanism in GRUs allows them to effectively capture and retain crucial information across extended sequences of data. The GRU cell comprises three consecutive steps: firstly, a reset gate is employed to selectively discard or forget portions of past information; secondly, both gates collaborate to generate a candidate activation for the current state; and lastly, an update gate regulates the combination of the previous hidden state and the candidate activation to produce new states. This design contributes to the GRU's notable efficiency in certain applications compared to LSTM networks, while maintaining competitive performance across various modeling tasks.

Hellinger's distance is a quantification of similarity between two given probability distributions. It compares the probability distribution functions P and Q as shown in Eq. 2.

$$H(P, Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_i \left(\sqrt{p_i} - \sqrt{q_i}\right)^2} \tag{2}$$

where p_i and q_i are the probabilities associated with the ith outcome in the sample space for P and Q. The Hellinger's distance ranges from 0 to $\frac{1}{\sqrt{2}}$ where 0 indicates perfect similarity between two distributions and $\frac{1}{\sqrt{2}}$ signifies complete dissimilarity.

The volume of intersection index is a similar overlap measure that ranges from 0 to 1 representing no overlap and completely identical, respectively.

4 Implementation, Performance Evaluation, and Results

The implementation of the proposed CKGLD framework is carried out with Python 3 and Google Colaboratory as the preferred IDE. The NLTK library is used to carry out NLP tasks. The deep learning GRUs are implemented using the deep learning configuration paradigm in Keras.

The experimentations were carried out on a single, large dataset that is an aggregation of four independent datasets that are: Einetic (2024). Cyber Ethics and Laws [Dataset], Einetic (2022). Tools and Methods Used in Cyber Crime [Dataset], Andien Septia Bundi Iffany (2022) Dataset Literature Review on Cyber law and Forensics [Dataset], and the Work With Data (2023) Cyber Law in Bangladesh [Dataset]. These datasets were strategically integrated by only obtaining the categories and terms from the dataset and crawling similar documents from Web 3.0 using a customized web crawler. A document and category set were created in an RDBMS format and linked to the documents. Reprioritization according to the categories was done, and customized annotators were used to generate annotations for the categories. In this manner, the final, single, large dataset was created.

The performance of the CKGLD framework which is the knowledge graph curation for linked open data generation focusing on cyber law and ethics is measured using precision, recall, accuracy, and F-measure as the primary metrics that evaluate

the relevance of the results while the FDR is an auxiliary metric that represents the proportion of false positives generated by the framework.

Table 1 indicates that the proposed CKGLD model has yielded the highest precision, recall, accuracy, and F-measure of 96.18%, 97.87%, 97.02%, and 97.02%, respectively, and the lowest FDR of 0.04. The comparisons are made with three other similar models namely the: KGC, KGCTO, and the RREKG. These models all underperform in comparison with the CKGLD model in all primary and auxiliary metrics.

The CKGLD outperforms all the other frameworks in consideration because it is driven by two distinct sets of knowledge namely the static and the dynamic knowledge. The dynamic knowledge is generated using data fountains as the tool of choice upon the extracted terms and categories of the dataset. Static metadata is obtained from several renowned, highly domain-specific repositories like e-books, glossaries, and e-query logs of cyber lawyers and domain experts along with metadata that is obtained from the highly domain-specific web. The metadata is obtained through highly focused crawlers. This makes the static knowledge quite rich in terms of its domain. Apart from this, the model uses two distinct classifiers: the decision tree classifier to classify the dataset and the generative adversarial networks to classify the dynamic knowledge generated using the data fountains. The decision tree classifier is a machine learning, lightweight classifier, while the GANs are a strong deep learning classifier. This gives the framework a good balance of the complexities of the learning strategies in the model. The organization of the classified dataset terms into a knowledge map using Co-SIM rank and the clustering by the DBSCAN to overlap the dynamic and the static knowledge generated to yield clustered relevant entities which are further organized into a knowledge graph through Hellinger's distance and VOI index provides very strong relevance computation mechanisms and semantics-oriented qualitative reasoning mechanisms into the model, making it the best performing framework in knowledge graph curation for linked open data for cyber law and ethics.

The KGC is a practical framework for curating knowledge graphs where the data is targeted using personal assistance question–answering systems, search engines, and the dataset. The preprocessing of data is the main insight provided by the framework. The lack of human cognition and complete automation of relevant computation tasks

Table 1 Performance of CKGLD against other frameworks

Model	Average precision (%)	Average recall (%)	Average accuracy (%)	Average F-measure (%)	FDR
KGC [1]	90.11	91.43	90.77	90.76	0.10
RREKG [2]	90.27	92.41	91.34	91.33	0.10
KGCTO [3]	92.09	93.36	92.725	92.72	0.08
Proposed CKGLD	96.18	97.87	97.025	97.02	0.04

and auxiliary knowledge segregation and organization at every step is the downfall of the KGC model.

The KGCTO model generates knowledge graphs from text ontologies which is a strong source of knowledge. This provides a strong structural and systematic feed of knowledge into the framework. Although the KGCTO uses machine learning and semantic inferencing along with strong knowledge integration, it has poor relevance computation mechanisms putting it at a disadvantage to the CKGLD.

The RREKG model uses an intermediate language called the biological expression language where syntactic and semantic quality is assessed through structured entities. The NeuroMMSig Database yields the knowledge contents which were in turn obtained from MEDLINE abstracts and PubMed Central articles. Text mining is done using the INDRA framework. The RREKG, however, lacks human cognition and also in expertise reasoning and learning mechanisms through semantic associations and semantic similarity measurements. Further, it only obtains knowledge from two fixed sources and not several heterogeneous sources making it lag when compared to the CKGLD.

Precision versus recommendations distribution curve of the CKGLD is depicted in Fig. 2. It is observed that the CKGLD shows the highest performance among all the frameworks considered. This is due to strong knowledge integration, good learning algorithms, and semantic similarity measures being used.

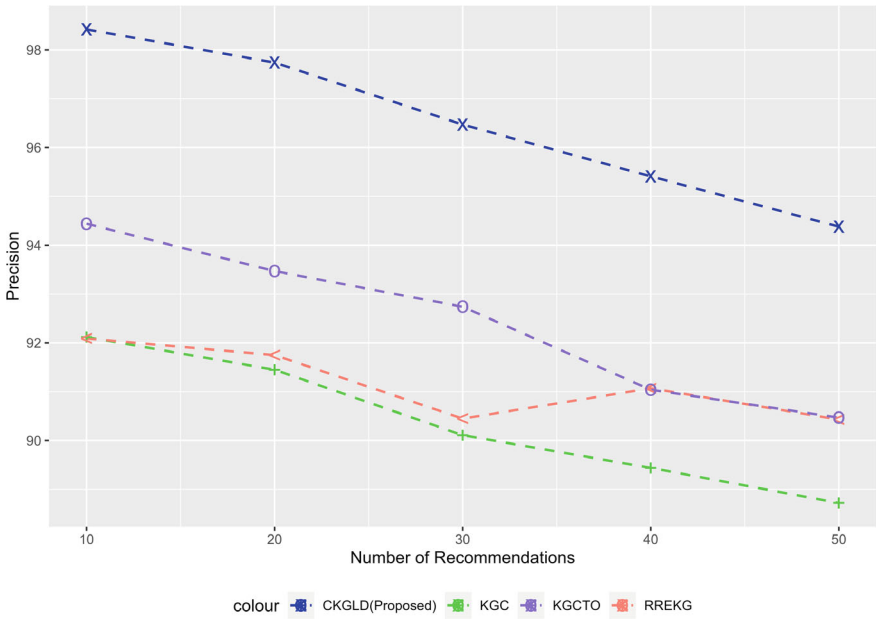


Fig. 2 Precision versus No. of recommendations

5 Conclusions

The CKGLD is a framework that performs the task of knowledge graph curation for cyber law and ethics. The framework incorporates static metadata and dynamic knowledge from Web 3.0, addressing issues of scale and specificity. The decision tree classifier being lightweight helps maintain a low computational load while ensuring domain relevance. GANs and DBSCAN clustering help in obtaining a full understanding of the dataset. This paired with the multi-agent paradigm enhances computational efficiency, making the framework suitable for Web 3.0. Overall, the CKGLD has been able to produce the highest precision of 96.18% and lowest FDR of 0.04.

References

1. Huaman E, Fensel D (2021) Knowledge graph curation: a practical framework. In: Proceedings of the 10th International joint conference on knowledge graphs, pp 166–171
2. Lakshmi VR, Deepak G, Santhanavijayan A, Radha S (2022) Knowledge graph curation from text via ontologies. In: 2022 International conference on augmented intelligence and sustainable systems (ICAISS). IEEE, pp 726–732
3. Hoyt CT, Domingo-Fernández D, Aldisi R, Xu L, Kolpeja K, Spalek S, Wollert E, Bachman J, Gyori GM, Greene P, Hofmann-Apitius M (2019) Re-curation and rational enrichment of knowledge graphs in biological expression language. Database 2019:baz068
4. Iglesias E, Jozashoori S, Vidal ME (2023) Scaling up knowledge graph creation to large and heterogeneous data sources. J Web Semantics 75:100755
5. Ma X (2022) Knowledge graph construction and application in geosciences: a review. Comput Geosci 161:105082
6. Kumar KM, Reddy ARM (2016) A fast DBSCAN clustering algorithm by accelerating neighbor searching using groups method. Pattern Recogn 58:39–48
7. Qaiser S, Ali R (2018) Text mining: use of TF-IDF to examine the relevance of words to documents. Int J Comput Appl 181(1):25–29
8. Daradkeh M, Mansoor W, Atalla S, Himeur Y, Kerdjidj O (2023) Lifelong machine learning for topic modeling based on Hellinger distance. In: 2023 International joint conference on neural networks (IJCNN). IEEE, pp 1–8
9. Wu Y, Chen Y, Wang L, Ye Y, Liu Z, Guo Y, ... Fu Y (2018) Incremental classifier learning with generative adversarial networks. arXiv preprint arXiv:1802.00853
10. Oggier F, Datta A (2021) Renyi entropy driven hierarchical graph clustering. PeerJ Comput Sci 7:e366

Neuro-ocular Dynamics in Stress Detection: An Integrated Analysis of EEG and EOG Modalities

Abhijit Das, Akanksha Subudhi, Priyanka Jain, Rahee Walambe, Ketan Kotecha, N. K. Jain, Rahul Neiwal, and Manoj Jain

Abstract This research investigates the dynamic interrelation between neurological and ocular markers in response to stress, employing an integrated analysis of electroencephalogram (EEG) and electrooculogram (EOG) modalities. Utilizing 28-channel EEG gel-based electrodes of brain product, we extracted frequency band information, Hjorth parameter, spectral entropy, and statistical parameters, while 2-channel vertical EOG data provided insights into blink rate and the number of blinks. The late fusion of these modalities was implemented for machine learning algorithms, complemented by unimodal prediction strategies. This study reveals findings that are intricate stress-related patterns at the neuro-ocular interface, offering a comprehensive exploration of stress dynamics. Late fusion techniques demonstrated enhanced predictive capabilities, enriching our understanding of stress responses. By elucidating the interconnected roles of the brain and eyes, this study contributes novel perspectives to stress assessment methodologies, potentially shaping future research in the field.

Keywords EEG · EOG · Multi-model · Stress detection · ICA · Frequency bands

A. Das · A. Subudhi (✉) · P. Jain · N. K. Jain
C-DAC, Delhi, India
e-mail: akankshasubudhi09@gmail.com

A. Das
e-mail: dabhijit@cdac.in

P. Jain
e-mail: priyankaj@cdac.in

R. Walambe · K. Kotecha
SIT Pune, Mulshi, India

R. Neiwal · M. Jain
MeitY, Delhi, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_32

1 Introduction

Stress, a ubiquitous aspect of human experience, exerts a profound impact on physiological and cognitive functions. Stress detection is important across various domains, impacting health, performance, and overall well-being. Early identification of stress allows for timely interventions, contributing to improved mental health outcomes, optimized performance, and enhanced safety in critical environments [1, 2]. In this context, electroencephalogram (EEG) and electrooculogram (EOG) play pivotal roles in stress detection. EEG, measuring brain electrical activity, offers insights into neural correlates of stress through the observation of specific patterns such as increased theta and alpha activity. The temporal dynamics captured by EEG enable the examination of how stress unfolds over time, shedding light on cognitive processing changes associated with stress [1, 3]. On the other hand, EOG, which monitors eye movements, provides valuable information about ocular responses influenced by stress. Analysis of blink characteristics and gaze direction, as assessed by EOG, can offer indicators of stress levels. Integrating EEG and EOG in stress detection allows for a comprehensive understanding of both neural and physiological responses to stress. This holistic approach facilitates not only the identification of stress but also the development of effective strategies for stress management, biofeedback, and self-regulation [4, 5]. Machine (ML) and deep learning (DL) play a transformative role that goes beyond traditional approaches. Instead of simply processing data, these technologies reshape our methods, providing a more insightful and adaptable way to interpret neural signals for detection purposes. ML serves as a perceptive guide, navigating the intricate to discern meaningful insights. DL, on the other hand, ventures into the profound layers of neural complexity, delving into the intricacies of signal patterns and deciphering representations that traditional methods may overlook. The role of DL lies in enhancing the precision of detection through its capacity to unearth latent features [6–8]. Understanding the dynamics of stress responses necessitates a comprehensive exploration of both neurological and ocular indicators. This study focuses on the integrated analysis of EEG and EOG modalities to unravel the nuanced interplay between brain and ocular activity during stress.

This experiment holds significance not only in refining stress evaluation methodologies but also in advancing the understanding of the holistic nature of stress responses. By bridging the neuro-ocular gap, this study aims to contribute unique insights into the complex patterns and correlations between brain and ocular markers during stress, ultimately enhancing our ability to comprehend and address stress-related challenges.

2 Related Work

Intensive research endeavors have been dedicated to the classification and diagnosis of stress detection; however, significant opportunities remain for further enhancement. While various biosignals are actively employed in this field, employing a combination of these multimodal approaches holds promise for achieving improved detection outcomes. Despite some ongoing research in this direction, there is still a noticeable gap in understanding the full potential of integrating multiple modalities for enhanced stress detection. This section aims to address and delve into this specific research gap. Reference [9] introduces a novel approach, combining EEG signals with non-scalp electrodes to classify the P300 ERP. By incorporating non-scalp electrode signals, particularly during physical activity on a stationary bike, the proposed model demonstrates significantly improved performance. The publication [10] introduces a real-time DL framework designed to integrate ECG, voice, and facial expressions for the detection of acute stress. The study employs the Montreal Imaging Stress Task (MIST) to evaluate the framework's effectiveness. Through a multimodal approach, the paper showcases enhanced accuracy in discerning acute stress. In the work [11], the scientists created a multimodal emotion database by simultaneously recording participant EEG, voice, photoplethysmography, and facial pictures while eliciting neutral, happy, sad, and furious feelings. They concluded that verbal signals are less accurate in identifying emotions than EEG signals. The paper suggests that fusion procedures that mix voice and EEG inputs increase overall accuracy of emotion recognition compared to solo speech or EEG. Several modalities were included in the experiment [2] to characterize stress by taking advantage of the complementarity of multimodal inputs. They have researched the viability of combining physiological signs and facial features to estimate a person's stress level. The findings of the experiment showed that combining face features with remote pulse rate variability provided high performance and increased accuracy. The goal of the paper [12] was to combine various EEG data sources under three different auditory stimulation conditions: neutral, negative, and positive. The study was distinguished between normal controls and depressed patients. A feature-level fusion technique was utilized to combine data from many modalities and create a model for recognizing depression. Feature weighting was done using genetic algorithms. Compared to the individual modality schemes, the study showed that the fusion modality may reach a greater accuracy rate for depression identification.

3 Methodology and Implementation

3.1 Data Acquisition

A 'Brain Product' actiCAP slim/snap was employed for data collection, utilizing gel-based active electrodes to ensure top-notch recordings. The data was captured from 32 channels, with 28 channels specifically allocated for EEG ['Fp1', 'Fz', 'F3', 'F7', 'FT9', 'FC5', 'FC1', 'C3', 'CP5', 'CP1', 'Pz', 'P3', 'P7', 'O1', 'Oz', 'O2', 'P4', 'P8', 'CP6', 'CP2', 'Cz', 'C4', 'FT10', 'FC6', 'FC2', 'F4', 'F8', 'Fp2'] and 2 channels designated for EOG [EOGR, EOGL] signal acquisition. Brain vision served as the tool for recording and analysis of the data, with EEG and EOG recorded at a sampling frequency of 250 Hz. The data acquired would be made available upon request at dataecho.in.

3.2 Subject

Thirty-three individuals in good health participated in this experiment. From the initial cohort of 33 subjects, 28 were deemed suitable for further analysis, while the data from the remaining subjects were excluded due to issues with the recordings, rendering them faulty. The participants comprised of 14 females and 14 males. All participants fell within the age range of 18–45 years. Pregnant women and individuals with any health or psychological conditions were not included in the experiment. Additionally, volunteers were given instructions to abstain from caffeine, alcohol, and smoking for 3 h before sample collection, as these factors could potentially introduce confounding variables.

While multimodal data offers valuable insights for stress detection, ethical considerations along with privacy concerns are paramount. This study emphasized transparency by thoroughly educating all participants on the data collection process and obtaining their voluntary written consent, ensuring their informed participation.

3.3 Paradigm Description

Although there are diverse stress-inducing factors, our experiment specifically aimed to investigate psychological stress and stress arising from cognitive load. To induce stress, we employed mathematical tasks, Stroop tests, and thriller/horror videos for each participant. The math task involved solving various mental math questions without the aid of pen and paper, creating a cognitive stressor. The Stroop test, a required participants to quickly name the color of ink in which color words were printed, ignoring the actual word. Thriller/horror videos were chosen to heighten stress levels through suspenseful elements. Before engaging in these tasks, the subject

is initially relaxed through a sequence of calming videos leading to the meditation phase to reach the relaxed state of mind. These initial steps acted as non-stressors before the stress-inducing tasks began. The participants were given a 3-min time frame, with a set timer, for each of the above stress and relaxing events. The State-Trait Anxiety Inventory (STAI) is a commonly used measure of trait and state anxiety. It can be used in clinical settings to diagnose anxiety, where anxiety is a significant component of the stress response. It also is often used in research as an indicator of distress. A baseline control feedback of the participant's current state of mind is taken before starting the experiment. The STAI survey is being conducted three times: at the start, the conclusion of the non-stressor/calm phase, and the conclusion of the stressor phase. This is being done to establish a baseline for stress detection and enable a comparison for assessment purposes.

The experiment spanned a total duration of 23 min, encompassing 9 min allocated to stressor tasks and an additional 9 min dedicated to non-stressor segments along with the other segments like the welcome screen, STAI survey, and neutral images.

3.4 EEG Signal

Preprocessing: The first step in preprocessing involves identifying bad channels, which are electrodes yielding poor-quality signals. These bad channels were then interpolated, thereby correcting for lost or noisy data. This step helps to minimize the influence of common noise across the EEG recording. To further refine the data, a bandpass filter ranging from 1 to 48 Hz and a notch filter designed to eliminate electrical noise at 50 Hz were applied. These filters are essential for removing high-frequency noise, slow drifts, and isolating the relevant frequency bands for cognitive process analysis. Independent component analysis (ICA) was used to separate mixed signals into their independent components, which is particularly important in EEG for artifact rejection. Components associated with eye blinks, eye movements, and other non-brain activities were then removed for smooth analysis [12, 13]. This comprehensive preprocessing as depicted in Fig. 1 ensured that the EEG data were as clean and reliable as possible, providing a solid foundation for accurately interpreting the neural correlates of stress.

Features: The frontal region is commonly associated as a potential marker for stress detection. To prove the hypothesis, a repeated measures analysis of variance (ANOVA) test was performed on EEG channel data.

ANOVA Test 1: On frontal EEG channels (Fp1, Fp2, Fz, F3, F4, F7, F8) and output:

Source	ddof1	ddof2	F	p-unc	ng2	eps
Event	6	180	5.641509	0.000022	0.043378	0.306

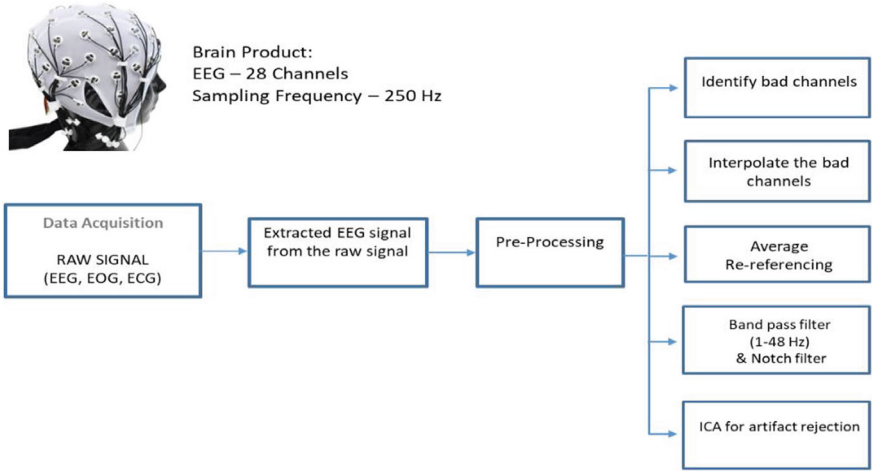


Fig. 1 Various stages of EEG signal preprocessing

The alternate hypothesis was accepted, i.e., 'there are significant differences in the alpha power of subjects across the different conditions'.

ANOVA Test 2: All 28 EEG channels with the result:

Source	ddof1	ddof2	F	p-unc	ng2	eps
Event	6	48	1.682705	0.145707	0.031356	0.294809

Therefore, as the P value is 0.14, i.e., $p > 0.05$, so the null hypothesis is not rejected, i.e., 'there are no significant differences in the alpha power of subjects across the different conditions'. Finally, there are differences between the stress and relaxation conditions when focusing on the frontal regions as opposed to considering all channels.

Frequency Bands: EEG signals are composed of several frequency bands, each associated with different types of brain activity. Common bands include Delta (1–4 Hz), Theta (4–8 Hz), Alpha (8–13 Hz), Beta (13–30 Hz), and Gamma (>30 Hz). This study employed the short-time Fourier transform (STFT) to extract frequency bands from the signal to analyze temporal variations in brain activity. Applying STFT to the data was useful to isolate specific frequency bands of interest—Delta, Theta, Alpha, Beta, and Gamma. The decomposition provided insights into tracking how the energy distribution across these bands varied throughout the EEG recording, providing insights into the brain's dynamic responses under stress conditions.

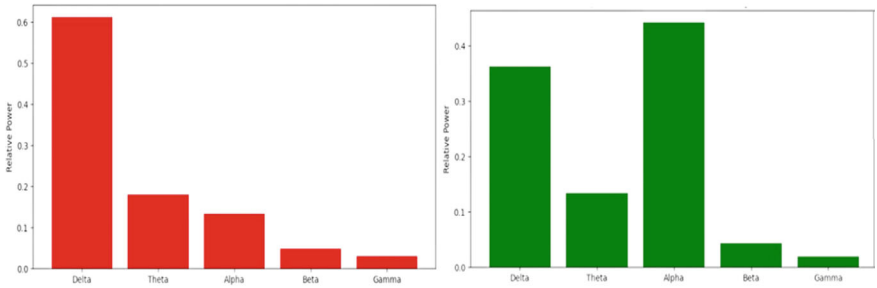


Fig. 2 **a** Alpha power band under stress and **b** alpha power during relax state

Typically, alpha waves are associated with a relaxed, calm, and resting state of the brain as shown in Fig. 2. The lower power in the alpha band under stress compared to relaxation aligns with our common findings that stress reduces alpha wave activity, indicating heightened alertness or anxiety.

Statistical Feature Extraction: Statistical features are calculated to quantify the characteristics of the EEG data. These features can include measures such as mean, median, variance, standard deviation, skewness, and kurtosis. These statistical parameters serve as input features for machine learning models because they capture essential properties of the signals that may correlate with cognitive states.

Entropy and Hjorth Parameters: Entropy is a statistical measure of randomness used to characterize the complexity of signal patterns. Higher entropy values indicate more complex, less predictable patterns, which can be indicative of certain brain states or responses. A higher spectral entropy suggests a more chaotic and potentially stressed brain state, while a lower value indicates a more ordered state, potentially reflecting relaxation. A higher spectral entropy suggests a more chaotic and potentially stressed brain state, while a lower value indicates a more ordered state, potentially reflecting relaxation. The combination of frequency band extraction, statistical features, entropy, and Hjorth parameters provides a comprehensive set of features that capture both the frequency and time-domain characteristics of EEG signals [14, 15].

3.5 EOG Signal

In stress detection, EOG provides a non-intrusive means of gauging the physiological responses linked to stressors. The variations in eye movement patterns and blinking rates, as detected by EOG, offer valuable insights into the individual's cognitive and emotional state during stress-inducing tasks.

The incorporation of EOG in stress detection not only enhances the understanding of how individuals visually process stressors but also offers a potential avenue for

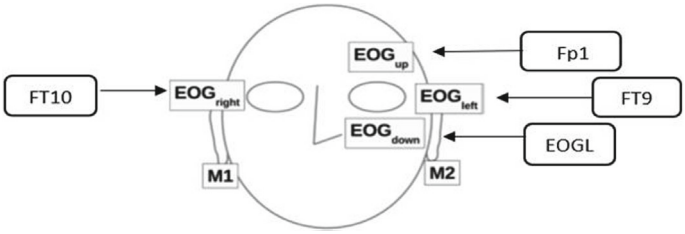


Fig. 3 Channels position vertical [Fp1—EOGL] and horizontal [FT10—FT9] EOG

developing objective and real-time stress monitoring systems [1, 4]. For the analytical process, vertical EOG (EOGv) is being extracted for further preprocessing and extracting various ocular features**.** Figure 3 depicts the EOGv with respect to the experimental setup presented in this paper. While EOG provides valuable information, it is often used in conjunction with other physiological measures and behavioral assessments to form a comprehensive understanding of cognitive and emotional processes.

Preprocessing: From the initial cohort of 33 subjects, 28 participants were deemed suitable for further analysis, while the data from the remaining subjects were excluded due to issues with the recordings, rendering them faulty. The channels required for the calculation of EOGv (Fp1and EOGL) were extracted from the raw signals. Recognizing that the raw signals inherently harbor a considerable amount of noise, a critical preprocessing step was undertaken on these channels. This preprocessing stage as shown in Fig. 4. is imperative before computing EOGv, as it serves to mitigate the impact of noise on the analysis. By addressing noise concerns at this stage, the aim is to prevent inaccuracies in the subsequent analysis, ensuring the reliability and validity of our findings. To enhance the clarity of our recorded signals, a 50 Hz Notch filter was implemented to effectively eliminate power-line interference noise. Additionally, a bandpass filter spanning from 1 to 48 Hz was applied to the individual channels, i.e., EOGL and Fp1. This specific bandpass filter serves the purpose of isolating the frequency bands of interest within this range while effectively excluding any inappropriate noise. By this approach, the focus is targeted toward the analysis on the relevant frequency components. This meticulous filtering process enhances the overall quality of our data and facilitates a more robust exploration of the targeted physiological signals. After performing preprocessing on individual channel level, the EOGv is computed which is further processed to ensure maximum signal-to-noise ratio.

To address any potential baseline drift in the EOGv data, a high-pass filter was applied with a cutoff frequency of 0.1, using a Butterworth filter. Its purpose was to effectively eliminate any slow, gradual variations in the signal that might have been introduced during the recording process.

The aim was to ensure that our subsequent analyses were not affected by undesirable shifts in the baseline of the EOGv data. Furthermore, to refine the frequency characteristics of the EOGv signals, a fourth-order Butterworth bandpass filter was

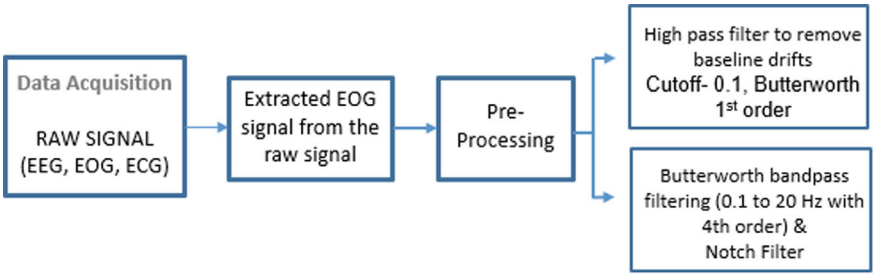


Fig. 4 Various stages of EOG signal preprocessing

systematically applied. This filter was designed to selectively retain frequencies within the range of 0.1–20 Hz, effectively isolating the relevant components of the EOGv signal. These intricate filtering procedures were integral in minimizing unwanted artifacts and noise in the EOG data, ensuring that the subsequent analyses and interpretations are founded on clean and accurate signals.

Features: Exploring the intricacies of the analysis, distinctive features were delved into characterizing eye movements and employed a thorough approach to quantify and understand these phenomena. Each of the features were computed over fixed-length epochs of 10 s. This strategic segmentation allowed for a detailed examination of the temporal dynamics within the data, ensuring that our analyses were not only comprehensive but also capable of capturing subtle variations over specific time intervals. Within the domain of blink characteristics, the spotlight was directed toward two metrics: EOG Blink Peaks and EOG Mean Rate (blink) [16, 17]. Figure 5 depicts detection of single blink and blink rate of a subject. The blink peaks took to identify and quantify the prominent peaks associated with eye blinks, offering a detailed examination of the dynamics of these ocular movements. Meanwhile, Mean Rate (blink) provided a comprehensive measure of the average rate at which blinks occurred, offering insights into the overall blink activity during the recorded epochs.

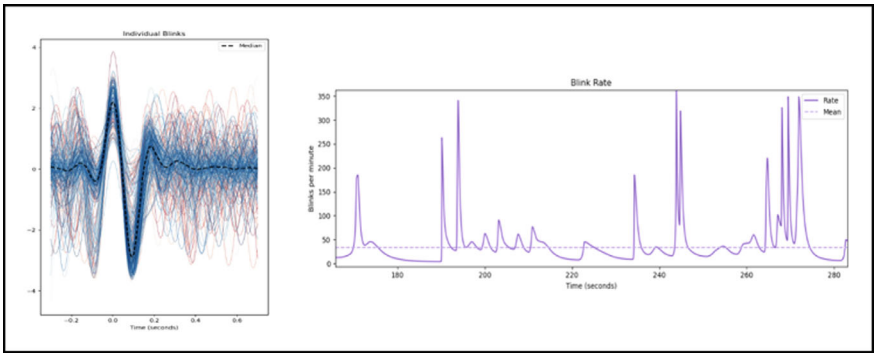


Fig. 5 Individual blinks detected and the blink rate of a particular subject in the experiment

Broadening our range, statistical features were also investigated to gain a deeper insight into the fundamental patterns present in the data. We systematically calculated basic statistical values like the mean, standard deviation, and kurtosis. These calculations not only highlighted the average and spread of the data but also provided information about its overall shape and distribution. This approach enhanced our ability to interpret the observed physiological signals in a more detailed and minute manner.

A total of 5 features were extracted from EOG data for further predictions. This meticulous focus on feature extraction and computation serves as the cornerstone of our analytical framework, enabling a thorough exploration of the complexities embedded in the recorded physiological data.

4 Results and Discussion

In the context of this experiment, the utilization of ML techniques takes center stage for the identification and detection of stress onset in participants. While ML models exhibit commendable proficiency in recognizing patterns for predictive purposes, their efficacy hinges significantly on the nature and quality of the features employed in the prediction process. In essence, the predictive prowess of these models relies heavily on the selection and relevance of features extracted from the data. The relationship between the ML algorithms and the features they operate upon becomes a crucial determinant in the accuracy and reliability of stress identification. This underscores the importance of meticulous feature selection and extraction. In essence, the success of the stress detection endeavor is intricately linked to the thoughtful curation of features that serve as the foundation for the ML models' predictive capabilities [18, 19]. This study has utilized different kinds of machine learning models: Logistic Regression, XGBoost, Random Forest, SVM, and KNN. They were employed for classification and stress detection, as outlined in the accompanying Table 1. Notably, it became apparent that both the Random Forest and XGBoost algorithms exhibited significant overfitting tendencies. To ensure the robustness and generalizability of the models, they were excluded from subsequent analyses. KNN emerged as the top performer in both EEG and EOG classifications among the remaining three models. This observation underscores the efficacy of the KNN algorithm in the context of stress detection and signifies its potential suitability for future applications within this domain.

Despite initially demonstrating commendable accuracy, the models faced challenges in accurately classifying stress on data that hadn't been used during the training and testing phases [20]. To assess their real-world efficacy, only data from 25 out of the 28 participants were employed for training the models, leaving the remaining 3 participants' data exclusively for inferencing purposes. Considering the inter-subject variability, a significant phenomenon in neural signal processing becomes imperative as values may vary among subjects, exerting a considerable influence on the training of models. While partitioning the data into training and testing sets, the

Table 1 Uni and multimodal accuracies (stress) for every ML model

Accuracy (%)	LR		XGB		RF		SVM		KNN	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
EEG	75.7	77.6	90.1	83.3	100	84.8	77.3	77.3	83.1	84.0
EOG	66.6	69.4	98.2	77.6	100	75.3	72.3	72.9	75.3	71.1
Late fusion	71.1	73.5	94.2	80.4	100	80	74.8	75.1	79.2	77.5

models employ a random assignment approach, introducing a substantial bias due to multiple feature rows corresponding to a single subject. To mitigate this bias, deliberate efforts were made to ensure that feature rows from each subject were proportionally divided into a 70–30 (train-test) ratio, creating a set devoid of bias. Despite a reduction in accuracy percentage, this adjustment notably enhanced the stress inference capabilities of the model on previously unseen data.

After each modality has undergone independent analysis, late fusion with averaging is a useful method for combining data from various sources in multimodal stress analysis. This method improves generalization over a variety of datasets, retains flexibility across modalities, strengthens model robustness, and preserves information richness [9, 12, 21]. By making use of the complimentary nature of multimodal information, late fusion with averaging emerges as a potent technique for creating precise and flexible stress detection models.

5 Conclusion

A comprehensive study demonstrates the potential of combining EEG and EOG data for enhanced stress detection. Utilizing advanced machine learning techniques like KNN, Logistic Regression, and SVM, the study effectively analyzes neuro-ocular dynamics, providing insightful correlations between brain and eye responses under stress. The findings, emphasizing a late fusion approach, offer a novel perspective in stress assessment, suggesting a more accurate method for detecting stress. This integrated analysis not only advances our understanding of stress responses but also opens new avenues for future research in the field.

References

1. Katmah R, Al-Shargie F, Tariq U, Babiloni F, Al-Mughairbi F, Al-Nashash H (2021) A review on mental stress assessment methods using EEG signals. Sensors 21(15):5043
2. Ouzar Y, Lagha L, Bousefsaf F, Maaoui C (2022) Multimodal stress state detection from facial videos using physiological signals and facial features. In: International conference on pattern recognition. Springer Nature Switzerland, Cham, pp 139–150

3. Altaheri H, Muhammad G, Alsulaiman M, Amin SU, Altuwaijri GA, Abdul W, Bencherif MA, Faisal M (2023) Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: a review. Neural Comput Appl 35(20):14681–14722
4. Belkhiria C, Boudir A, Hurter C, Peysakhovich V (2022) EOG-based human-computer interface: 2000–2020 review. Sensors 22(13):4914
5. Zhu Y, Li Y, Lu J, Li P (2020) A hybrid BCI based on SSVEP and EOG for robotic arm control. Front Neurorobot 14:583641
6. Subudhi A, Hasan N, Nene MJ (2023) DeEN: deep ensemble framework for neuroatypicality classification. In: 2023 3rd International conference on intelligent technologies (CONIT). IEEE, pp 1–6
7. Kushwaha S, Jain P, Jain NK (2023) Deep learning for visual perceptual brain decoding as image classification. In Shinde SV, Bendre V, Hemanth DJ, Balafar MA (eds) Applied artificial intelligence: a biomedical perspective, 1st edn. CRC Press, p 26. https://doi.org/10.1201/9781003324430
8. Jain P, Mishra RK (2024) Xplainable AI for deep learning model on PCOD analysis
9. Ding Y, Huynh B, Xu A, Bullock T, Cecotti H, Turk M, Giesbrecht B, Höllerer T (2019) Multimodal classification of EEG during physical activity. In: 2019 International conference on multimodal interaction, pp 185–194
10. Zhang J, Yin H, Zhang J, Yang G, Qin J, He L (2022) Real-time mental stress detection using multimodality expressions with a deep learning framework. Front Neurosci 16:947168
11. Wang Q, Wang M, Yang Y, Zhang X (2022) Multi-modal emotion recognition using EEG and speech signals. Comput Biol Med 149:105907
12. Cai H, Qu Z, Li Z, Zhang Y, Hu X, Hu B (2020) Feature-level fusion approaches based on multimodal EEG data for depression recognition. Inf Fusion 59:127–138
13. Khosla A, Khandnor P, Chand T (2020) A comparative analysis of signal processing and classification methods for different applications based on EEG signals. Biocybern Biomed Eng 40(2):649–690
14. Liu L, Ji Y, Gao Y, Ping Z, Kuang L, Li T, Xu W (2021) A novel fatigue driving state recognition and warning method based on EEG and EOG signals. J Healthc Eng 2021
15. Nagabushanam P, Thomas George S, Radha S (2020) EEG signal classification using LSTM and improved neural network algorithms. Soft Comput 24:9981–10003
16. Das D, Chowdhury MH, Chowdhury A, Hasan K, Hossain QD, Cheung RC (2023) Application specific reconfigurable processor for eyeblink detection from dual-channel EOG signal. J Low Power Electron Appl 13(4):61
17. Wang M, Cui X, Wang T, Jiang T, Gao F, Cao J (2023) Eye blink artifact detection based on multi-dimensional EEG feature fusion and optimization. Biomed Signal Process Control 83:104657
18. Raghavendra U, Acharya UR, Adeli H (2020) Artificial intelligence techniques for automated diagnosis of neurological disorders. Eur Neurol 82(1–3):41–64
19. Zhang J, Yin Z, Chen P, Nichele S (2020) Emotion recognition using multi-modal data and machine learning techniques: a tutorial and review. Inf Fusion 59:103–126
20. Jain P, Sharma G, Kirode P, Koria H, Deep A, Jain NK (2023) EEG-based real-time prediction of cognitive state on smartphone. In: International conference on emerging trends and technologies on intelligent systems. Springer Nature Singapore, Singapore, pp 147–159
21. Sun J, Cao R, Zhou M, Hussain W, Wang B, Xue J, Xiang J (2021) A hybrid deep neural network for classification of schizophrenia using EEG Data. Sci Rep 11(1):4706

Ransomware Detection in Healthcare Organizations Using Supervised Learning Models: Random Forest Technique

Rimpa Dutta and Sandip Karmakar

Abstract Ransomware attacks pose a formidable challenge to healthcare organizations, disrupting critical operations and demanding ransom for compromised server access. In response, the paper introduces an innovative model focused on static analysis for efficient detection and proactive prevention of such attacks. The approach diverges from the traditional reliance on time-consuming dynamic analysis, offering a streamlined yet robust defense mechanism. AIIMS Delhi's firsthand encounter with daily disruptions and ransom demands underscores the urgency for effective solutions. Our model not only identifies ransomware attacks but also actively works to impede and thwart illegal activities, contributing to heightened cyber security resilience in the healthcare sector. The research represents a crucial step toward fortifying healthcare systems against the evolving landscape of cyber threats. The unique aspect of the proposed method lies in it's avoidance of the disassembly step. This innovation is achieved by directly extracting features from raw byte data using pattern extraction through regular occurrence, thus eliminating the need for disassembly. The approach simplifies the process and also results in a remarkable acceleration of the detection speed. The examination employed a Random Forest classifier to methodically assess how altering the quantities of trees and seeds affects the efficiency of ransomware detection. The results indicated that the optimal blend for achieving a balance between time efficiency and accuracy was found with 100 trees using a seed number 1. Through experimental evaluation, it was evident that the proposed method excelled, achieving a robust detection accuracy of 98.32% in identifying ransomware.

Keywords Ransomware · Cyber security · Random forest · Attacks

R. Dutta (✉) · S. Karmakar
Department of Computer Science and Engineering, National Institute of Technology Durgapur, Durgapur, West Bengal 7000139, India
e-mail: rimpa.cseeng@gmail.com

S. Karmakar
e-mail: skarmakar.cse@nitdgp.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_33

1 Introduction

In the present scenario, malicious actors are utilizing advanced techniques to develop novel and lucrative forms of malware. Among these, ransomware has emerged as a particularly widespread and concerning threat in recent times. Ransomware poses an unparalleled challenge in cyber security, given it's irreversible nature and the difficulties associated with prevention and containment [1] and highlighted the deficiencies of current approaches [1] in efficiently identifying and thwarting these attacks. The methodology involves extracting features directly from the raw byte data of files [1], providing a novel perspective on strengthening cyber security through advanced classification methods. In 2016, Osterman Research and Inc. combined forces, pooling their resources to confront this growing menace. The collaboration during that pivotal year underscored a shared commitment to understanding and addressing the complexities of ransomware. Together, they worked toward developing comprehensive strategies for prevention and mitigation, emphasizing involving approximately 290 organizations across diverse industries in Europe. Simultaneously, conducted a similar analysis for organizations in the United States, gaining valuable insights into their operations and practices [2]. The study findings unveiled that half of these entities had succumbed to ransomware incidents within the course of that year [2]. Approximately 40 percentile of the victims found themselves compelled to comply with the demands of attackers. In February 2017, a statistical report sourced from VirusTotal [2] highlighted the submission of nearly 1.37 million novel [3] cyber threat samples. This substantial figure underscored the persistent and evolving nature of cyber threats during that period [3]. Distinguishing between malware and ransomware hinges on temporal and behavioral nuances. Unlike malware, which covertly infiltrates and damages systems without extorting a ransom [4], ransomware explicitly demands payment. Wecksten et al. [5] highlighted the ongoing difficulties in preventing such attacks. Similarly, Kharaz et al. [6] reiterated the persistent challenges associated with thwarting these security threats. As a result, there is a pressing demand for creative methods capable of identifying ransomware effectively [7]. Investigating the use of artificial intelligence algorithms for ransomware classification [7], a Random Forest model is utilized. Through experimental iterations with varied seed and tree sizes, an optimal Random Forest classifier was crafted to ensure precise and reliable ransomware detection. Ransomware attacks in healthcare organizations pose critical challenges, notably impacting patient safety, data security, and operational integrity. The disruption of access to vital patient information can lead to delays in treatments and compromised care delivery. The theft of sensitive patient data during such attacks not only raises privacy concerns but also puts healthcare providers at risk of regulatory non-compliance, resulting in potential financial penalties. Beyond immediate financial losses, the erosion of patient trust and reputational damage can have long-term consequences, affecting the organization's relationships with patients and partners.

Moreover, the operational disruption caused by ransomware can paralyze day-to-day functions, impacting appointment scheduling, medical records access, and

the functionality of critical medical equipment. As healthcare organizations grapple with these challenges, robust cyber security measures and strategic planning are imperative to safeguard patient well-being and maintain the integrity of healthcare services.

2 Literature Review

In [8], machine learning algorithms were applied to a dynamic feature dataset to detect ransomware. Gradient Regression Trees, Random Forests, and Neural Networks were evaluated using a tenfold cross-validation technique. The average accuracy surpassed 99%. In [9], ransomware detection often relies on the utilization of machine learning algorithms and frameworks. These tools are frequently utilized to extract dynamic attributes associated with ransomware behavior. In the paper [10], study provides a thorough evaluation of ransomware detection approaches, encompassing a wide range of methodologies, including machine learning, deep learning, or a combination of both. The paper [11], conducts an in-depth analysis of how machine learning plays a crucial role in the development of automated detection tools, examining multiple dimensions and effectively tackling the challenges associated with machine learning-based models. Through comprehensive exploration, it sheds light on the intricate dynamics involved in leveraging machine learning for enhanced detection capabilities in various domains. In paper [12], outlines static analysis, a method examining file attributes like size and opcode frequency without execution, proposing a machine learning integrated approach for ransomware detection based on file entropy and opcode frequency. Results indicate a robust 97.4% detection rate achieved solely on a dataset of ransomware files.

3 The Proposed Method

In this proposed model, our focus lies in the detection of ransomware attacks through the implementation of a static method. Through meticulous evaluation, it has been observed that the utilization of the Random Forest machine learning model yields significantly enhanced accuracy when compared to alternative machine learning approaches. In the concluding step, a normalization procedure is implemented to standardize the impact of all prevalent patterns, thereby stabilizing variance. The described process, as expressed in [2], involves allotting uniform weights to all frequent patterns. The normalized frequency (N, f) is calculated by dividing the frequency of specific features (ni, j) by the total number of features within a given file (k, nk, j). In simpler terms, it means that each frequent pattern is given an equal weight, determined by the ratio of the frequency of a particular feature to the total number of features in the file [2]. Moving on to the next stage, the Gain Ratio (GR) has been chosen as a fundamental technique for feature selection [2]. Feature selection is

fundamentally about diminishing the dimensionality of features by excluding irrelevant ones, thereby preserving only the most essential variables for the present predictive model as well as the predictive process of a Random Forest, the final decision is determined by combining forecasts from numerous decision trees using a majority voting approach [2]. The findings presented in the research paper underscore the efficiency of the Random Forest technique, showcasing it's superiority in delivering more precise results in the context of ransomware attack detection. The comprehensive analysis and experimentation detailed in the research accentuate the model's ability to outperform other machine learning models, thereby establishing Random Forest as a robust and reliable choice for bolstering the accuracy of ransomware detection systems [13]. In paper [14], successfully incorporated the hierarchical aspects of ransomware attacks into it's implementation. Random Forest techniques offer a notable advantage by requiring fewer input parameters, thereby simplifying the model tuning process. This characteristic proves highly beneficial in obtaining results with relative ease. Moreover, the algorithm's intrinsic nature tends to mitigate overfitting, contributing to it's robust performance. An additional strength lies in the ensemble nature of Random Forests, where increasing the number of trees in the forest works to reduce variance. This property ensures that the model delivers accurate and reliable results without introducing bias. In essence, the algorithm's ability to harness the collective wisdom of multiple trees enhances it's predictive power while maintaining a balanced and unbiased outcome [15]. Firstly, the preprocessing procedure involves three main steps, extracting features from raw bytes, identifying frequent patterns, and normalizing the data. In the feature extraction phase, a virtual machine (VM) is utilized to analyze 32-bit sliding windows, specifically focusing on 4G features. This approach significantly enhances the accuracy of detection. Moving on to frequent pattern mining, the algorithm shifts through the database to extract recurring patterns associated with significant items in the data. The steps are crucial for uncovering meaningful insights by highlighting prevalent structures within the dataset. Lastly, the normalization step is implemented to standardize the data. This ensures uniformity, making the dataset more conducive to subsequent analyses [16]. The first stage entails creating a decision tree that aims to identify the most effective combination of variables [17]. Subsequently, the dataset is partitioned into subtrees based on this identified optimal combination. However, the task of finding the most suitable variable combination is recognized as a non-trivial challenge [17]. A distinctive feature that contributed to the choice of the Random Forest algorithm in the proposed technique is it's efficient training capability without the need for data scaling [7]. This efficiency aspect played a crucial role in the preference for Random Forest in the current approach.

4 Dataset

VirusTotal, a no-cost utility, was utilized to determine whether a file was categorized as ransomware. The methodology was executed on a computer equipped with a Core i7 CPU boasting 8 cores and 16 GB RAM, utilizing Windows 10 operating system. Additionally, Linux 4.1 was employed as the second operating system in the execution of the methodology [1]. The whitelist comprised two categories of executable files [3]. The initial category was sourced from the Windows Platform, while the second category [3] originated from the Portable Applications Environment. Three categories of ransomware files can be found, deriving from distinct families: Cerber, comprising 267 samples; TeslaCrypt, consisting of 315 samples; and Locky, totaling 258 samples [6]. In other words, there are three kinds of ransomware files sourced from various families, including Cerber with 267 instances, TeslaCrypt with 315 instances, and Locky with 258 instances [6]. In the preceding study [14], there are 1680 executable files accessible, encompassing 840 distinct types of ransomware executable files and an additional 840 whitelisted files. All datasets are sourced from VirusTotal [18]. Alternatively, in the earlier research [19], a total of 1680 executable files are present, with 840 belonging to various types of ransomware executables and an equal number of whitelisted files. Random Forest, an ensemble learning technique, utilizes diverse subsets of the training data through bagging, where random samples with replacement create unique training sets for each decision tree. During the tree-building process, features are randomly selected at each split, fostering variability among the trees. The model's strength lies in aggregating predictions from individual trees during the prediction phase, ensuring a robust and accurate overall output. The final result reflects a consensus among the trees, minimizing overfitting and enhancing generalization. This approach makes Random Forest resilient to noise and outliers while improving the model's predictive performance. The use of bagging and random feature selection mitigates the risk of highly correlated trees, contributing to the ensemble's effectiveness. In essence, Random Forest combines the power of multiple decision trees to create a more reliable and versatile predictive model. In the second proposed model, the approach involves a comprehensive preprocessing pipeline for both non-malicious executable files and ransomware malicious files. The process begins with an initial step of feature transformation, where the characteristics of the files are modified to enhance their representation for subsequent analysis. Following this, association rule mining is employed to identify patterns and relationships among the features, aiding in the extraction of meaningful insights. After the association rule mining, the data undergoes standardization using the Random Forest technique. This involves leveraging the power of a Random Forest algorithm to refine and normalize the data, ensuring that it conforms to a consistent and comparable format. Random Forest, being an ensemble learning method, contributes to the robustness and accuracy of the standardization process (Figs. 1 and 2).

To summarize, the preprocessing steps involve feature transformation to enhance representation, association rule mining for pattern discovery, and data standardization through Random Forest, collectively forming a sophisticated approach to prepare

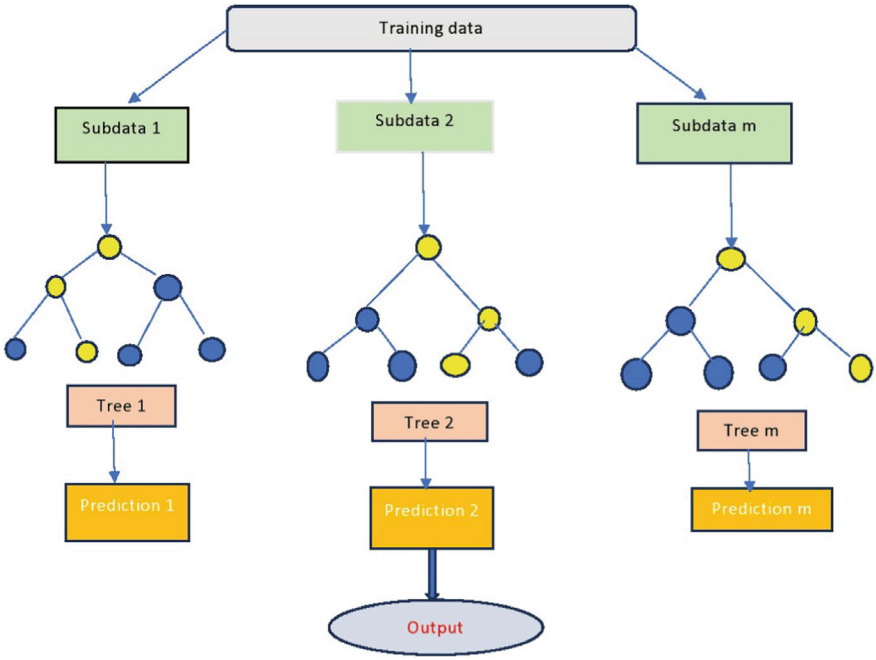


Fig. 1 Our proposed model-1, the proposed model employs bagging, where multiple decision trees are trained on bootstrap samples from the training data. Each tree is built by recursively selecting features to split the data, stopping when a predefined criterion is met. During prediction, the outputs of all trees are aggregated, typically by averaging for regression or by majority voting for classification. The final model output provides predictions for new data points. This ensemble technique enhances model robustness and generalization, making it effective in various machine learning tasks

the data for subsequent stages of analysis and modeling. Following the data standardization phase, the next step involves selecting relevant features based on the gain ratio of the dataset. The gain ratio is a metric that assesses the usefulness of each feature in terms of information gain, helping identify the most informative attributes. Subsequently, a decision tree is employed for the categorization of non-malicious executable files and ransomware malicious files. Decision trees are powerful tools for classification, where the dataset is split based on the selected features and their associated gain ratios. This categorization process enables the model to discern patterns and make informed decisions about the nature of the files, distinguishing between benign and malicious entities. In essence, the gain ratio-driven feature selection optimizes the dataset for classification, while the decision tree facilitates the segregation of non-malicious and ransomware files, enhancing the overall effectiveness of the model in threat detection.

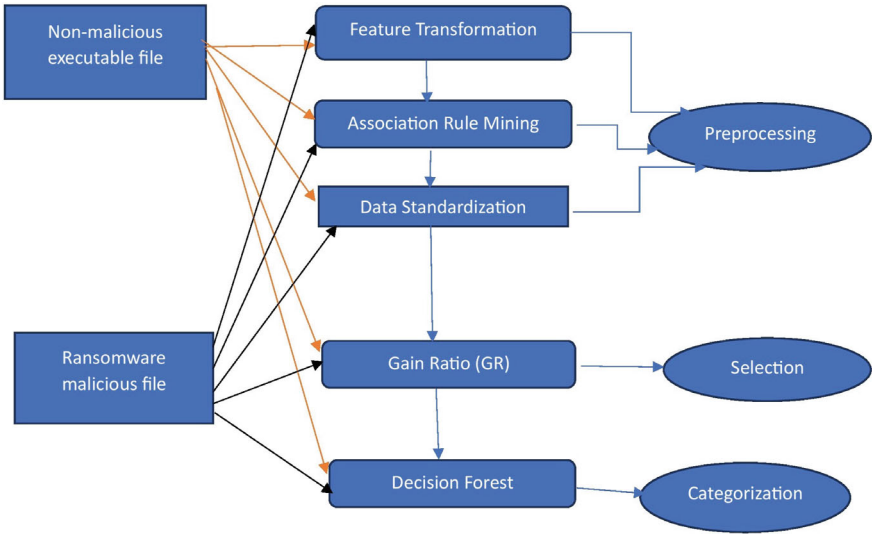


Fig. 2 Our proposed model-2, preprocessing, feature transformation involves scaling numeric features and encoding categorical variables. Associate rule mining discovers relationships between variables, aiding in feature engineering. Data standardization ensures uniform scales across features, preventing dominance of certain variables. Feature selection identifies relevant features using techniques like correlation analysis or feature importance scores. Categorization converts continuous variables to categorical ones, enhancing algorithm performance

5 The Result Analysis

Random Forest model is employed to analyze a dataset comprising 840 files for each group. In each group, a random selection of 420 files is made, specifically focusing on ransomware files for individual analysis. The WEKA GUI with the Random Forest technique for ransomware detection, key evaluation metrics include false positive ratio (FPR), rate of non-ransomware instances wrongly identified as ransomware. False negative ratio (FNR), rate of ransomware instances wrongly identified as non-ransomware. True positive ratio (TPR), rate of correctly identified ransomware instances. True negative ratio (TNR), rate of correctly identified non-ransomware instances. Accuracy measures the percentage of correctly classified instances out of the total. It reflects the overall correctness of a classification model. These metrics offer insights into the model's ability to distinguish between ransomware and non-ransomware, providing a comprehensive assessment of its performance. Now, using the formulas [1, 5, 8].

$$\text{True Positive Rate (TPR): } \mathrm{TPR} = \mathrm{TPTP} + \mathrm{FNTPR} = \mathrm{TP} + \mathrm{FNTP}$$

$$\text{False Positive Rate (FPR): } \mathrm{FPR} = \mathrm{FPFP} + \mathrm{TNFPR} = \mathrm{FP} + \mathrm{TNFP}$$

$$\text{Precision: Precision} = \text{TPTP} + \text{FP Precision} = \text{TP} + \text{FPTP}$$

$$\text{Recall: Recall} = \text{TPTP} + \text{FN Recall} = \text{TP} + \text{FNTP}$$

$$\text{F-Measure: F-Measure} = 2 \times (\text{Precision} \times \text{Recall}) \text{ Precision} + \text{Recall}$$

$$\text{F-Measure: Precision} + \text{Recall2} \times (\text{Precision} \times \text{Recall})$$

These formulas are fundamental metrics for evaluating the performance of a ransomware detection model [8]. Evaluating the performance of the classifier, a comprehensive analysis was conducted on the impact of varying feature sizes, ranging from 100 to 10,000, with a fixed test set comprising 100 instances. Notably, when utilizing a set of 10,000 features, the classifier demonstrated an exceptional accuracy of 98.30%, underscoring it's efficiency in accurate predictions. Further investigation into the classifier's capabilities involved an examination of the F-measure, which consistently surpassed 97.80% across the spectrum of 10,000 features. Additionally, the receiver operating characteristic (ROC) analysis yielded a remarkable result of 99%, affirming the robustness of the model in distinguishing between positive and negative instances.

Table 1 highlights the false positive rate (FPR), false negative ratio (FNR), and true negative rate (TNR) at various feature sizes for classifier.

So here, in the Table 2, for dimensional feature size 100, accuracy level 95.20%, F-measure 93.50, and ROC value 96.20. In the feature size 1000, accuracy level 97%, F-measure value 95.30, ROC value maximum 99%. In the feature size 10,000, our model accuracy level maximum 98.30%, F-measure value 97.80, ROC value maximum 99%. Table 3 shows the evaluation of different classifiers for detecting attacks, including Bagging, Random Forest, Add Boost M1, and Rotation Forest, and it was observed that the performance of Rotation Forest closely approached that of Random Forest. However, it is noteworthy that the construction time for Rotation Forest was significantly higher, approximately 26 seconds, compared to the remarkably faster Random Forest, which only required 1.5 seconds for model building. These findings highlight the efficiency of the Random Forest classifier in terms of both model accuracy and computational speed. While Rotation Forest

Table 1 Highlights the False positive rate (FPR), False negative ratio (FNR), and True negative rate (TNR) at various feature sizes for classifier

Feature size	FPR (%)	FNR (%)	TNR (%)
100	1.29	1.00	95.10
500	2.15	1.92	97.85
1000	1.75	1.20	98.25
5000	0.90	0.75	99.10
10,000	0.50	0.30	99.50

Table 2 Highlights the accuracy, f-measure, ROC analysis at various feature sizes for classifier

Feature size	Accuracy (%)	F-measure	ROC analysis
100	95.20	93.50	96.20
500	96.50	94.80	98.00
1000	97.00	95.30	99.00
5000	98.00	97.20	99.00
10,000	98.30	97.80	99.00

Table 3 Highlights the accuracy, f-measure, ROC analysis at various feature sizes for classifier

Classifier category	FPR	FNR	TPR	TNR
Bagging	0.038	0.073	0.945	0.978
Random forest	0.045	0.001	0.999	0.960
Add Boost M1	0.07	0.19	0.88	0.97
Rotation forest	0.030	0.037	0.972	0.980

demonstrated competitive performance, it's longer model building time may impact it's practical applicability, especially in scenarios where rapid model deployment is crucial. Therefore, based on the evaluation results, Random Forest emerges as the most efficient choice among the classifiers considered for detecting the specified types of attacks.

6 Conclusion

The proposed Random Forest method demonstrates superior performance in byte-level static analysis, outperforming the accuracy levels achieved through behavior analysis detection. The results indicate a noteworthy advantage in accurately assessing and classifying data at the byte level. This underscores the efficacy of the Random Forest approach for static analysis in comparison with behavioral analysis for the given task (Table 4).

Within healthcare organizations, our model demonstrates consistent excellence in evaluating datasets spanning 100–10,000 dimensions. The model is designed to maintain a false positive rate (FPR) below the 0.50% while capping the false negative rate (FNR) at a peak of 0.30%. It ensures precise identification of both positive and

Table 4 Comparing our suggested methodology with the approach presented by Zhang et al. [13]

Approaches	Attributes	Dataset	Correctness	Duration (s)
Random forest	Byte by byte based	840/840	98.30	1.12
Zhang et al. [13]	Operation code based	1787/100	91.43	7.27

negative instances [15]. This is complemented by an outstanding true negative rate (TNR) at 99.50%, contributing to an overall accuracy of 98.30%. The F-measure, a vital metric for healthcare applications, attains a robust 97.80%. In the context of healthcare, where precision is paramount, the model's discriminative power, reflected in the ROC curve's value at the 99%, underscores it's reliability. These findings position our model as a valuable tool for healthcare organizations, promising accurate predictions and efficient utilization across diverse feature sizes in medical data analysis.

References

1. Chittooparambil HJ, Shanmugam B, Azam S, Kannoorpatti K, Jonkman M, Samy GN (2019) A review of ransomware families and detection methods. In: Recent trends in data science and soft computing: Proceedings of the 3rd International conference of reliable information and communication technology (IRICT 2018). Springer International Publishing, pp 588–597
2. Razaulla S, Fachkha C, Markarian C, Gawanmeh A, Mansoor W, Fung BC, Assi C (2023) The age of ransomware: a survey on the evolution, taxonomy, and research directions. IEEE Access
3. Burnap P, French R, Turner F, Jones K (2018) Malware classification using self organising feature maps and machine activity data. Comput Secur 73:399–410
4. Luo X, Liao Q (2007) Awareness education as the key to ransomware prevention. Inf Syst Secur 16(4):195–202
5. Weckst´en M, Frick J, Sj¨ostr¨om A, J¨arpe E (2016) A novel method for recovery from Crypto Ransomware infections. In: 2016 2nd IEEE International conference on computer and communications (ICCC). IEEE, pp 1354–1358
6. Kharaz A, Arshad S, Mulliner C, Robertson W, Kirda E (2016) UNVEIL: a large-scale, automated approach to detecting ransomware. In: 25th USENIX security symposium (USENIX Security 16), pp 757–772
7. Takase H, Kobayashi R, Kato M, Ohmura R (2020) A prototype implementation and evaluation of the malware detection mechanism for IoT devices using the processor information. Int J Inf Secur 19(1):71–81
8. Herrera-Silva JA, Hern´andez-A´lvarez M (2023) Dynamic feature dataset for ransomware detection using machine learning algorithms. Sensors 23(3):1053
9. Smith D, Khorsandroo S, Roy K (2022) Machine learning algorithms and frameworks in ransomware detection. IEEE Access 10:117597–117610
10. Urooj U, Al-rimy BA, Zainal A, Ghaleb FA, Rassam MA (2021) Ransomware detec- tion using the dynamic analysis and machine learning: a survey and research directions. Appl Sci 12(1):172
11. Hsu CM, Yang CC, Cheng HH, Setiasabda PE, Leu JS (2021) Enhancing file entropy analysis to improve machine learning detection rate of ransomware. IEEE Access 9:138345–138351
12. Pan Y, Ge X, Fang C, Fan Y (2020) A systematic literature review of android malware detection using static analysis. IEEE Access 8:116363–116379
13. Zhang H, Xiao X, Mercaldo F, Ni S, Martinelli F, Sangaiah AK (2019) Classification of ransomware families with machine learning based onN-gram of opcodes. Futur Gener Comput Syst 90:211–221
14. Homayoun S, Dehghantanha A, Ahmadzadeh M, Hashemi S, Khayami R (2017) Know abnormal, find evil: frequent pattern mining for ransomware threat hunting and intelligence. IEEE Trans Emerg Top Comput 8(2):341–351
15. Meidan Y, Bohadana M, Shabtai A, Ochoa M, Tippenhauer NO, Guarnizo JD, Elovici Y (2017) Detection of unauthorized IoT devices using machine learning techniques. arXiv preprint arXiv:1709.04647

16. Khammas BM, Monemi A, Ismail I, Nor SM, Marsono MN (2016) Metamorphic malware detection based on support vector machine classification of malware sub-signatures. TELKOMNIKA (Telecommun Comput Electron Control) 14(3):1157–1165
17. Bhalla D (2014) Random forest tutorial. Available: http://www.listendata.com/2014/11/random-forest-with-r.html
18. Mishkovski I, S´cepanovi´c S, Mirchev M, Gramatikov S. Anti-virus tools analysis using deep web malwares
19. Hashemi H, Azmoodeh A, Hamzeh A, Hashemi S (2017) Graph embedding as a new approach for unknown malware detection. J Comput Virol Hacking Tech 13:153–166

Comparative Assessment of Accessibility and Readability in Generative AI: A Case Study of GPT and Google BARD

C. P. Afsal and K. S. Kuppusamy

Abstract The accessibility and readability of Generative Artificial Intelligence systems like GPT and Google BARD are crucial factors that require thorough examination. In today's digitally connected world, where AI-generated content is becoming increasingly prevalent across various sectors, ensuring that such content is accessible to a broad audience, easy to understand, and user-friendly is essential. This research was motivated by the growing influence of AI-generated content on various user groups, including individuals with varying levels of familiarity with technology and diverse cognitive abilities. This paper comprehensively assesses and draws comparisons between the accessibility and readability of Google BARD and GPT to identify areas for enhancement. GPT exhibits various accessibility issues across different WCAG 2.0 conformance levels, contrasting with Google BARD, which shows no accessibility problems. The results underscore the importance of addressing accessibility concerns in AI models to ensure inclusivity and user-friendliness for individuals with disabilities. Google BARD consistently provides more readable content than GPT, underscoring the importance of selecting the appropriate tool based on the target audience and desired readability.

Keywords Accessibility · Readability · GPT · Google BARD · Generative AI · Artificial intelligence

1 Introduction

In the constantly evolving realm of digital content and technology, ensuring that information is accessible and readable is significant. Accessibility and readability evaluation are crucial in guaranteeing that content and interfaces are available to a

C. P. Afsal (✉) · K. S. Kuppusamy
Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India
e-mail: afsalcp2428@pondiuni.ac.in

K. S. Kuppusamy
e-mail: kskuppu@pondiuni.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_34

diverse audience and easily understandable and user-friendly [1]. These two pillars intersect to create an inclusive and practical user experience across various platforms, from websites and applications to documents and user interfaces [2]. This analysis aims to delve into the accessibility and readability aspects of two prominent entities in the field of artificial intelligence and language processing: OpenAI's GPT and Google BARD.

GPT and Google BARD are both advanced natural language processing models that act as conversational generative artificial intelligence chatbots that have garnered significant attention in the field of artificial intelligence [3]. They can generate human-like text, understand context, and perform various language-related tasks. However, they differ in certain aspects of their architecture and design [4, 5]. Analyzing accessibility in the context of generative AI chatbots, like GPT and Google BARD, is highly crucial for plenty of influencing reasons. This attempt highlights the basic values of inclusion and equitable access, overcoming individual abilities and ensuring that technology is accessible to all [6]. This ethical commitment extends to following legal responsibilities, preserving equality, and avoiding potential legal consequences. Also, accessibility efforts are consistent with the ethical use of AI, encouraging social equality by not excluding those with disabilities [7].

In real life, enhancing accessibility improves the user experience across all platforms with well-structured design, clear communication, and efficient navigation that accommodates a wide range of users. Making AI chatbots available opens up previously unexplored areas, increasing user engagement and commercial prospects. Prioritizing accessibility encourages creativity, resulting in unique solutions that enable AI chatbots to adapt to a wide range of user needs [8]. Spreading awareness about accessibility issues promotes understanding and compassion among both developers and consumers. Ultimately, addressing accessibility in AI chatbots promotes a technologically inclusive, creative, and equal environment [9]. The accessibility can be measured in three levels based on WCAG 2.0 (Web Content Accessibility Guidelines) conformance levels: completely accessible, partially accessible, and inaccessible. A website is considered to be accessible when its contents are easily available to users regardless of disabilities [10].

The significance of research studies on the readability of information generated by AI chatbots such as GPT and Google BARD is undeniable [11]. The readability of AI-generated material directly impacts user comprehension and interaction, influencing how effectively information is communicated [12, 13]. Ensuring clear and intelligible material is especially important for improving accessibility since it allows information to be accessed by people with varying literacy levels and cognitive capacities. So, easily readable material encourages confidence, credibility, and a favorable user experience, which contributes to user satisfaction and effective communication [14].

1.1 Research Objectives

The primary aim of this study is to comprehensively evaluate and draw comparisons between the accessibility and readability aspects of two prominent conversational AI chatbots, GPT and Google BARD. In the context of this study, the term 'GPT' explicitly refers to the publicly accessible GPT 3.5 version. Thus, any mention of 'GPT' throughout this paper exclusively refers to GPT 3.5. The overall objectives of this research endeavor can be briefly summarized as follows:

1. To determine the accessibility scores of GPT and Google BARD in terms of WCAG 2.0 guidelines by employing specialized accessibility evaluation tools.
2. Using different readability assessment tools, measure the readability of text generated by GPT and Google BARD across varying levels of prompts.
3. Carry out an in-depth comparative analysis of accessibility and readability aspects between GPT and Google BARD by using a statistical approach.

2 Related Work

This section offers a brief overview of previous research concerning the readability and accessibility of chatbots. We also look through the assessment techniques used in this context.

A study highlights the significance of addressing the challenges faced by people with disabilities regarding chatbots and conversational interfaces. The paper draws attention to the lack of specialized research on chatbot accessibility and suggests more research on user interactions and ways to make these interfaces more accessible. [15]. The characteristics and quality problems associated with the development and implementation of chatbots provide a method for evaluating quality based on the Analytic Hierarchy Process (AHP). The ability to compare conversational systems, the use of quality attributes as a checklist for chatbot implementation, and the flexibility of the process for assessing various performances over time [16].

A novel strategy to enhance chatbot interaction and accessibility for individuals with hearing loss, utilizing clarity and Italian Sign Language multimedia. The study's findings show improved content understandability and user engagement, highlighting the potential of user-friendly chatbot interfaces for better integration of people with deafness into society [17]. A paper discusses training and adapting chatbots using user-supplied training corpora, evaluating them through linguistically motivated comparisons, and advocating open-ended trials with users. The final remarks underscore the necessity for customized evaluation based on application and user requirements [18]. A study investigates the state of chatbot usability, the applied human–computer interaction techniques, and the evaluation of chatbot usability. The study outcome highlights the necessity for more rigorous user experience testing and implementing usability-conscious design principles for chatbot creation [19].

The linguistic accuracy of various chatbots focuses on their ability to provide grammatically accurate answers and meaningful responses. While chatbots show

progress in grammar, advancements are needed for generating coherent and meaningful responses in human-like conversations [20]. A paper evaluates the usability of the iHelpr chatbot for mental health care, finding that participants found it enjoyable and easy to use but identified areas needing improvement, such as Error Management and Intelligence. The study concludes that further research is needed to assess the chatbot's usefulness in a broader context of individuals with mental health issues [21]. Chatbot usability highlights the current research state and advocates for more formal experiments to enhance user experience. It stresses the importance of usability techniques, characteristics, and research methods in evaluation and underscores the need for usability-aware design guidelines in chatbot development [22].

Exploring the accessibility and readability of AI-generated content via GPT and Google BARD offers a unique opportunity to refine the technology and contribute to designing AI systems that are inclusive, informative, and user-centered. This study aims to significantly influence AI-generated content's ethical and practical dimensions, thereby serving as a cornerstone for improving individuals and societies.

3 Methodology

In this paper, our primary objective is to comprehensively assess and draw comparisons between the accessibility and readability of Google BARD and GPT in order to identify areas for enhancement. The study was conducted during the middle of August 2023. Our methodology involves conducting separate experiments to evaluate these essential aspects.

3.1 Accessibility

Web accessibility is an important principle that ensures equal access and usability of digital content for all individuals, including those with disabilities. Regardless of a user's ability, inclusive web design seeks to remove obstacles preventing them from interacting with websites and applications. By systematically evaluating websites for conformance to accessibility standards and guidelines, web accessibility evaluation tools play a crucial role in helping to realize this objective. These tools scan websites for potential problems related to text alternatives for images, keyboard navigation, color contrast, and more. These tools enable designers and developers to address accessibility issues by highlighting places that need renovation, thereby promoting a more inclusive online environment. Using evaluation tools and embracing online accessibility shows a commitment to social responsibility, improves user engagement, and increases the audience for digital content.

Accessibility Evaluation Tools A wide range of accessibility evaluation tools are available, spanning both commercial and free options. In our study, we employed

two primary accessibility evaluation tools to assess the accessibility of both the GPT and BARD. For accessibility assessment, we opted for a focused approach by employing two widely recognized accessibility evaluation tools: AChecker [23] and WAVE [24]. These tools evaluated the accessibility of the appropriate web pages by simply inputting their respective URLs. These tools play a crucial role in examining the accessibility aspects of digital content.

3.2 Readability

In the context of textual communication, readability refers to the clarity and readability of written material for readers. Readability becomes much more crucial when used in the context of text generated by language models like Google BARD and GPT. Readability ensures that the generated content is accessible and useful for a diverse audience by creating content that is concise and appropriately formatted. This applies to users with different cognitive capacities, linguistic backgrounds, and literacy levels. Ultimately, this improves the overall user experience and improves the applicability of language models.

The readability evaluation was performed on the text generated by both chatbots in response to distinct user prompts posed through their respective user interfaces. For this purpose, a set of ten diverse prompts was curated, maintaining an equal distribution between subjects related to science and technology. We presented each prompt to the chatbots individually, progressing consecutively through a sequence of levels. The various levels associated with each prompt are outlined in Table 1.

In our study, we focused on specific readability formulas to evaluate the readability of text generated by both GPT and Google BARD. While numerous formulas exist, we selected the following for our assessment provided in Table 2.

4 Experimental Result

In this section, we present the findings obtained from various experiments conducted to evaluate the accessibility and readability of GPT and Google BARD.

Table 1 Prompt levels

S. No.	Level	Description
1	Level I	Prompt asked with 100 words
2	Level II	Prompt asked with 125 words
3	Level III	Prompt asked with 150 words
4	Level IV	Prompt asked with 175 words
5	Level V	Prompt asked with 200 words

Table 2 The readability formulas utilized in this study

S. No.	Readability formula	Equation
1	FRE [25]	$\text{FRE} = 206.835 - (1.015 \times \text{ASL}) - (84.6 \times \text{ASW})$
2	FKGL [26]	$\text{FKGL} = (0.39 \times \text{ASL}) + (11.8 \times \text{ASW}) - 15.59$
3	ARI [27]	$\text{ARI} = 4.71 * ACW + 0.5 * AWS - 21.43$
4	CLI [28]	$\text{CLI} = (0.0588 * L) - (0.2965 * S) - 15.8$
5	GFOG [29]	$\text{GFOG} = 0.4(AWS + PCW)$
6	SMOG [30]	$\text{SMOG} = 3 + \sqrt{\text{PC}}$

FRE: Flesch-Kincaid Reading Ease, **FKGL**: Flesch-Kincaid Grade Level, **ARI**: Automatic Readability Index, **CLI**: Coleman-Liau Index, **GFOG**: Gunning Fog Index, **SMOG**: Simple Measure of Gobbledygook, **ASL**: Average Sentence Length, **ASW**: Average Syllables per Word, **ACW**: Average Complex Words, **AWS**: Average Words per Sentence, **L**: Average number of letters per 100 words, **S**: Average number of letters per 100 words, **PCW**: Percentage Complex Words, **PC**: Polysyllable Count

4.1 Accessibility Analysis

The comprehensive findings concerning accessibility issues of GPT and Google BARD across different levels of WCAG 2.0 conformance are presented in Tables 3 and 4, respectively, as reported by the AChecker tool. The collective assessments of accessibility challenges for both GPT and Google BARD across varying levels of WCAG 2.0 conformance are presented in Table 5, as reported by the WAVE tool. The graphical representation of accessibility analysis results of GPT and Google BARD reported by AChecker and WAVE tool is depicted in Fig. 1.

Comparing the result reported by AChecker, it is clear that GPT demonstrates various accessibility issues across different levels of WCAG 2.0 conformance, as

Table 3 Classification of level-wise problems of WCAG 2.0 by AChecker for GPT

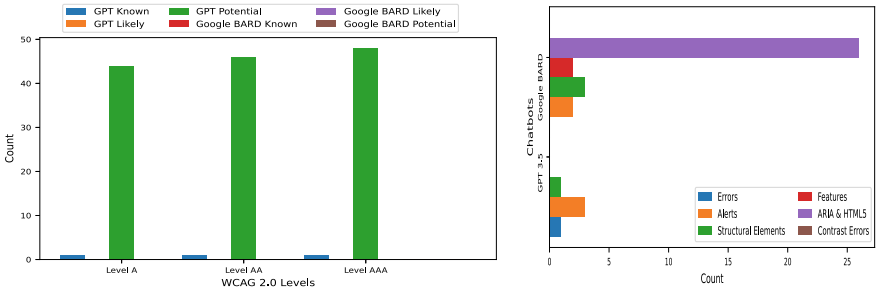
Levels	WCAG 2.0			
	Known problems	Likely problems	Potential problems	Total problems
Level A	1	0	44	45
Level AA	1	0	46	47
Level AAA	1	0	48	49

Table 4 Classification of level-wise problems of WCAG 2.0 by AChecker for Google BARD

Levels	WCAG 2.0			
	Known problems	Likely problems	Potential problems	Total problems
Level A	0	0	0	0
Level AA	0	0	0	0
Level AAA	0	0	0	0

Table 5 Classification of problems reported by Wave tool for GPT and Google BARD

Chatbot	Errors	Alerts	Structural elements	Features	ARIA and HTML5	Contrast errors
GPT	1	3	1	0	0	0
Google BARD	0	2	3	2	26	0



(a) AChecker Tool (b) WAVE Tool

Fig. 1 Problems reported by accessibility evaluation tools for GPT and Google BARD

indicated by the presence of known, likely, and potential problems. On the other hand, Google BARD does not appear to have any accessibility issues across any of the WCAG 2.0 levels. This remarkable contrast highlights the potential difference in the accessibility performance between GPT and Google BARD. These findings indicate the significance of addressing and rectifying the identified accessibility issues in GPT to ensure its generated content is more inclusive and useful for a wider audience, including people with disabilities. These results suggest the significance of conducting thorough accessibility assessments of AI systems and the subsequent necessity for enhancing AI models to provide an inclusive user experience for all individuals.

Table 5 showcases a detailed comparison of accessibility-related issues identified by the Wave tool in both GPT and Google BARD. Across various categories, provide insight into the accessibility performance of the two chatbots. GPT reported one error and three alerts, indicating potential areas of concern with respect to its adherence to accessibility guidelines. In contrast, Google BARD demonstrated a broader array of issues, with two alerts and three problems related to structural elements. Remarkably, Google BARD exhibited many issues (26) concerning ARIA & HTML5, indicating potential limitations in leveraging these technologies to improve accessibility. Interestingly, GPT did not encounter any issues in this category. The fact that both chatbots reported any contrast issues suggests that the text-background contrast guidelines were followed correctly. The accessibility of both chatbots may be greatly improved by addressing the concerns found, resulting in a more pleasant and user-friendly experience for those with various abilities.

4.2 Readability Analysis

In this experiment, we conducted a comprehensive analysis of the readability of text generated by both GPT and Google BARD in response to a set of diverse prompts. The prompts were presented at five levels (Table 1), each designed to gradually increase in length, thereby providing a range of challenges to the chatbots' text generation capabilities. This experiment was designed to achieve three main goals: first, to assess how well the chatbots adhered to user-defined requirements with respective levels in terms of the word count in the generated text; second, to analyze and establish a connection between the readability of the text across various levels; and third, comprehensive comparison of the readability of GPT and Google BARD.

Word Count Analysis: Table 6 and Fig. 2 represent a comprehensive overview of the minimum, maximum, and average number of words generated by GPT and Google BARD responding to prompts at various length levels. GPT showcased adaptive response generation, consistently meeting specified word count requirements across prompts of varying complexity. Its flexibility was evident as it produced a minimum of 100 words for a 100-word prompt, reaching up to 267 words for a 200-word prompt, with an average response length increasing from 112.2 to 229.9 words.

Google BARD consistently exceeded specified word limits across varying prompt lengths, generating responses with a minimum of 107 words for a 100-word prompt and reaching 555 words for a 200-word prompt. The average word count increased steadily from 181.5 to 361.8 words, showcasing a trend of consistently longer responses compared to GPT.

Readability Correlation Analysis: Table 7 offer a glimpse into the average readability scores attained by GPT and Google BARD across various readability formulas and prompt levels. Specifically, these tables present insights into how readability assessments vary in response to different levels of prompts.

In our study, it's important to note that lower scores indicate higher readability levels for all of these readability measures except FRE readability formula.

As the word count increases from 100 to 200 words, there is a mixed pattern in the readability scores obtained from GPT and Google BARD across various readability measures. While some measures such as FRE, FKGL, ARI, CLI, and GFOG show an inconsistent trend in readability with fluctuations across different word counts, others like SMOG display a relatively stable or slightly decreasing trend in

Table 6 Minimum, maximum and Average no. of words obtained from GPT and Google BARD against prompt asked in various levels mentioned in Table 1

Statistics	100 words		125 words		150 words		175 words		200 words	
	GPT	BARD	GPT	BARD	GPT	BARD	GPT	BARD	GPT	BARD
Minimum	100	107	102	128	138	211	166	270	203	284
Maximum	141	280	139	310	171	446	199	418	267	555
Average	112.2	181.5	125.8	236.7	151.2	334	182	354	229.9	361.8

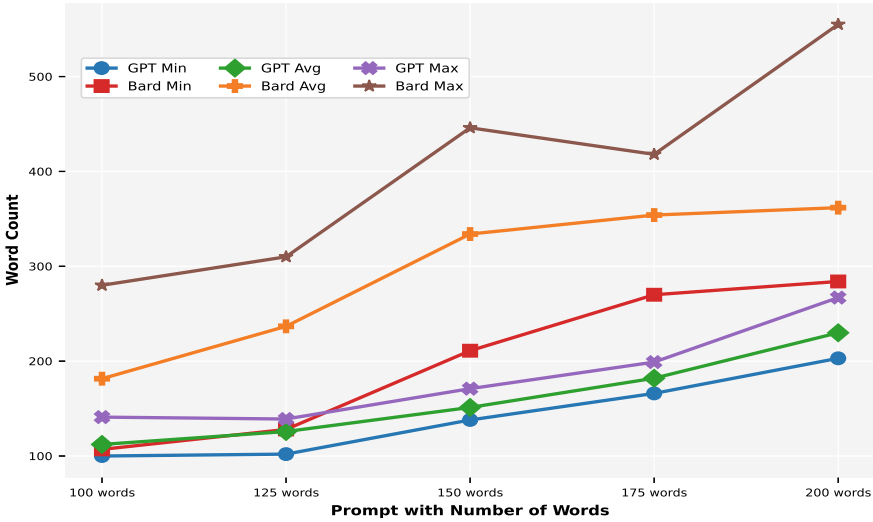


Fig. 2 Minimum, maximum, and average no. of words generated by GPT and Google BARD against the prompt

Table 7 Average readability score obtained from GPT and Google BARD through various readability measures

Formula	100 words		125 words		150 words		175 words		200 words	
	GPT	BARD	GPT	BARD	GPT	BARD	GPT	BARD	GPT	BARD
FRE	11.56	42.39	7.38	42.55	10.18	43.38	12.09	43.19	10.13	44.50
FKGL	16.84	11.78	16.97	11.86	16.82	11.57	16.33	11.69	16.62	11.41
ARI	18.80	12.26	18.57	12.32	18.62	11.95	18.02	12.02	17.88	11.80
CLI	19.41	13.01	20.17	12.83	19.69	12.78	19.48	12.68	19.27	12.60
GFOG	19.85	13.83	19.68	14.03	20.12	13.76	19.17	13.90	20.01	13.61
SMOG	40.47	34.79	39.53	34.98	40.03	34.39	39.09	34.36	39.28	34.30

readability scores. Overall, the findings suggest that the impact of word count on readability varies depending on the specific readability measure used, with no clear linear relationship observed across all measures.

The correlation between word count and readability score varies across different readability measures, with some showing a positive correlation, while others display no clear pattern or even a negative correlation. This relationship is influenced by readability formula, content characteristics, text complexity, and readability; therefore, in order to identify particular correlations, an in-depth study of each readability measure is required.

Readability Comparison Analysis: We examine the average readability scores produced from GPT and Google BARD in detail by carefully examining Table 7,

which presents the GPT and Google BARD readability scores. These table encompass various readability measures, prompts, and word count levels, facilitating a thorough investigation of the readability trends provided by each tool. We will thus evaluate their effectiveness in generating easily understandable content.

In Table 7, the average readability scores derived from GPT, we observe distinct trends across different readability measures, prompts, and word counts. Irrespective of the readability measure used, the content generated by GPT consistently showcases relatively lower readability scores. This shows that GPT tends to produce text that may take a certain level of understanding or effort across many prompts and word count variations. It is important to note that the SMOG scores, which measure content complexity, typically tend to be higher than average. This suggests that GPT's content occasionally incorporates intricate language structures and vocabulary, contributing to the overall complexity of the text.

However, a distinct pattern becomes noticeable in the average reading ratings obtained from Google BARD. Higher average readability scores are seen in content produced by Google BARD, indicating that this content is generally simpler to understand and accessible across a range of prompts, word counts, and measurements. This implies that Google BARD may provide content that is better suited for simple understanding, which could be useful in a variety of situations.

5 Discussion and Conclusion

In this paper, we thoroughly assess and draw comparisons between the accessibility and readability of Google BARD and GPT to identify areas for enhancement. GPT exhibits diverse accessibility issues across WCAG 2.0 levels, contrasting sharply with Google BARD's absence of such issues, emphasizing the need to enhance GPT's accessibility for broader inclusivity and usability, particularly for individuals with disabilities. Addressing identified concerns in both chatbots could significantly enhance their user experience, emphasizing the importance of thorough accessibility assessments in AI systems.

When comparing the two tools, Google BARD offers content with generally higher readability scores than GPT. This implies that Google BARD's content might be more suited in contexts that require simple, concise language, even in cases where the prompt's complexity or word count varies. Although each platform has advantages and issues of its own, in general, Google BARD appears to provide information that is consistently more readable than GPT. This analysis highlights the significance of selecting the appropriate tool based on the target audience and the desired level of readability in generated content.

We presented each prompt with a specific word limit, expecting a concise paragraph explanation of the topic. However, GPT generated the content in paragraph format, while Google BARD provided mixed information, including sentences, bullet points, and tables. Google BARD's advantage lies in providing content sources,

related topics, and anticipated prompts for future exploration. This analysis demonstrates the text generation capabilities of both GPT and Google BARD. Both chatbots demonstrated a capacity to generate responses that align with required word counts for various levels of prompts. While GPT consistently adapted its responses, Google BARD displayed a tendency to produce longer replies on average. In terms of readability, Google BARD outperforms GPT, suggesting its potential superiority in generating content that is easier to comprehend. These findings underscore the need for ongoing improvements in accessibility across both GPT and Google BARD to promote inclusivity, particularly for users with disabilities, while also highlighting their unique text generation patterns and abilities to adapt responses to diverse levels of prompt complexity.

References

1. Ansari B, Barati M, Martin EG (2022) Enhancing the usability and usefulness of open government data: a comprehensive review of the state of open government data visualization research. Gov Inf Q 39(1):101657
2. Kumar S, Shree DVJ, Biswas P (2021) Comparing ten WCAG tools for accessibility evaluation of websites. Technol Disabil 33(3):163–185
3. Ahmed I, Roy A, Kajol M, Hasan U, Datta PP, Reza MR (2023) Chatgpt vs. bard: a comparative study. Authorea Preprints
4. Kasneci E, Seßler K, Küchemann S, Bannert M, Dementieva D, Fischer F, Gasser U, Groh G, Günnemann S, Hüllermeier E et al (2023) Chatgpt for good? On opportunities and challenges of large language models for education. Learn Indiv Diff 103:102274
5. Dwivedi YK, Kshetri N, Hughes L, Slade EL, Jeyaraj A, Kar AK, Baabdullah AM, Koohang A, Raghavan V, Ahuja M et al (2023) "So what if chatgpt wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. Int J Inf Manage 71:102642
6. Imrie R (2012) Universalism, universal design and equitable access to the built environment. Disabil Rehabil 34(10):873–882
7. Morrison C, Cutrell E, Dhareshwar A, Doherty K, Thieme A, Taylor A (2017) Imagining artificial intelligence applications with people with visual disabilities using tactile ideation. In: Proceedings of the 19th International ACM Sigaccess conference on computers and accessibility, pp 81–90
8. Ray PP (2023) Chatgpt: a comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and CyberPhysical Syst
9. Santos RPd (2023) Enhancing chemistry learning with Chatgpt and Bing chat as agents to think with: a comparative case study. arXiv preprint arXiv:2305.11890
10. Ismail A, Kuppusamy K (2018) Accessibility of Indian universities' homepages: an exploratory study. J King Saud Univ-Comput Inf Sci 30(2):268–278
11. Meyer JG, Urbanowicz RJ, Martin PC, O'Connor K, Li R, Peng PC, Bright TJ, Tatonetti N, Won KJ, Gonzalez-Hernandez G et al (2023) Chatgpt and large language models in academia: opportunities and challenges. BioData Mining 16(1):20
12. Gill SS, Kaur R (2023) Chatgpt: vision and challenges. Internet of Things CyberPhysical Syst 3:262–271
13. Wu J, Gan W, Chen Z, Wan S, Lin H (2023) Ai-generated content (AIGC): a survey. arXiv preprint arXiv:2304.06632
14. Shneiderman B, Plaisant C (2010) Designing the user interface: strategies for effective human-computer interaction. Pearson Education India

15. Rapp A, Curti L, Boldi A (2021) The human side of human-chatbot interaction: a systematic literature review of ten years of research on text-based chatbots. Int J Hum Comput Stud 151:102630
16. Radziwill NM, Benton MC (2017) Evaluating quality of chatbots and intelligent conversational agents. arXiv preprint arXiv:1704.04579
17. Apuzzo C, Burresi G (2022) Designing accessible chatbots for deaf people. In: 2022 11th Mediterranean conference on embedded computing (MECO). IEEE, pp 1–4
18. Abu Shawar B, Atwell E (2004) Evaluation of chatbot information system. In: Proceedings of the Eighth Maghrebian conference on software engineering and artificial intelligence
19. Ren R, Castro JW, Acun˜a ST, De Lara J (2019) Usability of chatbots: a systematic mapping study. In: SEKE, pp 479–617
20. Coniam D (2014) The linguistic accuracy of chatbots: usability from an ESL perspective. Text & Talk 34(5):545–567
21. Cameron G, Cameron D, Megaw G, Bond R, Mulvenna M, O'Neill S, Armour C, McTear M (2019) Assessing the usability of a chatbot for mental health care. In: Internet science: INSCI 2018 International workshops, St. Petersburg, Russia, October 24–26, 2018, Revised Selected Papers 5. Springer, pp 121–132
22. Ren R, Castro JW, Acun˜a ST, de Lara J (2019) Evaluation techniques for chatbot usability: a systematic mapping study. Int J Softw Eng Knowl Eng 29(11n12):1673–1702
23. AChecker (2023) AChecker (online). https://achecks.org/achecker/
24. Wave (2023) Web accessibility evaluation tools (online). https://wave.webaim.org/
25. Flesch R (1948) A new readability yardstick. J Appl Psychol 32(3):221
26. Kincaid JP, Fishburne Jr RP, Rogers RL, Chissom BS (1975) Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel
27. Smith EA, Senter R (1967) Automated readability index, vol 66. Aerospace Medical Research Laboratories, Aerospace Medical Division
28. Coleman M, Liau TL (1975) A computer readability formula designed for machine scoring. J Appl Psychol 60(2):283
29. Gunning R (1952) The technique of clear writing
30. Begeny JC, Greene DJ (2014) Can readability formulas be used to successfully gauge difficulty of reading materials? Psychol Sch 51(2):198–215

Fuzzy Logic Based Trust Evaluation and Malicious Node Detection During DoS and DDoS Attack in IoT Networks

Himani Tyagi, Rajendra Kumar, and Santosh Kumar Pandey

Abstract In this connecting world of IoT, communication is in danger from insider and outsider attacks. The network nodes are continuously threatened to be attacked and fooled by the one with whom they are communicating. For this problem, a trust value evaluation and monitoring model for smart home-based IoT networks is proposed in this work. For accurate evaluation of trust values, the communication-specific features are taken and selected as compared to the random selection of features. The feature set shown helps evaluate the correct trust value of a node in an IoT network. Then, the membership value for each feature in trustworthy set is estimated, based on which the final trust value using min–max composition is evaluated. Also, using machine learning based K means clustering approach, malicious nodes with similar communication characteristics are grouped in one group and normal nodes in another group with similar characteristic values. The proposed model is a fuzzy logic based trust evaluation FLBTE and clustering-based malicious node detection model during DoS or DDoS attack, showing 97% detection accuracy.

Keywords Fuzzy logic · Machine learning · Malicious node · Detection · K means clustering · Membership function · Attacks · Insider · Trust evaluation · IoT

H. Tyagi (✉) · R. Kumar
Department of Computer Science, Jamia Millia Islamia, New Delhi, India
e-mail: gaura.15.03.2021@gmail.com

R. Kumar
e-mail: rkumar1@jmi.ac.in

S. K. Pandey
Department of Electronics and Information Technology, Ministry of Electronics and Information Technology, New Delhi, India
e-mail: santosh.pandey@meity.gov.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_35

1 Introduction

The uncertainty lies in every event of the environment. The characteristic of nature is that only one instance can change the whole working of the system. Likewise, sudden changes in environmental parameters make it challenging to predict the weather accurately. While talking about IoT-based Wireless Sensor Networks, researchers often quoted that the IoT environment being ubiquitous and requiring seamless internet connection makes it vulnerable to many insider and outsider attacks. Insider attacks are conducted by some internal node controlled by some affected program, whereas outsider attacks require an outsider entity to conduct or manipulate nodes in the network. Due to these security challenges, the adoption of IoT by users has become another challenging task. Hence, proposing solutions for these problems is of utmost importance.

Methods and approaches are proposed in the literature for dealing with attacks and vulnerabilities in IoT networks. They are known as anomaly and attack detection models. Modern techniques like machine learning, artificial intelligence, neural networks, and deep learning are popular for designing anomaly and attack detection models. These models are primarily responsible for dealing with anomalous transactions in which one communicating node is attacker and other is a normal node. The proposed model in this paper deals with detecting which communicating node is an attacker so that it does not communicate with the malicious node in future or remove that path from the passage. This model will help routing algorithms to find the optimum and reliable path for communication.

Additionally, many effective solutions have been proposed in the literature. However, they majorly talk about outsider attacks like MIRAI, man in middle, and bot attacks, for which many network-based Intrusion Detection Systems (IDS) [1, 2] and cryptography-based encryption and decryption techniques are popular [3]. IDS for ON–off [4], information leakage, bad-mouthing attack, good mouthing attack, sybil attack [5], HELLO flood, Selective Repeat, black hole, or only the malicious behaviors are addressed repeatedly in recent research on protecting IoT environment (smart homes, smart vehicular systems, smart health care systems, and smart transportation). But, other types of attacks common in IoT environments like DoS and DDoS attacks that can be insider or outsider attacks are hardly considered. Additionally, the solutions proposed are general solutions not related specifically to IoT. Hence, there is an utmost need to design a protective system against these types of attacks.

Insider attacks are also termed as trust-based attacks. Trust-based attacks are those that try to attack the trust of other nodes by wrongly behaving. The wrong behavior can be due to malign intentions or just a selfish nature of a node. The accurate behavior analysis is a challenging task in an IoT environment where nodes communicate using air as a medium, which is dynamic and uncertain. However, with the help of accurate behavior analyzing techniques and trust parameters, the trust of a node in the network can be effectively accessed.

For accurate behavior analysis, first step is the quantification of behaviors. As in IoT environment, the trust factors are the parameters that represent the properties of a node. With the help of these factors, trust of node is quantified. The value is known as trust value. The higher the value, the higher the trust, resulting in secure communication through that path. The less trustworthy nodes are to be eliminated from the network. But, the sudden change in the behavior of nodes in an IoT-based Wireless Sensor Network is hard to predict the trust value of a node accurately. In WSNs like IoT and MANET, the nodes are vulnerable to internal attacks and easily convert from normal to malicious nodes. Therefore, its continuous evaluation and monitoring is important for uninterrupted functioning of IoT networks.

TMS are powerful security tools to defend against insider attacks and malicious node detection. Various algorithms, models, and approaches have been proposed for designing TMS. The difference lies in the technique used for trust evaluation and synthesis. Major work has been done on anomaly detection, which involves one communicating node as a malicious node, but which is malicious cannot be detected using anomaly detection systems. The work is limited to anomaly detection or trust evaluation. The extension part of malicious node detection and containment is missing or rarely addressed. Hence, in this paper, an approach to more accurately predict the trust values of a node so that a malicious node is discarded from the path of further communication. In this paper fuzzy logic based approach is used to quantify the trust values of the node. Then, a machine learning based K Means clustering approach is used to detect the malicious node. The major contributions of this study are listed below:

RC1: A valid set of communication-specific features is selected compared to a randomly selected feature set for correct evaluation of trust value of a node.
RC2: A robust model is proposed as the number of bad and good nodes being equal.
RC3: The trust-based model is a fuzzy logic based trust evaluation model which is robust and more reliable for networks like IoT.
RC4: The proposed model is node detection compared to the models for anomaly or attack detection models.
RC5: The proposed model is a combination of modern fuzzy logic and machine learning based malicious node detection compared to traditional trust evaluation methods based on weighted summation, which is not a robust and reliable method.

2 Related Work

This section of the paper includes the work done in literature and a comparison with our proposed work on the basis of Trust Indicators, its estimation approach, simulator used in study, attacks addressed in study, Results comprising of advantages and disadvantages of the proposed method, and finally whether it offers algorithmic-based node detection or just the trust value evaluation of nodes.

As in [6], a simple model employing popular Dempster–Shafer Theory for trust evaluation is discussed. The work includes smoke and flame temperature detectors for detecting any misbehavior. Feature set consists of readings from these three types of sensors which are converted into 0–1 range as mass values. The proposed method showed Accuracy (95%), Precision (85.71%), Recall (97.3%), F-Measure (91.14%), and AUC (99%).

To analyze the node behaviors DS theory is used in [7], the work is based on true meaning of trust that is hard to acquire and easy to lose. The proposed work uses a series of trust indicators to analyze the accurate behavior of malicious nodes. The model can detect misbehaviors but the performance of proposed model on IoT-based attacks is to be validated.

In [8] another trust model is proposed that uses number of successful and unsuccessful transactions between the trustor and trustee. The model is simple considering only one type of behavior.

Further, in [9] more factors influencing trust are considered like communication trust, data consistency, and energy trust to evaluate nodes' trustworthiness. The method is computationally intensive.

Another work [10] uses Hidden Markov Model for analyzing the real behavior of nodes to detect malicious and non-malicious nodes. The method showed promising results for particular attack detection, but a popular IoT attack DDoS is not considered in this study.

In [11] a trust model is developed using direct and indirect trust values. The evaluated node's direct trust value is derived by Bayesian and periodically updated based on the combination of effective history records and adaptive decay factor. A filtration method is employed to make it lightweight by introducing a comparison based on confidence in direct value. If it is less than a threshold then only direct trust is evaluated. The method is a lightweight trust evaluation method (Table 1).

3 Methodology

3.1 Work Flow

Phase 1: Dataset, Feature set, and Membership Function Evaluation.

3.1.1 Dataset

The dataset accumulation is the first step in building a successful model as shown in Fig. 1. This study uses an IoT-specific Bot-IoT dataset that includes all the communication logs between nodes. Because of its applicability and availability, this dataset has been used by researchers to design an anomaly detection model for IoT. This study uses this dataset for trust evaluation and malicious node detection in the IoT

Table 1 Related work in the literature study

Reference	Trust indicators	Trust estimation approach	Simulator	Attacks addressed	Results	Node detection
[6]	Sensor readings dataset	Min–max normalization technique	NA	Malicious behavior based on threshold	Advantage: the method is an IoT-specific solution Limitations: the normalization technique transformation do not cater the real meaning	No
[7]	Received packet rate, successfully sending packets rate, data forwarding rate, consistency factor, time factor, security grade	Direct trust evaluation-DS theory based	MATLAB	Malicious behavior	Advantage: Very simple trust model with many trust indicators Limitations: lacks to consider IoT-Specific attack type	No
[8]	No. of Successful and unsuccessful transactions	Direct evaluation-Number of successful and unsuccessful events Indirect evaluation-confidence interval based recommendation approach	Omnett++ 5.6	Bad mouthing attack, on–off attack	Advantages: Introduction of penalty and reward term results in robust and correct trust values Limitations: The results are not validated against node detection. No proper justification of trust indicators	No

(continued)

Table 1 (continued)

Reference	Trust indicators	Trust estimation approach	Simulator	Attacks addressed	Results	Node detection
[9]	Communication trust, data consistency, energy trust	Direct evaluation-Beta density function Indirect Evaluation-Recommendation based	MATLAB	Data attack, on–off attack	Advantages: introduction to balance factor in trust aggregation and windowing method Limitations: Small window size makes it computation intensive	No
[10]	Residual energy, packet forward rate, drop rate, modification rate	Hidden Markov model	MATLAB	Sinkhole, blackhole, and grey hole attacks	Advantages: Wide range of attack coverage Limitations: Trust indicators are attack specific instead of node specific. Hence, only applicable to detect considered attacks	No

(continued)

Table 1 (continued)

Reference	Trust indicators	Trust estimation approach	Simulator	Attacks addressed	Results	Node detection
[11]	No. of Successful and unsuccessful transactions	Direct trust evaluation-beta distribution Indirect trust evaluation-simple recommendation	MATLAB	Bad mouthing, on–off attack	Advantages: Simple trust evaluation model Limitations: No justification of successful and unsuccessful transactions is given Lacks to incorporate dynamicity of trust values	No
[12]	Data repetition rate, packet size abnormality, data correlation, volatility of transmission delay	Direct trust evaluation- Indirect trust evaluation-	NA	Malicious behavior	Limitations: Number of malicious nodes are less than the number of normal nodes	No
[13]	Transaction status, response time, device type	Trust evaluation based on weighted summation	NS-3	Malicious behavior	Advantages: introduction of device type module is effective in capturing real IoT environment Limitations: lacks to consider IoT-Specific attack type	No

(continued)

Table 1 (continued)

Reference	Trust indicators	Trust estimation approach	Simulator	Attacks addressed	Results	Node detection
[14]	Number of packets properly forwarded, number of packets dropped, number of packets falsely injected	Direct trust evaluation-beta distribution Indirect trust evaluation-ARMA/Garch based evaluation	NA	Address spoofing, routing table overflow, byzantine, sleep deprivation attack	Advantages: real-time predictive model is proposed Limitations: TI are specific to attack	No
[5]	Social features	Trust-based context and QoS aware technique	OMNET++	Sybil attack	Advantages: a reliable and light weight trust model is proposed Limitations: the integrity of produced results is not discussed	No
Proposed Model	Communication-specific SPR, SBR, Packet size	Fuzzy Logic based trust evaluation and node identification	BoT-IoT	DoS, DDoS	The fuzzy logic based trust evaluation and machine learning based malicious node identification model for DoS attack (common in IoT environment) is proposed	Yes

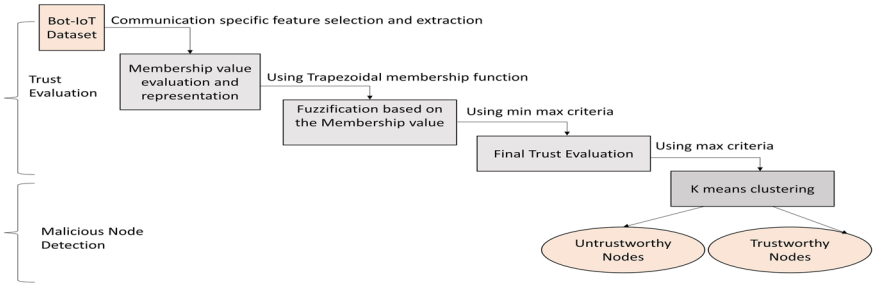


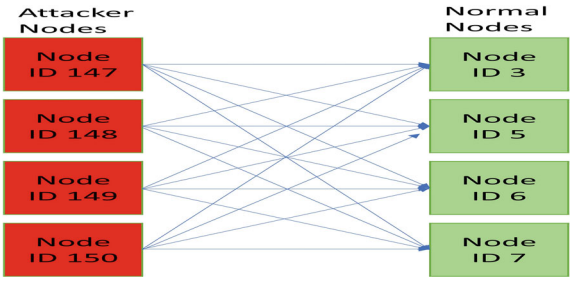
Fig. 1 Showing both the phases of study

environment. This open-source dataset is gathered from the UNSW-Canberra cyber data repository [15]. This dataset was created by UNSW Canberra's Research Cyber Range Laboratory. The environment comprises both normal and abnormal conditions. The network platform is made up of virtual machines controlled by the vSphere platform. The setup consists of four attacking nodes, four normal nodes, and one server. The dataset is IoT specific as it contains the log of nodes simulating IoT smart home services like a smart fridge, a weather station that generates temperature, humidity, and air pressure information, a motion detection lighting system, a smart garage door, and a thermostat. This IoT-specific log information differentiates this dataset from others such as UNSWNB15, NSL-KDD, and CIDDS-001. This dataset is used in this paper because we already know which node is attacking and which is normal. Hence, we can validate the findings by looking at the actual condition of nodes in the dataset.

Attacks like DoS and DDoS are considered both insider and outsider attacks. Hence, from the dataset, DoS (Denial of service attack), DDoS (Distributed Denial of Service), and normal communication log samples are selected for trust value evaluation of nodes. Further, the work can be extended to include more attack types during which the trust values of various nodes can be extracted and evaluated. This can be proved that the proposed model is not attack-dependent as compared to the proposed models that can accurately detect only certain types of attacks. The proposed model works well for DoS and DDoS attack data.

In the dataset, the communication patterns are as follows as shown in Fig. 2:

Fig. 2 Attacking architecture



- Attacker-Normal: In this scenario, the source node is attacking a normal node, this is labeled as anomaly in the dataset. This is the most dangerous communication pattern. Attacker wants to attack the normal node having high trust value. But based on the communication pattern, the trust value for the normal and attacking nodes is evaluated so that the normal node does not use these attacking nodes for further communication with nodes with small trust values.
- Normal-attacker: This scenario is also labeled as anomaly in the dataset. An anomaly detection model will only detect that it is an anomaly but can't detect which is the attacking node, so further communication with that node should be avoided to have successful and error-free communication.
- Normal-Normal: One very common communication which involves both highly trustworthy nodes in the network. The proposed model lets this communication to happen smoothly and swiftly.

The dataset is advantageous to consider as the nature of attackers and normal is already known. This helps our research to validate the results. After evaluating the trust values, the clustering is performed, and the results can be validated based on the given nature of node in the proposed research paper [16].

3.1.2 Data Preprocessing and Feature Extraction

In evaluating a node's trust values, the set of features are known as trust indicators (TI). A set of communication-specific features are selected and extracted for simplicity and applicability. All the selected features are node behavior-dependent rather than attack detection. With this set of features, improved performance of model is expected. The TI are defined and represented in Table 2.

Data Preprocessing

The raw dataset contains features that cannot be understood and processed in their present form. Hence, a preprocessing logic is to be applied to access the values without losing information. In this dataset, saddr and daddr columns represent IPv4 addresses of the form A:B:C:D. These are converted into node ID form by using mapping function in pandas library. Like,

Table 2 Feature set description

Feature name	Description	Feature type
saddr	Source IP address of the node	Selected
daddr	Destination IP address of the node	Selected
SPR(Sent packet Rate)	$= \frac{\text{number of packet sent from source to destination}}{\text{number of packets sent from destination to source}}$	Extracted
Packet size	$= \frac{\text{bytes}}{\text{packets}}$	Extracted
SBR(sent Byte Rate)	$= \frac{\text{number of bytes sent from source to destination}}{\text{number of bytes sent from destination to source}}$	Extracted

$$192 : 168 : 100 : 150 \rightarrow 150$$

This transformation from IPV4 IP address to a node ID is done for each source as well as Destination IP address in the dataset.

3.1.3 Membership Function

Fuzzy Logic is a knowledge representation rule where the information available cannot be grouped into crisp set of 0/1. Likewise, the trust of an entity based on its behavior cannot be fully groped into trustworthy or untrustworthy group. For these types of problems, fuzzy logic is used in this paper as compared to the min–max normalization algorithm [17], which only brings the value in [0–1] range but doesn't consider the feature's true meaning. Other work like [17], direct trust is evaluated using trust parameters, and all the values are combined using log function. The method is similar to average weighted method that is not robust and reliable.

In contrast to the abovementioned work, a membership function is used in this paper. Membership function is defined as the mathematical formula that shows how each point in the dataset belongs to a membership value. This value ranges between 0.0 and 1.0 having no membership to the set, 1 means having very high membership to the considered set. In this problem, the set considered is a set of trustworthiness. The membership value represents the belongingness to the trustworthiness set. These values are evaluated using a trapezoidal membership function. The function is defined by.

The range values like a, b, c, d, are decided by the research available in [18, 19]. This proves the validity of the considered ranges. For simplicity, the packet sizes lying between 50 and 250 are considered normal having values of membership 1 or near to 1. Whereas other values have values 0 or near to 0. The benefit of using membership values is that they represent the node's true behavior and give accurate and reliable trust values irrespective of normal or malicious communications as shown in Figs. 3 and 4.

3.1.4 Fuzzy Set Operations:

After the membership values of all three trust indicators have been evaluated, the max and min operations are considered to evaluate a node's final behavior and trust value.

$$\text{For each node ,} \quad \begin{matrix} V1 & V2 & V3 \\ V1 & V2 & V3 \\ \vdots & \vdots & \vdots \end{matrix} \quad \Longrightarrow \quad \begin{matrix} \mu 1 & \mu 2 & \mu 3 \\ \mu 1 & \mu 2 & \mu 3 \\ \vdots & \vdots & \vdots \end{matrix}$$

For a node's final trust value evaluation, min of all rows and maximum of all minimums are taken.

Like for each Transaction T_i, min($\mu 1$, $\mu 2$, $\mu 3$).

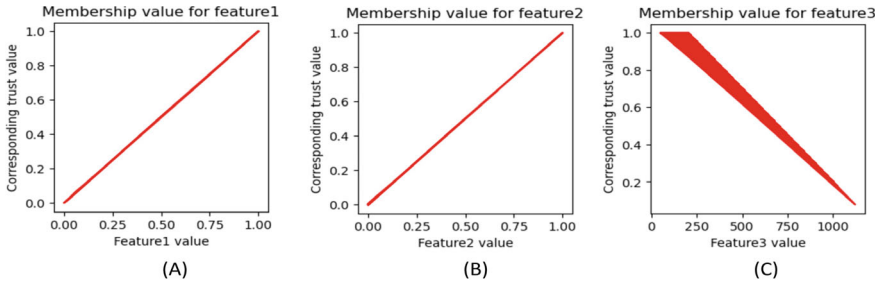


Fig. 3 (DoS Attack) The trust values according to the trapezoidal membership function, **a** represents feature 1 packet ratio and the membership values with a, b, c, d as 0, 1, 1, 2, respectively, **b** represents feature 2 bytes ratio with membership a, b, c, d, as 0, 1, 1, 2 respectively, **c** represents feature 2 packet size and the membership values with a, b, c, d as 0, 50, 200, 1200, respectively

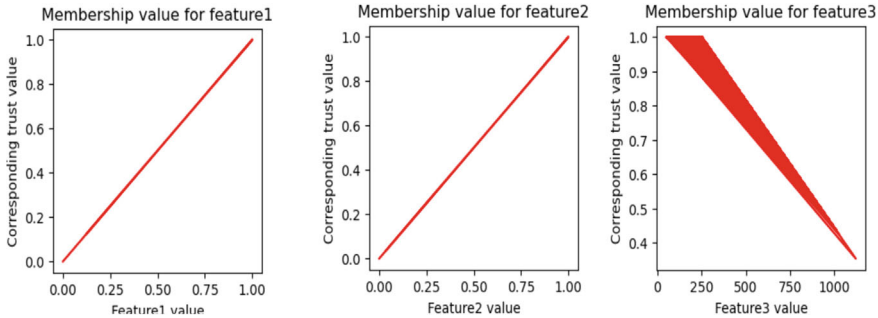


Fig. 4 (DDoS attack) The trust values according to the trapezoidal membership function, **a** represents feature 1 packet ratio and the membership values with a, b, c, d as 0, 1, 1, 2 respectively, **b** represents feature 2 bytes ratio with membership a, b, c, d as 0, 1, 1, 2 respectively, **c** represents feature 2 packet size and the membership values with a, b, c, d as 0, 50, 200, 1200, respectively

For each node, $N_{j,}$, Final Trust value(FTV) $= \max(\mu 1, \mu 2 \ldots\ldots\ldots\ldots \mu i)$.

3.1.5 Properties of the Evaluated Trust Values

Non-symmetric: This property is defined as if a node with ID 3 has a high trust value for a node with ID 5, then it is not necessary that 5 also trust node 3. This property should be satisfied for an efficient and accurate Trust Management Model, but in literature, these properties are not considered in trust value evaluation.

Dynamicity: The evaluated FTV are dynamic. It refers to the fact that it keeps changing with the nodes' changing behavior as shown in Figs. 5 and 6. Hence, it is able to capture the real trust values of the nodes which is rarely considered in the studies.

Case 1: DoS Attack

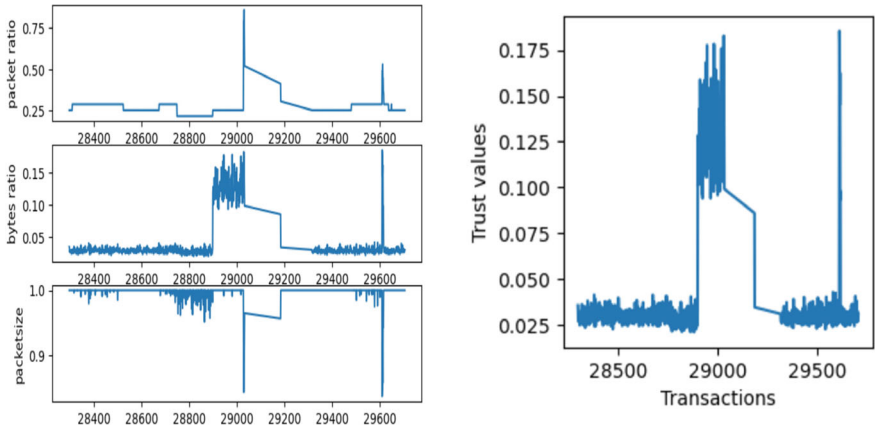


Fig. 5 **a** Showing the trust value distribution (Membership values) with respect to each feature for node 3 (Normal). **b** Trust values after applying min operation on the membership values for node 3 (Normal)

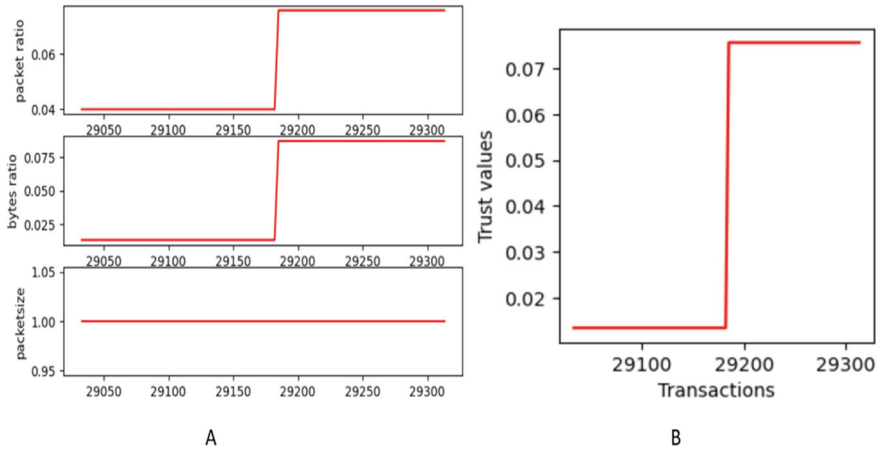


Fig. 6 **a** Showing the trust value distribution (Membership values) with respect to each feature for node 148 (Attack). **b** Trust values after applying min operation on the membership values for node 148 (Attack)

Case II: DDoS Attack.

Non-Transitive: This is defined as if node 3 trust node 5 and 5 trust 8. It does not imply node 3 also trust node 8. Using the trapezoidal membership function rule Figs. 7 and 8, the trust values evaluated are non-transitive, proving to be more reliable and accurate.

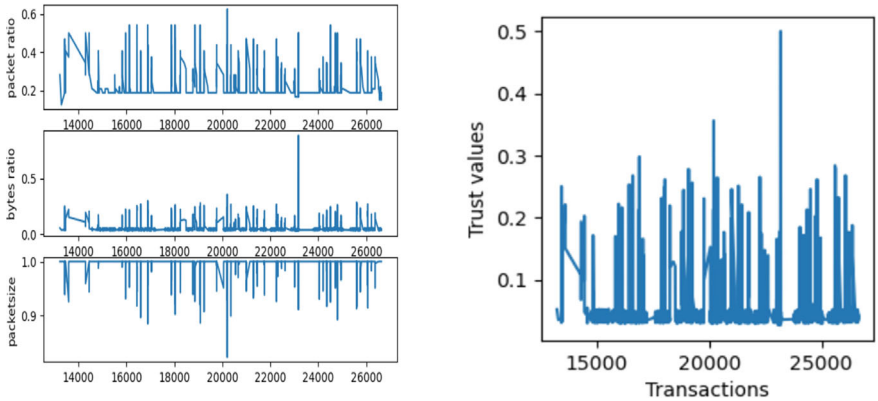


Fig. 7 **a** Showing the trust value distribution (Membership values) with respect to each feature for node 3 (Normal). **b** Trust values after applying min operation on the membership values for node 3 (Normal)

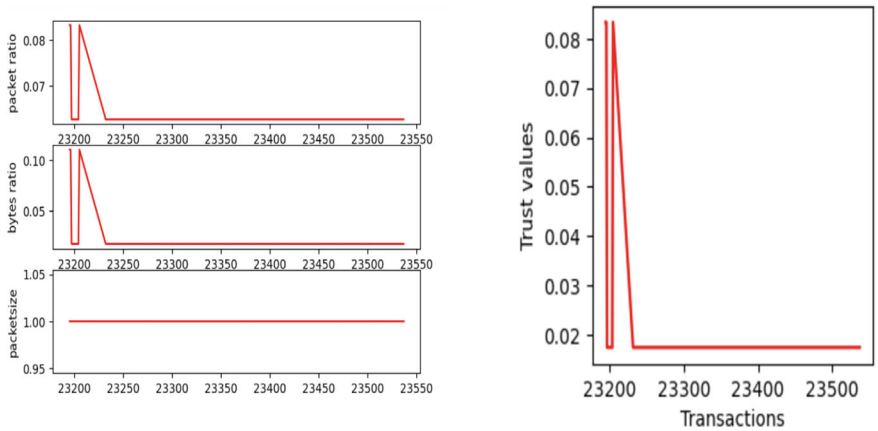


Fig. 8 **a** Showing the trust value distribution (Membership values) with respect to each feature for node 148 (Attack). **b** Trust values after applying min operation on the membership values for node 148 (Attack)

3.2 Phase 2: Malicious Node Detection

The malicious node detection is the process of labeling the nodes with 0/1, 0 being Untrustworthy and 1 being Trustworthy. In this paper, for labeling the nodes machine learning based K means clustering (K-Means Clustering) algorithm is used. The clustering algorithm uses IP address and FTV to cluster the nodes into these two groups (K = 2). The algorithm finds the centroid based on the dataset provided, and then the points Euclidean distance is calculated by using, X_i and X_j are two points in space,

$$D = \sqrt{\left((Xi - Xj)^2\right)}$$

Then, the distance value near to centroids is classified into untrustworthy and trustworthy nodes.

The reasons for selecting K Means clustering are:

1. This algorithm is straightforward and quick.
2. After a certain number of rounds, this method converges.
3. The algorithm works better with large datasets.

3.2.1 Detection Process

The main requirement of a malicious node detection system is accuracy, efficiency, and real-time detection. In this regard, traditional detection algorithms cannot meet these needs because they typically artificially adjust the algorithm in different network environments to establish a trust evaluation model appropriate for the current situation. Our goal is to create a countermeasure that adapts to changing surroundings in order to detect malicious nodes and reduce false positives. The results shown in Fig. 9 represent the model results after applying K means clustering to divide trustworthy nodes from trustworthy nodes. The method shows 97% accuracy, 98% recall, 0.98 F-Measure as shown by confusion matrix in Fig. 10.

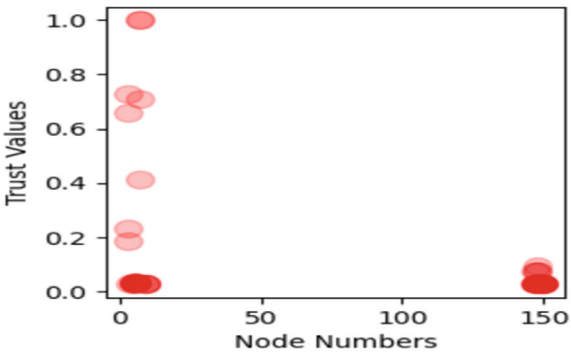


Fig. 9 Results after applying K Means Clustering approach (considering case1) on the dataset

	precision	recall	f1-score	support
0	1.00	0.95	0.98	22
1	0.95	1.00	0.97	19
accuracy			0.98	41
macro avg	0.97	0.98	0.98	41
weighted avg	0.98	0.98	0.98	41

Fig. 10 Confusion Matrix for Case 1

4 Conclusion

The security of communication in IoT environment is always on the stack because adversaries try to manipulate the nodes by showing themselves as an ordinary node, which is not true. Hence, there is a need for Trust Management System that is responsible for detecting malicious and non-malicious nodes. This paper proposes a Trust evaluation and malicious node detection model based on fuzzy logic and machine learning that can detect malicious nodes with 97.7% accuracy even when the number of normal and attacking nodes are equal. The proposed model is lightweight, robust, and efficient compared to existing methods. Also, it is visible from the results shown in the paper that the trust values are independent of attack types and the number of transactions. As DDoS and DoS attacks work differently, the trust values vary accordingly. The method can correctly detect trustworthy and untrustworthy nodes in the presence of any type of attack. For future work, more number of nodes with diverse range of attacks can be included. Also, the emotional trust parameter can be introduced in the trust indicator set for more reliable results in the future.

References

1. Thamilarasu G, Chawla S (2019) Towards deep-learning-driven intrusion detection for the Internet of Things. Sensors 19(9):1977. https://doi.org/10.3390/s19091977
2. Mahbooba B, Sahal R, Alosaimi W, Serrano M (2021) Trust in intrusion detection systems: an investigation of performance analysis for machine learning and deep learning models. Complexity 2021:1–23. https://doi.org/10.1155/2021/5538896
3. Thabit F, Can O, Aljahdali AO, Al-Gaphari GH, Alkhzaimi HA (2023) Cryptography algorithms for enhancing IoT security. Internet of Things 22:100759. https://doi.org/10.1016/j.iot.2023.100759
4. Mendoza CVL, Kleinschmidt JH (2015) Mitigating on-off attacks in the Internet of Things using a distributed trust management scheme. Int J Distrib Sens Netw 11(11):859731. https://doi.org/10.1155/2015/859731
5. Li J, Bai Y, Zaman N, Leung VCM (2017) A decentralized trustworthy context and QoS-aware service discovery framework for the Internet of Things. IEEE Access 5:19154–19166. https://doi.org/10.1109/ACCESS.2017.2756446

6. Bezerra EDC, Teles AS, Coutinho LR, da Silva e Silva FJ (2021) Dempster–Shafer theory for modeling and treating uncertainty in IoT applications based on complex event processing. Sensors 21(5):1863. https://doi.org/10.3390/s21051863
7. Feng R, Xu X, Zhou X, Wan J (2011) A trust evaluation algorithm for wireless sensor networks based on node behaviors and D-S evidence theory. Sensors 11(2):1345–1360. https://doi.org/10.3390/s110201345
8. Nangue A, Tagne EF, Tonye E (2021) Robust and accurate trust establishment scheme for wireless sensor network. Int J Comput Netw Inf Secur 12(6):14–29. https://doi.org/10.5815/ijcnis.2020.06.02
9. Ye Z, Wen T, Liu Z, Song X, Fu C (2017) An efficient dynamic trust evaluation model for wireless sensor networks. https://doi.org/10.1155/2017/7864671
10. Gamini Joshi VS, Hidden Markov trust for attenuation of selfish and malicious nodes in the IoT network
11. Che S, Feng R, Liang X, Wang X (2015) A lightweight trust management based on Bayesian and entropy for wireless sensor networks. Secur Commun Netw 8(2):168–175. https://doi.org/10.1002/sec.969
12. Yin X, Li S (2019) Trust evaluation model with entropy-based weight assignment for malicious node's detection in wireless sensor networks. EURASIP J Wirel Commun Netw 2019(1):198. https://doi.org/10.1186/s13638-019-1524-z
13. Sharma A, Pilli ES, Mazumdar AP (2017) Obviating capricious behavior in internet of things. In: 2017 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 480–486. https://doi.org/10.1109/ICACCI.2017.8125886
14. Alnumay W, Ghosh U, Chatterjee P (2019) A trust-based predictive model for mobile Ad Hoc network in Internet of Things. Sensors 19(6):1467. https://doi.org/10.3390/s19061467
15. https://research.unsw.edu.au/projects/bot-iot-dataset
16. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2019) Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Futur Gener Comput Syst 100:779–796. https://doi.org/10.1016/j.future.2019.05.041
17. Sharma A, Pilli ES, Mazumdar AP (2022) BD-Trust: behavioural and data trust management scheme for internet of things. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-022-03841-w
18. Doshi R, Apthorpe N, Feamster N (2018) Machine learning DDoS detection for consumer Internet of Things devices. In: 2018 IEEE security and privacy workshops (SPW). IEEE, pp 29–35. https://doi.org/10.1109/SPW.2018.00013
19. Tyagi H, Kumar R (2021) Attack and anomaly detection in IoT networks using supervised machine learning approaches. Revue d'Intelligence Artificielle 35(1):11–21. https://doi.org/10.18280/ria.350102

Classification, Designs, and Different Factors Involved in an Aquaponics System: A Review

Neelotpal Sharma, Yashaswini Trivedi, Varist Tripathi, Bhargav Dewri, Hanumant Singh Shekhawat, Rajamoni Neog, and Anamika Yadav

Abstract This paper summarizes all of the most recent research on aquaponics, a cutting-edge technique that grows plants using the wastewater from aquaculture. Numerous sustainability concerns are addressed with aquaponics, including pollution, soil depletion, water shortages, and rising fertilizer costs. The scientific and commercial applications of aquaponics are expanding; however, a thorough analysis of the scientific literature is still lacking to identify the optimal system type, the horticultural issues that will arise most frequently, and the direction that aquaponic research should take. Aquaponics is a method of farming fish and plants that incorporates several ideas, such as the effective use of fertilizer, water, and other biological and ecological principles. For the successful production of fish and plants, system water chemistry and the availability and balance of nutrients are essential. Different configurations of systems may be built, such as unconnected configurations (also known as decoupled systems) and completely recirculating designs. Aquaponics places a strong emphasis on the use of methods that have positive effects on the environment, economy, biology, chemistry, and technology.

Keywords Aquaponics · Hydroponics · Coupled aquaponics system · Decoupled aquaponics system

N. Sharma (✉) · H. S. Shekhawat · A. Yadav
IIT Guwahati, Guwahati, India
e-mail: neelotpal2021@rnd.iitg.ac.in

Y. Trivedi
Disha Delhi Public School, Kota, India

V. Tripathi · B. Dewri
Kendriya Vidyalaya IIT Guwahati, Guwahati, India

R. Neog
Vellore Institute of Technology, Vellore, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_36

1 Introduction

The field of agriculture must be transformed to tackle the problems arising due to the changes in the social demographic structure and its impact on the environment. Smart agriculture, resource recycling, and sharing, on the other hand, are not only crucial in agricultural transformation but also a key answer for governments around the world to encourage it [1]. Finding new, trustworthy, and sustainable methods of feeding the world's expanding population is crucial since it places more demand on natural resources like water, land, and nutrients. One such farming method that addresses environmental effects and gives farmers a chance to boost their revenue level is aquaponics. Aquaponics is a complementary method of growing plants and aquatic life. The aquaculture effluent is converted by microbes to provide nutrients for plant development. The plants are primarily responsible for absorbing nutrients and improving the quality of the water [2]. Aquaponics method of growing plants and aquatic animals at a time is still a relatively new agricultural technique, although its roots may be found in the work of the Aztec Indians, who began growing plants on rafts on a lake's surface around 1000 AD. Aquaculture and hydroponics are relatively new industries that have just started to gain traction. Even more recent is the notion of combining the two, which is only now feasible because of advancements in aquaculture and hydroponic growing methods. The management of aquaponics is simple and straightforward, as both hydroponics and re-circulatory aquaculture systems balance each other. The hydroponic system thus maintains water purity, while the excretory material generated by the fish satisfies nutritional requirements. It is a ground-breaking method for growing plants in which aquaculture effluent is purposefully diverted through plant beds in a closed system [3].

From the perspective of economics, the integration of two productive systems in aquaponic systems necessitates a significant investment. The marketing of fish and vegetables provides combined returns and decreases expenses related to management, increasing the earnings of the latter due to cost and return synergy [4]. From a social standpoint, an aquaponic system makes it easier to get secure agri-food items that are grown locally. This can improve the local community's quality of life while also contributing to the local economy [5]. Aquaponic systems can help prevent some of the main resource inefficiencies found in conventional agriculture; however, the resource efficiency described above has not been measured in scientific articles. A number of earlier reviews [6–8] state that effective aquaponic operations need to consider the effects of various factors, including system design, microbial populations, nitrogen levels, acceptable nutrient ranges, plant and fish species pairings, aeration and filtration technologies, pest management, and effective marketing. The people in charge of the system are primarily concerned with these issues. In this review paper the different types of aquaponic systems and their derivatives that can be advised for conserving our wetlands, as well as an insight into the technology's efficiency, will be discussed.

2 Literature Review

One of the types of integrated agri-aquaculture systems (IAAS) is known as aquaponics. However, aquaponics is much more closely related to combining aquaculture or hydroponic plant culture technologies with methods for fish rearing in tanks (such as recirculating aquaculture systems, or RAS). This integrated system, however, uses a wide range of aquatic animal and plant production technologies in a variety of situations [9]. Because of its relationship to aquaponics, which is typically defined as hydroponics/substrate culture combined with normal RAS aquaculture is "the process of raising fish (aquaculture) and growing soilless plants (hydroponics) with linked (coupled) or independent (decoupled) water circulation" [10]. This broad definition places a lot of focus on the integration of hardware, tools, or technologies while paying little to no attention to other process elements. The relevant term appears to be somewhat wide [13].

Over the years, aquaponics has incorporated a similar concept, with minor modifications. The largest definition has traditionally been supplied in Rakocy and his UVI team's scientific publications, stated as:

Aquaponics refers to a system where fish and plants are cultivated together in a closed recirculating environment [14, 15].

Due to changes and advances in the approach, Graber and Junge [16] broadened the concept as follows:

> A unique kind of recirculating aquaculture systems (RAS) is aquaponics, which is a polyculture made up of plants grown in the same water circle as the fish tanks (aquaculture) and fish tanks (aquaculture).

The definition of aquaponics has been a controversial topic in recent years. The decoupled aquaponics system, which aims to provide plants with additional fertilizers to expose them to an appropriate concentration of nutrients, has been a major area of development [17].

It's important to maintain the appropriate ratio of fish to plants in an aquaponics system. This is because the system relies on fish excreta to supply plants with nutrients as shown in Fig. 1, where the flow of nutrients from aquaculture waste (i.e., both fish excreta and fish food residue) travels and get converted into nitrates which becomes the source of nutrient for the plants. A system with only a single fish and a large area for plant cultivation cannot be considered an aquaponics system since the fish would not be able to provide enough nutrients for the plants to grow [18].

Until recently (for about the last 3–5 years), the term aquaponics was uniformly given to linked and completely recirculating system designs that strive to feed as much plant nourishment as possible from fish wastes [9].

Aquaponics is defined by ecological, water, and nutrient efficiency drivers and optimization; it is gradually broadening to include economic drivers [17, 19–21]. One advantage of this strategy is that the economic benefit of aquaponics technology is just as important as its ecological, technological, biological, and sustainable qualities. As such, the economic benefit should be taken into account when defining the technology as a whole [9].

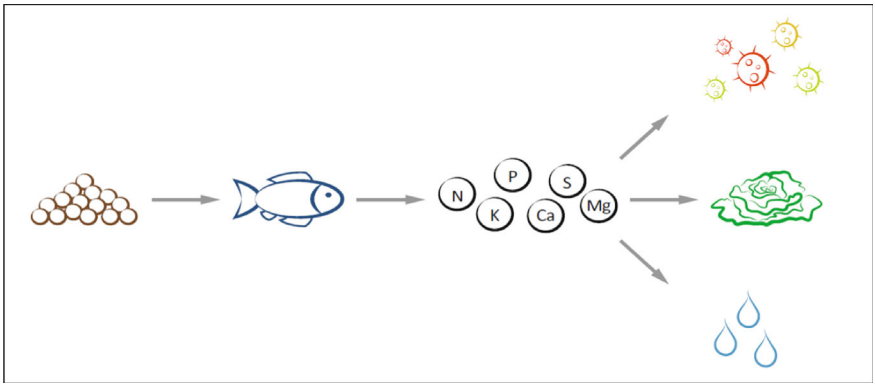


Fig. 1 An aquaponic system's nutrient flows are depicted in this diagram

3 Classification of Aquaponics System

3.1 Coupled Aquaponics System

The most basic type of aquaponics is called coupled aquaponics. A linked aquaponics system consists of one or more fish tanks, a hydroponic unit for plant cultivation, substrates for bacterial growth or suitable biofilters, and a sedimentation unit or clarifier as shown in Fig. 2. A closed water cycle is produced by the pipelines connecting these units. Aquaponic systems that are connected have several applications. Commercial aquaponics, home systems, and demonstration units (such as living walls connected) can benefit from the closed-loop approach [13]. The crop output in a connected aquaponic system is ultimately determined by the number of fish in the system. To accommodate the needs of the aquacultured species, the technical specifications of the fish-rearing unit must also be changed.

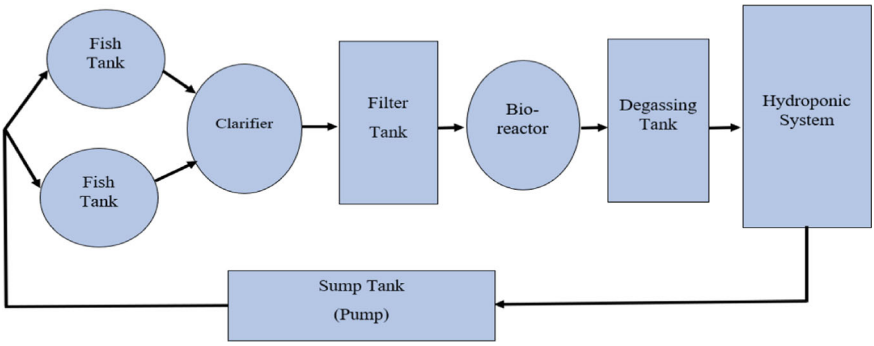


Fig.2 A diagram showing the flow water in a coupled aquaponics system [22]

The level of technical complexity involved in aquaponics is directly proportional to the size of production and the amount of water treatment required. This includes filtration, microbial control through the use of UV light, automated feeding, computerization, and biosecurity. In Germany, Naegel was the first person to introduce a coupled aquaponics system that combined fish and plant cultivation in a 2000 L size system (as shown in Fig. 3) within a controlled environment greenhouse [23] The technology was developed to evaluate the usage of nutrients from fish waste in completely regulated, water-recirculating environments intended for plant cultivation. Naegel processed wastewater via a dual sludge system that included aerobic and anaerobic processes. His idea was inspired by the open pond aquaponics system at an Experiment Station in the United States. The hydroponic cultivation of water chestnut was used there to eliminate excess nutrients from ponds containing channel catfish. Naegel combined nitrification and denitrification tanks to try to achieve complete oxidation of all nitrogenous materials and raise the nitrate content in his system. Iceberg lettuce, tomatoes, and carp grew effectively and generated harvestable output even though the system was only stocked at a low density of 20 kg/m^3.

Evidence suggests that the growth of aquatic animals and crops in a linked aquaponics system can be comparable to or even greater than that of standalone systems, such as hydroponics (vegetation) and aquaculture (aquatic animals). According to Rakocy, the yields of tilapia in linked aquaponics were somewhat higher (46.8 kg) than in solo fish culture (41.6 kg). Comparing Summer Bibb lettuce output

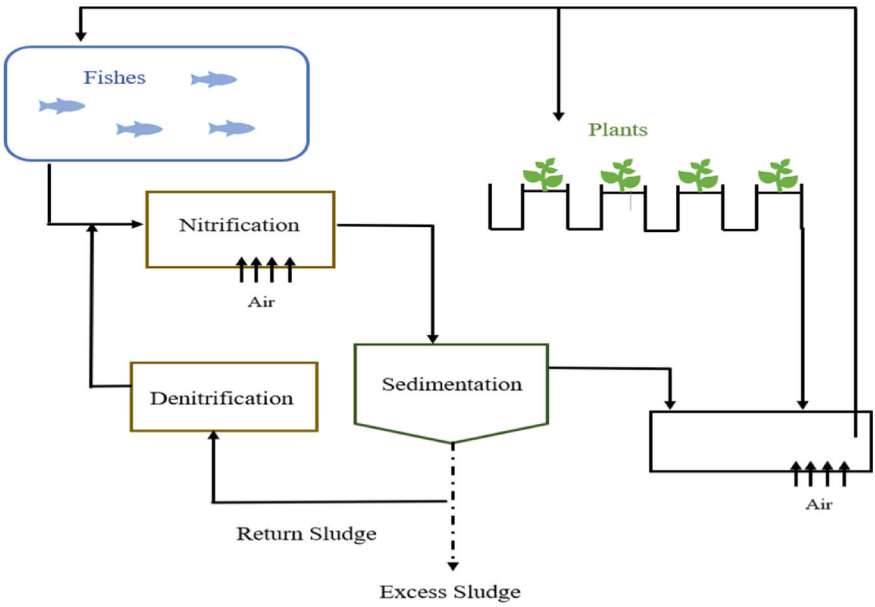


Fig. 3 Naegel's diagram of the first closed recirculation system

(385.1 kg) to vegetable hydroponic production (380.1 kg), there were also marginal gains [15].

Knaus found that compared to traditional hydroponics, which produces 2393 leaves, aquaponics produces greater plant leaf creation (3550 leaves), which can increase the biomass growth of *Oreum basilicum* [24]. Moreover, certain plants, such as lettuce, have been observed by Savidov to grow better in aquaponics systems than in conventional hydroponics. Cucumbers, and tomatoes, can absorb nutrients more quickly and thus flower earlier in aquaponics than in hydroponics [25].

3.2 Decoupled Aquaponics System

Fish, plants, and remineralization are all integrated into decoupled aquaponic systems (DAPS), which are autonomously controlled water cycles with discrete functions. The aim is to minimize the need for manipulation by designing and sizing components to achieve water loss through plant evapotranspiration on demand with tap water. In both units, this strategy produced the best species-specific water conditions and greater nutrient flow management. Water consumption is the most significant component of DAPS, primarily because of the evapotranspiration rate, which controls RAS water replacement and quality as well as plant nutrient delivery in the absence of extra fertilization. Therefore, figuring out DAPS's conceptual framework requires an understanding of the effects of water and nutrient flows within the system.

A DAPS including a hydroponic unit and a recirculating aquaculture system (RAS) built up as two independent water cycles was recently reported by Kloas et al. [20]. A one-way valve from the RAS was used to replenish the high degree of self-sufficiency across the system (Fig. 4).

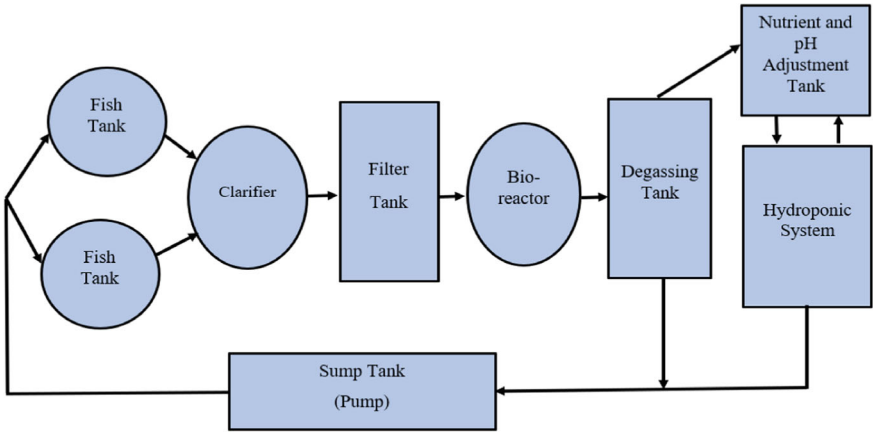


Fig. 4 A diagram showing the flow water in a decoupled aquaponics system [22]

In lab-scale experiments, decoupled aquaponics system shows better fish and plant growth. In comparison to a pure hydroponic control nutrient solution, Jijakli et al. 2016 observed a 39% increase in plant growth when additional fertilizer was added to the hydroponic component. Furthermore, anaerobic digestants promoted plant growth, as Goddek showed.

4 Hydroponic Components in Aquaponics System

4.1 Deep Water Culture (DWC)

As stated by Vermeulen and Kamstra [26], in a deep water culture system, plants float on the water in foam plates at a depth of around 5 cm and above. The water is kept in circulation or otherwise held in situ for crop production. DWC is also known in various places as the floating raft technique. Commercial growers favour DWC because of its high root-to-water contact, minimal maintenance needs, and capacity to produce a huge number of plants with few resources [27]. A DWC system is said to use 1% of the system's water supply per day, according to a researcher. It's astonishing that no other hydroponic system has been shown to have as high water use efficiency, especially in light of the widespread marketing of generalized aquaponics systems with water use efficiencies comparable to DWC.

4.2 Nutrient Film Technique (NFT)

Nutrient film systems are only suitable for growing light leafy green vegetables and fish. Plants are cultivated in long, thin channels or pipes using the Nutrient Film Technique (NFT). Perforated plastic glass/cups are put in small holes drilled into the pipe. Plants are cultivated in perforated glasses that allow water and nutrients to reach their roots. A study was conducted to compare the effectiveness of three different aquaponic systems: Deep Water Culture (DWC), Media Culture, and Nutrient Film Technique (NFT). According to the study, NFT produced the least amount of lettuce and was 20% less effective than DWC in removing nitrate. This could be because, in contrast to DWC, where all roots are submerged in water, there are fewer roots in NFT that are in touch with the aquaponic water [30]. Comparing NFT systems to media culture and DWC systems, research investigations found that NFT systems were less popular and had poorer returns. Furthermore, NFT requires the usage of a biofilter since it has a reduced surface area for beneficial bacteria [31].

4.3 Media-Filled Grow Beds

Inert solids like hydroton, perlite, and cocopeat are used in media-filled grow beds culture, a hydroponic technique, to cultivate plants. Because it works well for a range of plant species and small-scale study setups, this system is frequently utilized in aquaponic research publications. A benefit of hydroponics based on media is that the substrates offer sufficient surface area for both physical filtration and nitrification of bacterial growth, hence obviating the requirement for a biofilter [31, 32]. Furthermore, medium beds are a better choice for bigger plants since they provide greater stability for root growth. But in a hydroponic grow environment, the substrates nearest to the first water inflow are more likely to become clogged with debris, which can result in uneven fertigation or even anaerobic zones. This may be fixed by flushing the system with purified water or by physically removing, cleaning, and replacing the grow bed media. However, the cost of production may increase due to this maintenance procedure. Larger-scale aquaponic operations are more suited for DWC and other low-maintenance hydroponic components, whereas smaller-scale operations benefit more from media culture [13].

4.4 Other Hydroponic Setups like Ebb and Flow and Drip Irrigation

Aquaponics systems commonly employ NFT and DWC hydroponic components, but other, less common methods such drip irrigation, vertical towers or walls, Ebb and Flow, or flood and drain tables, can also work well under certain situations. Remarkably, tomato plants cultivated with drip irrigation produced the highest yield compared to NFT and DWC hydroponic systems. Aquaponic articles don't usually address these systems [32]. The research found that the advantages of ebb and flow include better aeration with less energy usage, more adaptability during periods of saturation and aridity, little infrastructure, and more surface area available for microbial growth [33]. However, these methods have some drawbacks and consumes a lot more energy in a drip irrigation system than in a DWC or NFT [32]. Considering the varying rates of flooding the tables, a further problem in the ebb and flow process is getting filtered water to the fish tanks at a consistent rate.

5 Water Resources and Quality Requirements

The water used in the aquaponics system is shared by the two main components of an aquaponics system—the fish and the plants—it serves as the primary medium for nutrient delivery in the system thus creating the general chemical environment in

which both the fish and the plants are cultivated. Therefore, it's imperative to ascertain the water's quality—or, more accurately, its chemistry—before subjecting fish or plants to it. Water types that may be utilized are rainwater, municipal water, well water, and reverse osmosis (RO) water. Since surface water may include excessive amounts of sediments, unwanted fish or insect larvae, aquatic vegetation, such as algae, and possibly hazardous bacteria, it should not be utilized [34]. Micronutrients like calcium, salt, or iron that are detrimental to fish but beneficial to plants at very small concentrations can be found in well water. Thus to maintain the overall aquaponics setup equilibrium, a water supply free of germs, salts, nutrients, and other pollutants is required.

Commercial aquaponics production requires optimization since commercial success (i.e., financial profitability) can only be achieved through optimization. As a result, the aquaponic system's water chemistry and water quality standards are critical to the business's eventual commercial and economic success [35].

The five main aspects of water quality in aquaponics are temperature, hardness, pH, dissolved oxygen (DO), and total nitrogen concentrations. While less usually addressed by farmers, alkalinity, carbon dioxide, settleable solids, and suspended particles are all as significant. Knowing how each characteristic impacts the well-being and productivity of fish, plants, and microorganisms is crucial. The appropriate values for these components vary between fish, plants, and microorganisms. To fulfill the demands of all species simultaneously in an aquaponic system, compromises are made for different water quality parameters.

Table 1, illustrates the comprises of ideal aquaponics that are required for the key water quality parameters. pH and temperature are the two most crucial parameters to balance. It is suggested that the pH should be kept at a compromised level of 6–7, which is somewhat acidic.

The overall objective is to preserve a healthy ecosystem with characteristics of water quality that support the growth of fish, vegetables, and bacteria all at the same time. There will be instances where adjusting the water quality actively is necessary to meet these standards and maintain system functionality.

Table 1 Ideal parameters required to maintain the equilibrium in an aquaponics system [34]

Ideal parameters	Range
Dissolved oxygen (DO)	5–8 ppm
pH	6–7
Water hardness	60–140 ppm
Water temperature	64–86 °F
Ammonia (NH3)	0 ppm
Nitrite (N02-)	0 ppm
Nitrate (N03-)	5–150 ppm

6 Nutrient Sources

The primary input in any aquaponic system is nutrients since these systems are made to effectively distribute the nutrients that are added to them to the three major living forms that are present: the microflora, which helps the fish and plants absorb the extra nutrients, and the fish and plants, which are the system's primary products [9].

Using fish feed, the primary source of nutritional input, as effectively as possible is one of the key design principles of classic, fully recirculating aquaponic systems. The fish are supplied the fish feed, which they then eat, digest, and use the nutrients as needed, creating a waste stream that includes both solid and dissolved waste. Fish meal becomes the primary source of nourishment for the plants, while the fish waste stream becomes the primary source of fertilizer. The remaining nutrients are supplied using a nutrient augmentation approach that fulfils the dual objective of giving the additional nutrients and supplying the nutrients that the fish meal lacks, such as iron chelate, potassium, and calcium, while also managing the system aquatic pH. This dual-duty method is known as "buffering", and the supplement is also known as "buffer" [36].

By emulating and comparing the nutrient mixes and strengths utilized in substrate culture and conventional hydroponics, decoupled designs seek to maximize plant development in an aquatic environment. In contrast to previous designs, they significantly supplement with external nutrients to reach the appropriate plant growth rates rather than depending only on the nutrients from fish feed [12].

An alternative for supplying fertilizers to aquaponic systems is external nutrient replenishment. For this, foliar plant sprays can be used. These sprays distribute hydroponic fertilizer salts or derivatives in an aquatic medium. In contrast to applying foliar sprays, which administer dissolved nutrient salts directly to the plant leaves, adding nutrient salts directly to the culture water allows the plants to absorb the nutrients by root absorption. When applying foliar sprays, stomatal or cuticle access is used to facilitate absorption [37–39].

7 Integration of Sensors and Artificial Intelligence (AI) in an Aquaponics System

Sensors play a vital role in creating an aquaponics system smart and efficient. Several sensors can be used [44] as shown in Fig. 5, which in turn produces a large database that can be integrated with AI/ML technology.

Artificial intelligence (AI) in aquaponics can boost efficiency, use resources wisely, and increase overall productivity. AI can be integrated into an aquaponics system by managing the sensors and controllers for instant checks on water quality, fish activity, and plant health. It uses machine learning to see patterns and weird parts in the data. So, it adjusts things to save resources and make conditions just right for growth.

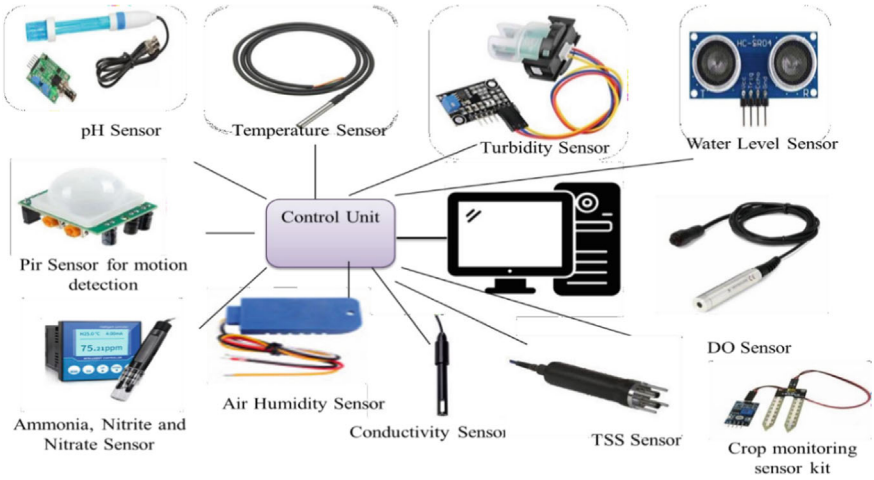


Fig. 5 Types of sensors used in aquaculture ponds

The integration of AI will check old data and current conditions to guess future plant growth, fish wellness, and harvest sizes. Farmers will be able to predict problems coming by using this. They can make appropriate plans early to get the most out of their farm.

AI also plays an important role in farming and decision-making. It can be used for the alteration of things like nutrient levels, water speed, and light times to use resources best and increase crop production. AI also lends a hand in deciding when to plant, how many to plant, and when to harvest, using predictability and real-time facts. Using AI, we can also fight against plant and fish diseases. It spots signs of sickness, lack of nutrients, and bugs in plants and fish using computer visuals. Early tattle-telling helps to stop diseases from spreading and keeps more crops safe.

Another vital role that AI plays is to control the energy consumption. It controls lights, heat, and pumps. It works based on what's needed and weather conditions. This means less energy is wasted. AI in aquaponics systems helps us be sustainable. Also, the remote monitoring and management of the aquaponics farm by the integration of AI will help farmers to stay one step ahead in making correct decisions and decreasing labour cost. This makes their system better, smarter, and more efficient over time.

AI techniques were found to be useful in the reviewed literature for visualizing plant growth stages, predicting fish size, identifying anomalies, identifying nutrient deficits and plant illnesses, and optimizing processes. The common uses of AI and ML in aquaponics are covered in this section. Essentially, it is intended to provide an overview of current research topics rather than covering all AI and ML research done in the aquaponics industry. Table 2 displays the AI techniques applied in the examined studies.

Table 2 Different AI method usage in an aquaponics system

Usage	Methods of artificial intelligence	Authors
For determining the amount of dissolved oxygen	Deep convolutional neural, K-nearest neighbour (KNN), Support, decision tree (DT) LSTM, CNN, CNN-LSTM	Taha et al. [40]; Barzegar et al. [41]
For estimating the growth of plants	Artificial neural network (ANN), Quantum support vector machine (QSVM)	Concepcion II et al. [42]
For the identification of deficiencies in nutrition	DCNN, KNN, SVM, DT	Taha et al. [40]
For detecting various disease	YOLOv5s, Fast-RCNN	Abbasi et al. [43]

8 AEQUA System at IIT Guwahati and Its Implication for the Self-Sustainability of the SC Community of Central and Lower Assam

Under the Science for Equity Empowerment and Development (SEED) Division of the Department of Science and Technology (DST), a three-year project was sanctioned to the EEE department, IIT Guwahati for the establishment of an AI-enabled advanced aquaponics (AEQUA) ecosystem for the self-reliance of the SC community in Central and Lower Assam. A detailed AI integrated aquaponics system has been set up to make it a start-up that will eventually support communities and households to implement commercially AI-Enabled Advanced Aquaponics Farming Ecosystems in their lands and support them in selling their high-value yields. Based on resource assessment inputs and selection of beneficiaries, an optimal layout is being planned for construction, fabrication, and installation of medium-scale AI-Enabled Advanced Aquaponics (AEQUA) Systems with all vital components as shown in Fig. 6.

In addition to that, for greener energy consumption, the installation of a hybrid power plant with a Solar Power Module, EB Supply, Inverter, and Battery Bank is designed to ensure 24 × 7 power supply to the AEQUA Systems and has been done concurrently with all strategically dimensioned components as shown in the Fig. 7.

Machine learning is proposed to be employed in this project for mainly predictive analysis and implementation. All data will be collected from the AEQUA setups in mainly three categories; initial setup, crop growth cycle, and yield and cash. The features, that will measure the initial setup phase, are initial setup time, cost, person-hours, the learning curve of the farmers, weather, temperature, soil conditioning, etc. The crop growth phase will be measured by crop growth rate, temperature, irrigation cycle, sunlight, etc. The yield and cash phases will be measured mainly in terms of total investment and the revenue generated through utilisation of AEQUA systems and cloud platform.

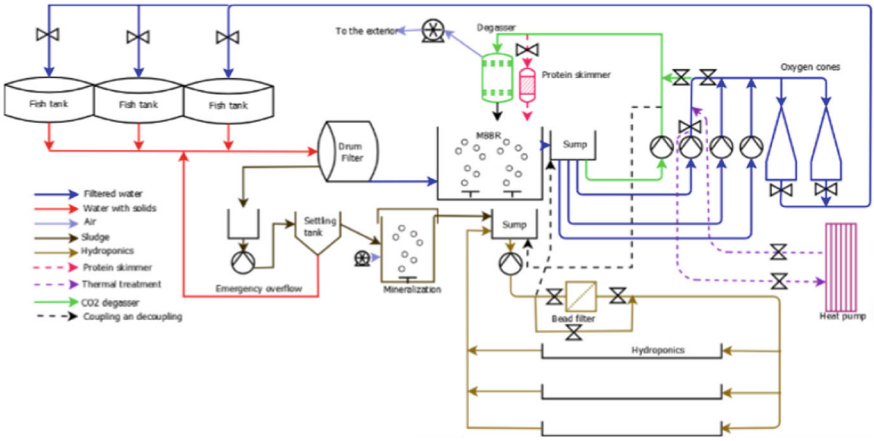


Fig. 6 AEQUA layout design with integration of sensors and implementation of artificial intelligence

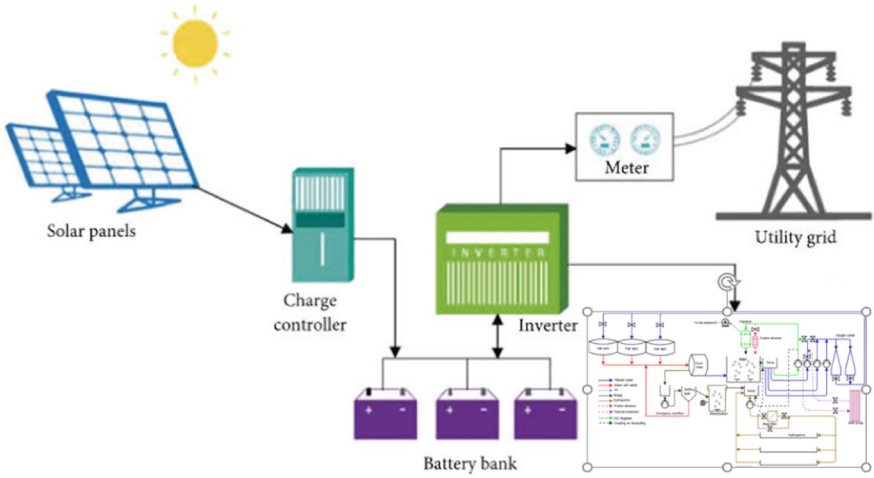


Fig. 7 Hybrid power plant with a solar power module for power supply to the AEQUA system

In addition to this, machine learning will be incorporated in optimizing various logistic issues related to the project, such as optimal route selection for collection of produce from various locations, supply-chain management, demand-response of the produce, user/customer specific response, etc. There will also be an endeavour to employ blockchain in the reliable and authentic distribution of the produce.

9 Conclusion

A possible answer to many agricultural sustainability problems is the use of aquaponic systems. They have the capacity to greatly reduce aquaculture's nutrient outflow and water waste while producing excellent yields with no additional fertilizer input. The rising amount of research papers on aquaponics that have been published in recent years indicates that people are becoming more interested in sustainable agricultural systems like aquaponics. These latest discoveries have led to the identification of clear patterns, consequences, and needs for more research on system design, as well as alternatives to aquaponics such as maraponics, haloponics, digeponics, and vermiponics; hydroponic components, and ideal fish, plant, and microbial species. Additionally, in terms of aquaculture, tilapia, carp, and African catfish are often the most productive fish species in aquaponic systems. Because of their ability to withstand high nitrate and low oxygen levels, these fish are successful. There has not been a study that investigates the advantages of polyculture in aquaponic systems. Given that polyculture may provide a more nutrient-balanced approach to plant growth, future research in this area could be extremely beneficial.

Acknowledgements The authors would like to acknowledge IIT Guwahati and the DST-sponsored project entitled, "AI-enabled advanced aquaponics ecosystem for the self-reliance of SC community in central and lower Assam".

References

1. Wang SC, Lin WL, Hsieh CH (2020) To improve the production of agricultural using IoT-based aquaponics system. Int J Appl Sci Eng 17:207–222. https://doi.org/10.6703/IJASE.202005_17(2).207
2. Yep B, Zheng Y (2019) Aquaponic trends and challenges—a review. J Clean Prod 228:1586–1599. https://doi.org/10.1016/j.jclepro.2019.04.290
3. Kumar Sharma P, Stephan Sampath Kumar J, Pawan Kumar Sharma C, Kumar Sharma P, Stephan Sampath Kumar J, Anand S (2018) Aquaponics: a boon for income generation in water-deficient areas of India like Rajasthan. Int J Fisheries and Aquatic Stud 6:170–3
4. Rupasinghe JW, Kennedy JOS (2010) Economic benefits of integrating a hydroponic-lettuce system into a barramundi fish production system. Aquac Econ Manag 14:81–96. https://doi.org/10.1080/13657301003776631
5. Wirza R, Nazir S (2021) Urban aquaponics farming and cities—a systematic literature review. Rev Environ Health 36:47–61. https://doi.org/10.1515/reveh-2020-0064
6. Palm HW, Knaus U, Appelbaum S, Goddek S, Strauch SM, Vermeulen T et al (2018) Towards commercial aquaponics: a review of systems, designs, scales, and nomenclature. Aquacult Int 26:813–842. https://doi.org/10.1007/s10499-018-0249-z
7. Tyson R v, Treadwell DD, Simonne EH (n.d.) Opportunities and challenges to sustainability in aquaponic systems
8. Endut A, Jusoh A, Ali N, Wan Nik WB, Hassan A (2010) A study on the optimal hydraulic loading rate and plant ratios in recirculation aquaponic system. Biores Technol 101:1511–1517. https://doi.org/10.1016/j.biortech.2009.09.040
9. Wilson L (n.d.) Commercial aquaponic systems: integrating recirculating fish culture with hydroponic plant production

10. Knaus U, Palm HW (2017) Effects of the fish species choice on vegetables in aquaponics under spring-summer conditions in northern Germany (Mecklenburg Western Pomerania). Aquaculture 473:62–73. https://doi.org/10.1016/j.aquaculture.2017.01.020
11. Cerozi BS, Fitzsimmons K (2017) Phosphorus dynamics modeling and mass balance in an aquaponics system. Agric Syst 153:94–100. https://doi.org/10.1016/j.agsy.2017.01.020
12. Delaide B, Goddek S, Gott J, Soyeurt H, Jijakli MH (2016) Lettuce (*Lactuca sativa L. var. Sucrine*) growth performance in complemented aquaponic solution outperforms hydroponics. Water (Switzerland) 8. https://doi.org/10.3390/w8100467
13. Palm HW, Knaus U, Appelbaum S, Goddek S, Strauch SM, Vermeulen T et al (2018) Towards commercial aquaponics: a review of systems, designs, scales and nomenclature. Aquacult Int 26:813–842. https://doi.org/10.1007/s10499-018-0249-z
14. Rakocy JE, Shultz RC, Bailey DS, Thoman ES (n.d.) Aquaponic production of Tilapia and Basil: comparing a batch and staggered cropping system
15. Rakocy JE, Bailey DS, Shultz RC, Thoman ES (n.d.) Update on tilapia and vegetable production in the uvi aquaponic system
16. Graber A, Junge R (2009) Aquaponic systems: nutrient recycling from fish wastewater by vegetable production. Desalination 246:147–156. https://doi.org/10.1016/j.desal.2008.03.048
17. Goddek S (n.d.) Opportunities and challenges of multi-loop aquaponic systems
18. Lennard W, Goddek S (2019) Aquaponics: the basics. Aquaponics food production systems, Springer International Publishing, pp 113–43. https://doi.org/10.1007/978-3-030-15943-6_5
19. Goddek S, Keesman KJ (2018) The necessity of desalination technology for designing and sizing multi-loop aquaponics systems. Desalination 428:76–85. https://doi.org/10.1016/j.desal.2017.11.024
20. Kloas W, Groß R, Baganz D, Graupner J, Monsees H, Schmidt U et al (2015) A new concept for aquaponic systems to improve sustainability, increase productivity, and reduce environmental impacts. Aquac Environ Interact 7:179–192. https://doi.org/10.3354/aei00146
21. Yogev U, Barnes A, Gross A (2016) Nutrients and energy balance analysis for a conceptual model of a three loops off grid, aquaponics. Water (Switzerland) 8. https://doi.org/10.3390/w8120589
22. Rakocy JE (2012) Aquaponics-integrating fish and plant culture
23. Naegel LCA (1977) Combined production of fish and plants in recirculating water. vol 10
24. Palm HW, Knaus U, Appelbaum S, Strauch SM, Kotzen B (2019) Coupled aquaponics systems. aquaponics food production systems, Springer International Publishing, pp 163–99. https://doi.org/10.1007/978-3-030-15943-6_7
25. Evaluation-of-Aquaponics-Technology-in-Alberta (n.d.)
26. Vermeulen T, Kamstra A (n.d.) The need for systems design for robust aquaponic systems in the urban environment
27. Iulia-Adriana M, Maria B (2015) Analysis of aquaponic organic hydroponics from the perspective of setting costs and of maintenance on substratum and floating shelves systems. vol 19
28. Love DC, Uhl MS, Genello L (2015) Energy and water use of a small-scale raft aquaponics system in Baltimore, Maryland, United States. Aquacul Eng 68:19–27. https://doi.org/10.1016/j.aquaeng.2015.07.003
29. Pattillo DA (n.d.) Technical bulletin series an overview of aquaponic systems: hydroponic components
30. Lennard WA, Leonard Bv (2006) A comparison of three different hydroponic sub-systems (gravel bed, floating and nutrient film technique) in an Aquaponic test system. Aquacul Int 14:539–50. https://doi.org/10.1007/s10499-006-9053-2
31. Maucieri C, Nicoletto C, Junge R, Schmautz Z, Sambo P, Borin M (2018) Hydroponic systems and water management in aquaponics: a review. Ital J Agron 13:1–11. https://doi.org/10.4081/ija.2017.1012
32. Schmautz Z, Loeu F, Liebisch F, Graber A, Mathis A, Bulc TG, et al. (2016) Tomato productivity and quality in aquaponics: comparison of three hydroponic methods. Water (Switzerland) 8. https://doi.org/10.3390/w8110533

33. Palm HW, Seidemann R, Wehofsky S, Knaus U (2014) AACL BIOFLUX significant factors affecting the economic sustainability of closed aquaponic systems. Part I: system design, chemo-physical parameters and general aspects. vol 7
34. Masabni J, Sink T (n.d.) Water quality in aquaponics. Associate professor and extension vegetable specialist 2 associate professor and aquaculture extension specialist
35. Goddek S, Delaide B, Mankasingh U, Ragnarsdottir KV, Jijakli H, Thorarinsdottir R (2015) Challenges of sustainable and commercial aquaponics. Sustainability (Switzerland) 7:4199–4224. https://doi.org/10.3390/su7044199
36. Rakocy JE, Masser MP, Losordo TM (n.d.) Recirculating aquaculture tank production systems: aquaponics-integrating fish and plant culture
37. Roosta HR, Hamidpour M (2011) Effects of foliar application of some macro- and micro-nutrients on tomato plants in aquaponic and hydroponic systems. Sci Hortic 129:396–402. https://doi.org/10.1016/j.scienta.2011.04.006
38. Roosta HR, Hamidpour M (2013) Mineral nutrient content of tomato plants in aquaponic and hydroponic systems: effect of foliar application of some macro- and micro-nutrients. J Plant Nutr 36:2070–2083. https://doi.org/10.1080/01904167.2013.821707
39. Roosta HR (2014) Effects of foliar spray of K on mint, radish, parsley and coriander plants in aquaponics system. J Plant Nutr 37:2236–2254. https://doi.org/10.1080/01904167.2014.920385
40. Taha M, ElMasry G, Gouda M, Zhou L, Liang N, Abdalla A, Rousseau D, Qiu Z (2022) Recent advances of smart systems and internet of things (IoT) for aquaponics automation: a comprehensive overview. Chemosensors 10:303
41. Barzegar R, Aalami MT, Adamowski J (2020) Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. Stoch Environ Res Risk Assess 34:415–433
42. Concepcion RS II, Alejandrino JD, Lauguico SC, Tobias RR, Sybingco E, Dadios EP, Bandala AA (2020) Lettuce growth stage identification based on phytomorphological variations using coupled color superpixels and multifold watershed transformation. J Adv Intell Inform 6:261
43. Abbasi R, Martinez P, Ahmad R (2022) An ontology model to represent aquaponics 4.0 system's knowledge. Inf Process Agric 9:514–532
44. Yadav A, Noori MT, Biswas A, Min B (2023) A concise review on the recent developments in the internet of things (iot)-based smart aquaculture practices. Rev Fisheries Sci Aquacul 31(1):103–118

An Improved Medical Entity Extraction Method from Annotated Records

Priti Bhardwaj, Nonita Sharma, and Niyati Baliyan

Abstract A significant percentage of the information within Electronic Health Records (EHR) consists of unorganized language that contains information regarding clinical events. The extraction of this information is necessary to facilitate subsequent analysis and application in both everyday healthcare settings and research endeavors. A key issue in Natural Language Processing (NLP) is clinical Named Entity Recognition (NER), which entails identifying and capturing key ideas (named items) from medical narratives. It identifies major concepts like frequency, dosage, duration, etc. from the medical text. In this work, the i2b2 2018 track2 dataset has been annotated to assign entity labels to individual tokens in a phrase, therefore increasing the quantity of annotated data and enhancing the process of extracting entities. A comparison of extant techniques for the automated extraction of clinical entities from medical records is presented in this article. The experimental findings demonstrate that our model outperforms state-of-the-art by an accuracy improvement of 2.35% with Bidirectional Encoder Representations from Transformers (BERT) based techniques.

Keywords Natural language processing (NLP) · Named entity recognition (NER) · Electronic health records (EHR) · Adverse events · Machine learning · Bidirectional encoder representations from transformers (BERT)

P. Bhardwaj (✉) · N. Sharma
IT Department, IGDTUW, New Delhi, India
e-mail: priti001phd18@igdtuw.ac.in

N. Sharma
e-mail: nonitasharma@igdtuw.ac.in

N. Baliyan
CE Department, NITKKR, Kurukshetra, Haryana, India
e-mail: niyatibaliyan@nitkkr.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_37

1 Introduction

Presently, data from many areas of our business, administration, and personal and professional generally appears in its raw state, which is unstructured and imprecise. This makes it challenging for individuals to summarize, investigate, draw inferences, and do mathematical computations. To achieve the aforementioned objectives, we use several NLP operations on the text. Over 80% of presently accessible healthcare data is reportedly comprised of unstructured data [1].

EHRs emerged as invaluable repositories of data for healthcare providers in their current patient care practices [2]. Healthcare reports can be transformed into structured health data, which can provide important new information for bioinformatics research [3]. The majority of the information contained in EHRs consists of unstructured data, rendering them resistant to statistical analysis. Due to the time-consuming, error-prone, non-scalable, and costly nature of manual abstraction, NLP approaches are increasingly being utilized in the biological and clinical domains [4]. NER is a fundamental component of all NLP systems. It is classified as a sub-task within the field of information retrieval. Named items, such as illnesses, medications, and diagnostic tests, mentioned in medical entries are detected using NER.

Biomedical NER [5], wherein we try to identify physiological factors in given documents, like medications, diseases, and enzymes. The capacity to recognize biological objects effectively is required for acquiring biomedical knowledge stored in unstructured texts and transforming it into structured representations. A set of significant medical features for 40 medical specialties have been extracted in [6] by multiclass classification of medical records.

This paper seeks to evaluate and compare the performance of BERT-based NER in entity extraction, building upon earlier research. For this study, we utilized the i2b2 NLP challenge dataset for medical extraction elements. The suggested work's principal contributions are:

- Annotation of i2b2 2018 track2 dataset having a focus on extracting coupled entities.
- Proposal of fine-tuned entity tagging model based on the BERT model.
- Comparison of performances of three NER models on the well-known clinical event recognition dataset.

This work is organized into the following areas. The document's Sect. 2 comprises studies that are pertinent to the topic at hand. Subsequently, Sect. 3 provide a clear description of the recommended pre-trained model, the dataset, and the experimental circumstances. This continues with an analysis of the outcomes. In Sect. 4, we finally conclude our work.

2 Related Work

Presently, NLP is a subject of considerable scholarly attention owing to its pragmatic implementation and efficacy across diverse domains. Text processing utilizes NLP methods to derive vital data from textual input. Numerous sophisticated text-mining techniques [7–9] have emerged. Medical text analysis has seen a notable increase in the utilization of NLP methods [10–13], Word sense disambiguation [14], NER, adverse drug events (ADE) detection, collecting data, and relation extraction.

The most pertinent tools, datasets, and deep-learning techniques—such as transfer learning, Bidirectional Long Short-Term Memory (BI-LSTM), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and many others reviewed in [15]. These techniques are currently being applied in modern NER problem environments and their applications. The authors conducted an assessment of the existing obstacles and prospective avenues.

The goal of this study [16] is to extract ADE mentions from medical text data and investigate any possible relationships between them. A novel approach has been introduced for extracting ADE using multi-job transfer learning. On the n2c2 2018 dataset, the model shows a total accuracy of 93.65%, and on the TAC 2017 dataset, it shows an 85.59% performance.

The researchers have utilized a BiLSTM-CRF model [17] to address the NER task, which involves classifying sequences into many classes. CRF stands for Conditional random fields. To do this, they have created several features like BERT embedding, Part of Speech (PoS), and surrounding sequences. As a result, they have created a collection of phrases that directly follow named entities from the training data set. The researchers utilized the i2b2 2010 [18] dataset, which contains annotations for three named entity types: therapy, examination, and issue.

In [5], a clinical and biomedical NER algorithm has been developed using an agile approach. Based on an enhanced BiLSTM-CNN-Char deep-learning design, the approach is optimized for production and is built on the foundation of Apache Spark. The implementation of NER achieved superior accuracy compared to previous methods on 7 out of 8 widely recognized benchmarks for biomedical NER, as well as 3 n2c2 challenges. NER models exhibit significantly higher accuracy of 89% on 2018 n2c2 dataset.

To ensure successful implementation of comprehensive pharmacovigilance on a large scale, it is crucial to have precise NER for prescription entities, including medicines, dosages, causes, and ADEs, extracted from clinical literature. Accurate scope identification is problematic due to the lack of unfavorable event annotations and the presence of semantic ambiguity. The current study [19] explores the integration of contextualized language models and simultaneous learning with multiple clinical NER datasets to tackle this challenge. An innovative method has been presented to enhance the embeddings generated by the BERT language model.

This research [20] provides a comparative analysis of present methods for automatically extracting unpleasant occurrences from texts written in natural language. The solutions being considered utilize neural network language models that have

undergone pre-training utilizing many forms of unlabeled data. The n2c2-2018 and CADEC corpora were used for conducting experiments, employing measures that were generated during the CoNLL competition. The models of the indicated type exhibit effective resolution of this task.

This article [21] aimed to compare the automatic entity recognition performance of three tools: Amazon Comprehend Medical (ACM), Clinical Language Annotation, Modeling and Processing (CLAMP) toolkit, and Spark NLP. A normal, checked dataset and its marked elements were used for the evaluation.

This study [22] introduces MedBERT, a new pre-trained transformer-based model specifically developed for biomedical NER. MedBERT undergoes training using 57.46 million units from N2C2, BioNLP, CRAFT contests, and Wikipedia scientific content. MedBERT has been used for four publicly available pre-trained models.

The authors in [23] conducted a performance evaluation of six NER algorithms using five biomedical text datasets. They discovered that moving systems to datasets with little ambiguity, such as the I2B2 2014 dataset, is more efficient compared to transferring them to datasets with noisy text, such as the SMM4H 2021 dataset.

In [24], The researchers performed an experiment employing a pre-trained BERT-based algorithm to detect drug-associated details from irregular discharge summaries. AB-XLNet shows a 3% improvement in F1-score in comparison to XLNet.

This work [25] presents a hybrid idea extraction method that integrates rule-based and deep-learning approaches to achieve improved accuracies. The suggested method addresses the limitations of the conventional rule-based methodology by integrating it with DL, which acquires rules implicitly from historical data through augmentation and bootstrapping. The proposed methodology is assessed using an extensive dataset of prescription information obtained from a pharmacy chain.

3 Implementation

3.1 Dataset

The Department of Biomedical Informatics at Harvard University [26] manages the National NLP Clinical Challenges (n2c2) database, where the diverse n2c2 datasets are stored [26]. A number of yearly NLP competitions that were initially coordinated as an aspect of the i2b2 project (Informatics for Integrating Biology and the Bedside), a former National Center for Biomedical Computing (NCBC), produced these datasets. In our study, the clinical notes on medications and adverse drug events (ADEs) from the n2c2 2018 [27] were utilized for both the training and evaluation of the medication task. The selection of nine entities from the medication history of patients was the focus of the n2c2 2018 task. This dataset includes 505 discharge summaries along with the entity files (.ann files). The ann file is the distribution of entities in a discharge summary. A sample of discharge summary and its ann file is shown in Fig. 1. Eight parameters (ADE, Strength, Dosage, Duration, Frequency,

```
Admission Date:  [**2115-2-22**]          Discharge Date:  [**2115-3-19**]

Date of Birth:  [**2078-8-9**]          Sex:  M

Service: MEDICINE

Allergies:
Vicodin

Attending:[**First Name3 (LF) 4891**]
Chief Complaint:
Post-cardiac arrest, asthma exacerbation

Major Surgical or Invasive Procedure:
Intubation
Removal of chest tubes placed at an outside hospital
R CVL placement


History of Present Illness:
Mr. [**Known lastname 3234**] is a 36 year old gentleman with a PMH signifciant
with dilated cardiomyopathy s/p AICD, asthma, and HTN admitted
to an OSH with dyspnea now admitted to the MICU after PEA arrest
x2. The patient initially presented to LGH ED with hypoxemic
respiratory distress. While at the OSH, he received CTX,
azithromycin, SC epinephrine, and solumedrol. While at the OSH,
he became confused and subsequently had an episode of PEA arrest
```

(a) Discharge Summary

```
T1    Reason 10179 10197    recurrent seizures
T3    Drug 10227 10233      ativan
T5    Route 10240 10242     IM
T6    Drug 10455 10465      Topiramate
T7    Strength 10466 10470  25mg
T8    Route 10471 10473     PO
T9    Frequency 10474 10477 BID
T10   Duration 10495 10497  PM
T11   Strength 10515 10519  50mg
T12   Route 10520 10522     po
T13   Frequency 10523 10526 BID
T14   Duration 10527 10541  for seven days
T16   Frequency 10581 10588 ongoing
T17   Drug 11014 11024      vancomycin
T18   Drug 11026 11034      cefepime
T19   Drug 11036 11042      flagyl
T20   Drug 11044 11049      cipro
T21   Drug 11055 11066      oseltamavir
T23   Drug 11091 11103      soludemedrol
T24   Drug 11108 11117      albuterol
T26   Form 11118 11121      MDI
T27   Reason 11442 11451    agitation
T28   Drug 11462 11470      sedation
T29   Reason 11577 11586    agitation
T30   Drug 11614 11622      seroquel
T32   Reason 11694 11700    Asthma
T33   Duration 11718 11726  standing
```

(b) ANN File

Fig. 1 Discharge summary and its associated ann file

Table 1 Description of dataset

Train	303
Test	202
Entities	9

Form, Route, and Reason) plus a mention of the drug source are included in the dataset. The breakdown of the dataset's entities and training and testing examples is displayed in Table 1.

3.2 Preprocessing

- **Data Annotation:** All the discharge summaries were annotated with BIO (Begin, inside, outside) annotation scheme with respect to their ann files with the help of various python functions. Each discharge summary has been annotated by

Table 2 Distribution of entities

Type of entity	Frequency of entity
O (Outside)	822,529
B-Drug, I-Drug	25,364, 3112
B-Form, I-Form	11,652, 3982
B-Strength, I-Strength	10,452,8121
B-Reason, I-Reason	9691,4505
B-Route, I-Route	9407,563
B-Frequency, I-Frequency	9143, 8085
B-Dosage, I-Dosage	5092,5017
B-ADE, I-ADE	2001,1024
B-Duration, I-Duration	1390,963

developing functions in python language. A separate function is developed for all the categories of entities like identifying:

1. *sequences of exact entities like "vancomycin"," cefepime", etc.,*
2. *sequences of all joint entities like "for seven days", etc., and*
3. *overlapping entities like "seven" can be represented as Duration entity and quantity of medicine also*

All the single entities (Exact entities) have been annotated as "B-Entity" like "B-Drug" and joint entities as "B-Entity I-Entity" like "for seven days" will be annotated as "B-Duration I-Duration I-Duration". The dispersion of entities across the text in discharge summaries has been displayed in Table 2.

- **Preprocessing:** Uppercase characters in discharge summaries were converted to lowercase letters to optimize the training of the discharge summary in the model. Subsequently, the text was detected via BIO tagging. Furthermore, the BERT models employed in this work had a restricted input size of 512 words. In order to achieve this objective, the BIO-tagged tokens were preserved in their original form but represented through WordPiece tokenization. Furthermore, the CLS and SEP markers denoted the beginning and conclusion for every statement.

3.3 Results and Evaluation

The BIO segment format is utilized, in which every word is designated to appear at the start (B) and end (I) of a source mention. Additionally, we extend the BIO segment to incorporate it into the WordPiece tokenization results, where the label supplied to the original word is taken by the first token, and the tag "X" is added for subsequent tokens. We assessed our models' performance using the 2018 n2c2 challenge. The primary evaluation criteria for all types of entities in the challenge are Precision (P), Recall (R), and F1-score (F1).

We utilized the PyTorch framework to implement and create the entity extraction system. The trials were conducted using a Google Colab Pro server equipped with a Tesla T4 graphics processing unit (GPU). We implemented the Adamw optimizer with a learning rate of $5e-5$, a batch size of 15, this task achieved peak performance over a span of eight epochs.

An investigation was undertaken to evaluate the impact of the following three commonly utilized pre-trained language models on the entity extraction system's overall efficacy:

- **Biobert**: This pre-trained model, which makes use of the BERT construction, incorporated summaries from PubMed and full-text studies from PMC. The latest version of this model has been used which was trained using the PubMed dataset.
- **Scibert**: This is initialized using the BERT model and trained on a dataset of 1.14 million entire text documents from Semantic Scholar. The scivocab model has been employed which is constructed using SentencePiece on a scientific corpus.
- **Roberta**: This shares the same structure as BERT, with the exception of removing the subsequent statement forecast task and modifying the constant concealing pattern to a dynamic one.

Figure 2 shows the reported results for entity extraction task on n2c2 2018 dataset, respectively. This figure also shows that when compared with different models, the SciBERT approach performs better overall for the entity retrieval tasks. Figure 3 shows the loss occurred during training and testing. Additionally, as Fig. 4 illustrates, we contrasted the effectiveness of our study with earlier studies on the n2c2 2018 dataset.

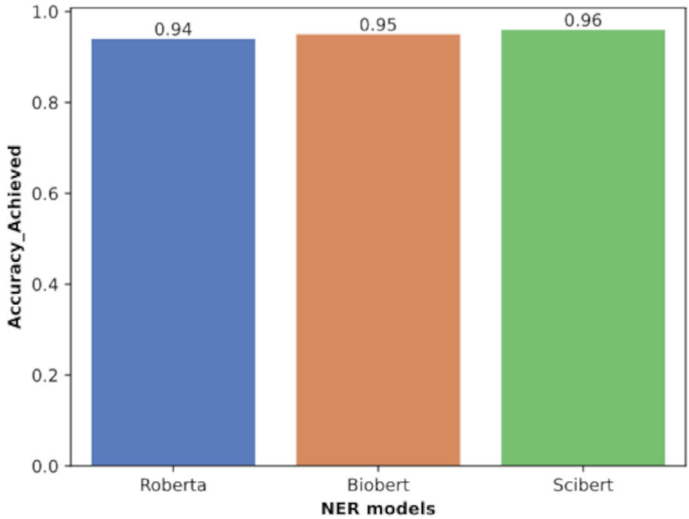


Fig. 2 Performance of applied models

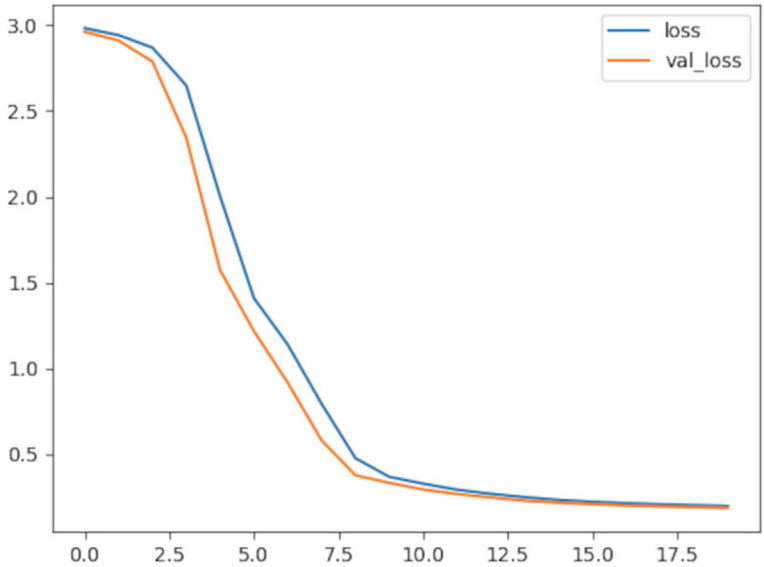


Fig. 3 Training loss and testing loss

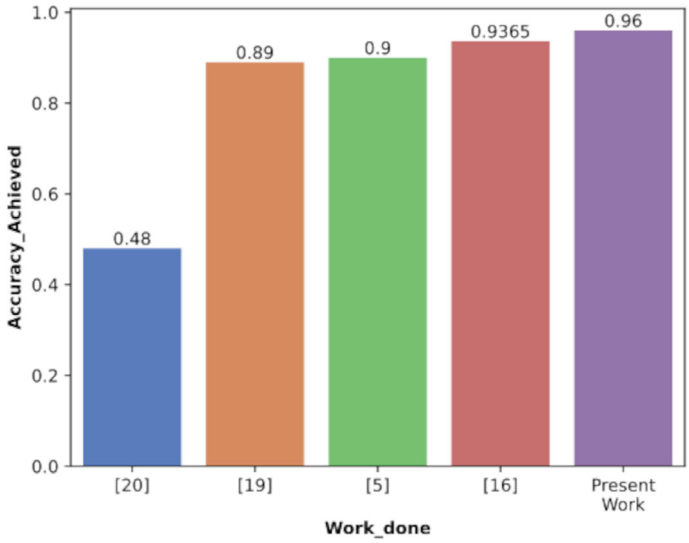


Fig. 4 Comparison of previous works

4 Conclusion and Future Work

This study aimed to extract information about medical entities, dataset was annotated using python functions and a pre-trained model that incorporates contextual information for the extraction process. Consequently, the efficiency of automatic entity extraction was enhanced using three NER models. Our trial, which involved nine entities, demonstrated that SciBERT exhibited the utmost level of efficiency in relation to the accuracy measure. In future, entity extraction can be enhanced by embedding medical ontologies to convert the entity recognition into multi-label entities.

References

1. Juhn Y, Liu H (2020) Artificial intelligence approaches using natural language processing to advance EHR-based clinical research. J Allergy and Clinical Immunol 145(2):463–469
2. Demner-Fushman D, Elhadad N, Friedman C (2021) Natural language processing for health-related texts. In: Biomedical informatics: computer applications in health care and biomedicine. Cham, Springer International Publishing, pp 241–272
3. Houssein EH, Mohamed RE, Ali AA (2021) Machine learning techniques for biomedical natural language processing: a comprehensive review. IEEE Access 9:140628–140653
4. Qin X, Liu J, Wang Y, Liu Y, Deng K, Ma Y, ... Sun X (2021) Natural language processing was effective in assisting rapid title and abstract screening when updating systematic reviews. J Clinical Epidemiol 133:121–129
5. Kocaman V, Talby D (2022) Accurate clinical and biomedical named entity recognition at scale. Software Impacts 13:100373
6. Bhardwaj P, Baliyan N (2024) A novel and effective multi-class classification method for imbalanced medical transcriptions. IETE J Res 1–11
7. Bitterman DS, Miller TA, Mak RH, Savova GK (2021) Clinical natural language processing for radiation oncology: a review and practical primer. Int J Radiat Oncol* Biol* Phys 110(3):641–655
8. Grossman Liu L, Grossman RH, Mitchell EG, Weng C, Natarajan K, Hripcsak G, Vawdrey DK (2021) A deep database of medical abbreviations and acronyms for natural language processing. Scient Data 8(1):149
9. Lauriola I, Lavelli A, Aiolli F (2022) An introduction to deep learning in natural language processing: models, techniques, and tools. Neurocomputing 470:443–456
10. Johnson SA, Signor EA, Lappe KL, Shi J, Jenkins SL, Wikstrom SW, ... Witt DM (2021) A comparison of natural language processing to ICD-10 codes for identification and characterization of pulmonary embolism. Thrombosis Res 203:190–195
11. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, ... Poon H (2021) Domain-specific language model pretraining for biomedical natural language processing. ACM Trans Comput Healthcare (HEALTH) 3(1):1–23
12. Brown IIIW, Balyan R, Karter AJ, Crossley S, Semere W, Duran ND, ... Schillinger D (2021) Challenges and solutions to employing natural language processing and machine learning to measure patients' health literacy and physician writing complexity: the ECLIPPSE study. J Biomed Inform 113:103658
13. Alla NLV, Aipeng CHEN, Batongbacal S, Nekkantti C, Dai HJ, Jonnagaddala J (2021) Cohort selection for construction of a clinical natural language processing corpus. Comput Methods and Programs in Biomed Update 1:100024

14. Dong Q, Wang Y (2021) Enhancing medical word sense inventories using word sense induction: a preliminary study. In: Heterogeneous data management, polystores, and analytics for healthcare: VLDB workshops, Poly 2020 and DMAH 2020, Virtual Event, August 31 and September 4, 2020, Revised Selected Papers 6, Springer International Publishing, pp 151–167
15. Jehangir B, Radhakrishnan S, Agarwal R (2023) A survey on named entity recognition—datasets, tools, and methodologies. Natural Language Process J 3:100017
16. El-allaly ED, Sarrouti M, En-Nahnahi N, El Alaoui SO (2021) MTTLADE: a multi-task transfer learning-based method for adverse drug events extraction. Inf Process Manage 58(3):102473
17. Landolsi MY, Romdhane LB, Hlaoua L (2022) Medical named entity recognition using surrounding sequences matching. Proc Comput Sci 207:674–683
18. Uzuner Ö, South BR, Shen S, DuVall SL (2011) 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc 18(5):552–556
19. Narayanan S, Mannam K, Achan P, Ramesh MV, Rangan PV, Rajan SP (2022) A contextual multi-task neural approach to medication and adverse events identification from clinical text. J Biomed Inform 125:103960
20. Sboev A, Selivanov A, Rylkov G, Rybka R (2021) On the accuracy of different neural language model approaches to ADE extraction in natural language corpora. Proc Comput Sci 190:706–711
21. Shah-Mohammadi F, Finkelstein J (2022) a comparison of automated named entity recognition tools applied to clinical text. In: 2022 7th international conference on intelligent informatics and biomedical science (ICIIBMS), November, vol 7. IEEE, pp 227–231
22. Vasantharajan C, Tun KZ, Thi-Nga H, Jain S, Rong T, Siong CE (2022) Medbert: a pre-trained language model for biomedical named entity recognition. In: 2022 Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC), November, IEEE, pp 1482–1488
23. Ge Y, Guo Y, Yang YC, Al–Garadi MA, Sarker A (2022) A comparison of few-shot and traditional named entity recognition models for medical text. In: 2022 IEEE 10th international conference on healthcare informatics (ICHI), June, IEEE, pp 84–89
24. Oh K, Kang M, Oh S, Kim DH, Kang SH, Lee Y (2022) AB-XLNet: named entity recognition tool for health information technology standardization. In: 2022 13th international conference on information and communication technology convergence (ICTC), October, IEEE, pp 742–744
25. Mishra RK, Roy S, Palla SK, Patel N, Patel M, Jos S (2023) Hybrid approach combining deep learning and a rule based expert system for concept extraction from prescriptions. In: 2023 45th annual international conference of the IEEE engineering in medicine and biology society (EMBC), July, IEEE, pp 1–4
26. Healthcare P (2020) n2c2 nlp research data sets. https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/
27. https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

Intonation Analysis and Classification of Declarative Questions and Statements in Nepali Language using Machine Learning Approaches

Ranjit Subba, Roshan Kumar Prasad, Pratika Rai, and Sharad Sinha

Abstract A speaker's perspective cannot be fully expressed by phoneme segment in a speech, but it also comes along with the intonation. In the Nepali language, declarative questions are uttered with rising intonation and are commonly used in daily conversation. In this paper, training on various machine learning models has been performed to classify declarative sentences and questions. The dataset consists of 700 utterances with ~ 300 declarative statements (falling intonation) and ~ 400 declarative question statements (rising intonation) which are either recorded in the studio or taken from the "Large Nepali ASR training dataset" from openslr.org. The dataset was pre-processed and annotated using PRAAT. Cepstral features MFCCs of every utterance in the data set were calculated and put to use in training five different supervised machine learning models. The data set is split into two parts: 70% for training and 30% for testing purposes, respectively. The comparative study shows that the Random Forest model performed better with the highest evaluation score of ~ 0.87% and the Naïve Bayes model achieved a minimal accuracy of approximately 0.69%. These findings have the potential to enhance the authenticity of Text-to-Speech, Automatic–Speech-Recognition, and Speech-to-Text systems in languages with limited linguistic resources, like Nepali.

R. Subba (✉)
Department of Computer Science, Hijli College Kharagpur,
Kharagpur, West Bengal 721306, India
e-mail: subbanit@gmail.com

R.K. Prasad · P. Rai · S. Sinha
Department of Computer Science and Technology, University of North Bengal,
Raja Rammohunpur, Darjeeling, West Bengal 734013, India
e-mail: rosahanprasad9932@gmail.com

P. Rai
e-mail: rs_pratika@nbu.ac.in

S. Sinha
e-mail: ssinha@nbu.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_38

Keywords MFCC · SVM · Nepali · TTS · STT · ASR

1 Introduction

Tremendous advancements in the field of neural Text-to-Speech (TTS), Speech-to-Text (STT), and Automatic-Speech-Recognition (ASR) systems are contributing to produce speech units with enhanced naturalness and realism. It integrates both long-term suprasegmental prosodic features of speech such as pitch accent, intonation, and rhythm (i.e., these parameters are associated with the tangible aspects of duration, pitch, and intensity) along with short-term cepstral features, i.e., Mel Frequency Cepstral Coefficient (MFCC) [1] to emulate human-like intelligible voices. Prosody and cepstral features also provide cues to the listener and help them interpret the speech correctly. However, the sentences spoken naturally may be impacted by extraneous factors such as regional effect, the speaker's motivation, mood, speed, etc.

Pitch variation, known as intonation, serves as a prosodic element that enhances the naturalness of speech. It involves altering the tone at which words are spoken, effectively communicating the speaker's emotions and attitude to the listener. Furthermore, it draws the listener's attention to crucial elements within the spoken message, while also indicating the intended form of the speech, be it a statement, a question, a command, or an exclamation.

The Nepali language exhibits intonation patterns, allowing a speaker to transform a statement into a declarative question with rising intonation. However, the listener can understand normal interrogative sentences through the auditory characteristics of spoken phonemes. An analysis has been done on the disparity of digital signal content on the prosodic stress emphasized at the end of an utterance, boundary tone, and its associated part of speech (i.e., Linguistic content). Linguistic analysis and manual prosodic annotation of utterances are done using PRAAT. For example, consider the two utterances of the same sentence:

कम्पुटर भाषा | (A statement). IPA notation / kɔmpjuʈɔr bʰasa./

कम्पुटर भाषा**?**(A declarative question). IPA notation / kɔmpjuʈɔr bʰasa/?

A statement utterance can be converted into declarative question through changes in prosody in intonational language such as Nepali. The difference in prosody between a declarative statement and a declarative question is evident in the intensity and fundamental frequency (f0) at the last word or syllable of the statement. To illustrate, accentuating the word "खायौ" in the statement "तमिले भात खायौ" [t̪imi:le: bʰɑ:t̪ɔ̃ kʰɑ:ja:u] /You ate rice. / turns it into the declarative question "तमिले भात खायौ**?**" [t̪imi:le: bʰɑ:t̪ɔ̃ kʰɑ:ja:u''] /You ate rice?/when stressed on the syllable "**-**यौ". Consequently, the use of rising intonation in the final word or syllable serves as a distinguishing factor between a statement and a declarative question.

In this paper, an analysis has been done on various supervised machine learning-based statement/question classifiers to maximize the naturalness and intelligibility of Nepali speech synthesis systems.

Section 2 discusses previous studies done in intonation analysis of Indian languages (i.e., Hindi, Bengali, Tamil, Assamese, Nepali) as well as other relevant languages. Section 3 talks about data preprocessing and manual prosodic annotation of speech resources used in building datasets. Section 4 focuses detailed discussion of the methodology used. Section 5 highlights results, discussions, and conclusions.

2 Related Work

Nepali, an Indo-Aryan language indigenous to the Himalayan province of South Asia, had a recorded 2,926,168 Nepali-speaking population in India according to the census of 2011. Nepali, spoken by over 40 million people globally, is considered a language with limited linguistic resources and hence limited prior research has been conducted on intonation patterns in Nepali. Allwood et al. [2], talk about the intonation pattern in the Nepali feedback unit where they have categorized it into two distinct units: the 'Feedback-Giving Unit (GFB),' which participants utilize to indicate their willingness and ability to sustain the communication, along with their level of engagement, comprehension, agreement, or disagreement with the conveyed message. Additionally, there is the 'Feedback-Eliciting Unit (EFB),' which is employed to assess the speaker's response in terms of attention, comprehension, and agreement with the information being communicated. The research reveals that rising intonation is employed in all feedback-eliciting units where feedback has giving non-accept function. All the other feedback-giving units in Nepali exhibited falling intonation patterns. Banerjee et al. [3], analyzed Hindi intonation to develop a semi-automatically labeled prosody database for Hindi which are valuable source of training data for ASR and TTS systems to incorporate natural-sounding voices. Rao et al. [4], delved into the application of neural network models for capturing intonation patterns in Indian languages, specifically Hindi, Telugu, and Tamil. Their study involved the analysis of recorded news data, utilizing both neural network and SVM models to predict the fundamental frequency (f0) of syllables in these languages to examine the intonation patterns.

3 Speech Dataset

This research makes use of a dataset that consists of 700 utterances from Nepali text recorded in a controlled environment. These audio recordings were sampled at a rate of 16 kHz, and they are represented as 16-bit numbers with mono tracks, each lasting approximately 10 s on average. The dataset consists of approximately 300 statements and around 400 declarative question utterances. The text is sourced from a diverse selection of textbooks, story books, and the Nepali newspaper Himalaya Darpan, encompassing a broad spectrum of domains. The speech

utterances have been recorded in the studio (to avoid external noises) with the help of professional male and female artists aged 21 to 24 years. An open-source dataset "Large Nepali ASR training dataset" from openslr.org [5] has been used. The corpus contained around ~ 157 K utterances within varying age groups of male and female skilled speakers. The speeches obtained from openslr.org exhibited variations in both duration (ranging from 3 to 10 s) and background noise levels when compared to those recorded in a controlled environment therefore data from openslr.org needed additional cleaning and filtering processes. Around ~ 300 relevant speech utterances were utilized for the research objectives. Each utterance of a statement concludes with either a comma or a full stop, while each question concludes with a transcription that includes a question mark. Rising intonation is used for all declarative questions, while non-rising intonation is employed for both statements and regular questions.

3.1 Prosodic Labeling

The sentences from the recorded speech are annotated manually using PRAAT [6], an openly accessible speech analysis software utilized to detect and label the prosodic boundaries and pitch-related tones through pitch movements with the help of a linguistic expert. Analysis was performed on word stress, Accentual Phrase (AP), and boundary tone at the ending portion of each speech utterance along with its associated part of speech. A text grid file is generated for each of the labeled utterances to extract cepstral features (i.e., Mel Frequency Cepstral Coefficient) MFCCs. For example, consider the declarative question कलवारको जनसङ्ख्या**?** [kolvarko ʤʌnsʌŋkʰja?] shown in Fig. 1.

In the given instance, there is a falling fundamental frequency (f0) that reaches its lowest point at the syllable "–सङ् -"[sʌŋ] in the word "जनसङ्ख्या" [ʤʌnsʌŋkʰja].

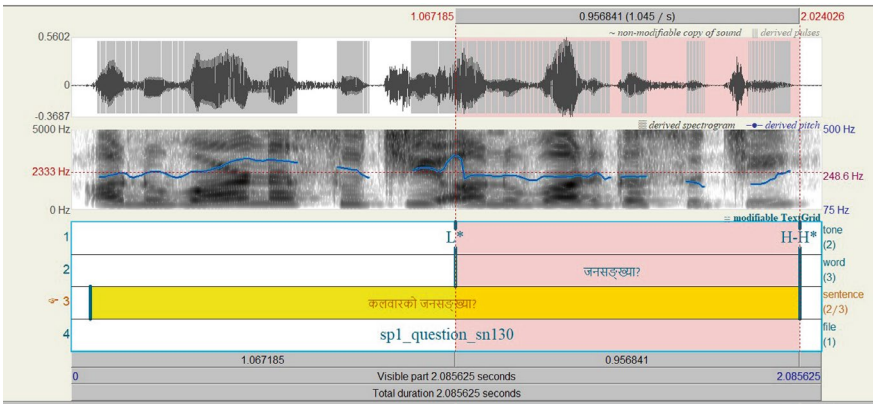


Fig. 1 Declarative question

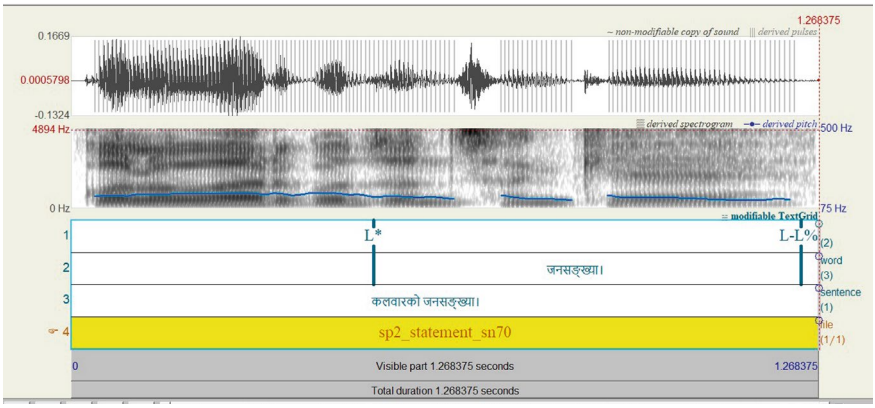


Fig. 2 A statement

Additionally, it is observed that there is a sharp rise in f0 at the end of each utterance. High phrase accent is a characteristic feature of this intonational phrase, indicated as H-, and is succeeded by a high boundary tone, represented as H%. Hence, labelled as H–H% Now consider the following utterances shown in Fig. 2. कलवारको जनसङ्ख्या। |[kolvarko ʤʌnsʌŋkʰja] which is a statement.

However, for the normal statement utterances, there is a falling fundamental frequency (f0) that reaches its lowest point at the particular syllable in the particular word, a low intonational phrase (L-) followed by a low boundary tone (L%).

Preprocessing tasks such as eliminating silence from the initial and final segments of an utterance, as well as reducing noise using PRAAT are conducted. Subsequently, for each labeled utterance, a text grid file is generated to extract cepstral features, specifically MFCCs (Mel Frequency Cepstral Coefficients), pertaining to the final word or syllable of an utterance.

4 Methodology

The Nepali language comprises 67 characters that can symbolize sounds through text. Figure 3 represents different characters in the Nepali language.

L = {'क', 'ख', 'ग', 'घ', 'ङ', 'च', 'छ', 'ज', 'झ', 'ञ', 'ट', 'ठ', 'ड', 'ढ', 'ण', 'त', 'थ', 'द', 'ध', 'न','प', 'फ', 'ब', 'भ', 'म', 'य','र', 'ल', 'व','श', 'ष', 'स', 'ह','अ', 'आ', 'इ', 'ई', 'उ','ऊ', 'ए', 'ऐ', 'ओ', 'औ', 'ा', 'ि', 'ी', 'ु', 'ू', 'ृ', 'े', 'ै', 'ो', 'ौ', 'ं', 'ँ', '०', '१', '२', '३', '४', '५', '६', '७', '८', '९','।', ' '}

Fig. 3 Nepali character

4.1 Feature Extraction

The Mel Frequency Cepstral Coefficient (MFCC) has been calculated for each utterance using their corresponding text grid file. Human auditory perception operates on a non-linear scale, and the MFCCs attempt to mathematically emulate the human ear's behavior. This involves a transformation of actual acoustic frequencies into Mel frequencies, typically falling within the range of 300–5 kHz. Below 1 kHz, the Mel scale maintains a linear relationship, while above 1 kHz, it becomes logarithmic. The MFCC Constants represent the energy attributed to each Mel frequency bin, and these values are specific to individual speakers [7]. In Fig. 4 we have shown the steps followed to calculate MFCC. It is calculated using the following formula:
where f is the frequency.

$$M = \frac{1000}{\log 2} \log \left(1 + \frac{f}{1000}\right) \tag{1}$$

It is found that the machine learning algorithms demonstrate improved classification accuracy when taking Mel coefficients (MFCC1, MFCC2, ……, MFCC39) as input features. Figure 5 represents an instance of input features and their target classes.

4.2 Proposed Methodology

- The audio signal is converted into a waveform.
- The waveform is sliced into different windows.

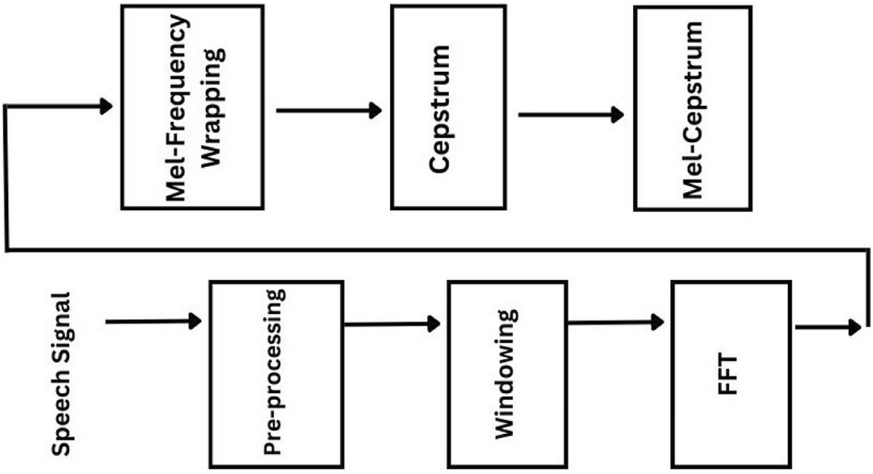


Fig. 4 MFCC steps for feature extraction

mffcc7	mffcc8	mffcc9	mffcc10	...	mffcc32	mffcc33	mffcc34	mffcc35	mffcc36	mffcc37	mffcc38	mffcc39	mffcc40	sentence
-7.264712	0.578228	-21.230816	4.928234	...	-8.131697	-2.848631	1.757815	3.793075	-2.786141	-4.871511	-2.435369	1.241434	-3.018312	1
1.378722	-3.673953	-18.563114	7.048325	...	6.234663	0.230803	-3.317028	-2.725587	-1.392985	-0.312711	-2.023439	-5.346755	-4.563718	1
-4.012286	-0.264864	-17.255156	-9.201622	...	5.985023	7.337078	7.938330	1.782390	1.088473	2.361807	-1.401609	-1.473703	-0.997713	1
-9.219626	7.208112	-27.225110	-10.611770	...	11.890949	17.006065	8.520269	4.357818	-0.652675	-0.712941	0.015991	0.400336	1.667514	1
-5.985364	-1.985432	-20.380562	-7.878456	...	-2.871200	-0.885420	-2.211178	-4.155213	-0.179658	-1.619475	-3.186996	-2.266893	-6.230665	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
-2.459495	-1.550547	-14.135111	-3.795042	...	-2.482360	-3.432692	-4.588032	-2.319709	-4.967949	-4.076546	-3.840144	-4.061710	-6.564909	0
-8.708251	-9.914806	-13.455994	6.428455	...	-4.075973	-6.834992	-6.239100	-4.134049	-4.854927	-2.329749	-2.877318	-4.189696	-7.042574	0
-6.478416	-3.088249	-11.644244	-1.658740	...	-2.025968	-1.345415	-4.606970	-3.227647	-2.891666	-2.440379	-4.494006	-2.326407	-7.102646	0
-9.122546	-14.762268	-13.030032	8.617766	...	-2.956638	-4.839528	-3.872472	-3.628063	-4.077293	-5.762932	-5.498899	-4.365483	-6.603572	0
-4.278409	-8.201500	-15.205486	10.907739	...	-2.702447	-4.674639	-2.543376	-6.329384	-1.961634	-2.675153	-1.728559	-5.008363	-3.472990	0

Fig. 5 Input features and classes

- The length of the window is dependent on the length of the audio signal.
- Following this, the Discrete Fourier Transform (DFT) is performed on each windowed frame and is converted into a magnitude spectrum, according to the equation:

$$x_i(k) = \sum_{N=0}^{N-1} \boldsymbol{x_i(n)h(n)e^{\frac{-j2\pi Ln}{N}}} \tag{2}$$

Figure 6 depicts the methodology used in the study.

4.3 Training

Detailed discussion on the Machine Learning Algorithms (MLAs) used is out of the scope of this study.

SVM Classification Model

The main goal of SVM is to construct a classification function, denoted as f: $R^n \rightarrow \{\pm 1\}$, by leveraging training data containing input–output pairs from two classes, represented as $(x_1, y_1) \ldots (x_i, y_i) \in R^n X \{\pm 1\}$. Here, x_i represents the input pattern, which comprises features such as MFCC1 through MFCC39 and y_i is the desired target class (i.e., Declarative question/statement). If there is linear separability among the points, it implies the existence of a vector w of dimension n and a scalar b such that:

$$wx_i - b \geq 1 \text{ if } y_i = 1, \text{ and } wx_i - b \geq -1, \text{ if } y_i = -1 \tag{3}$$

$$y_i(wx_i - \mathrm{b}) \geq 1 \tag{4}$$

$i = 1 \ldots\ldots l.$

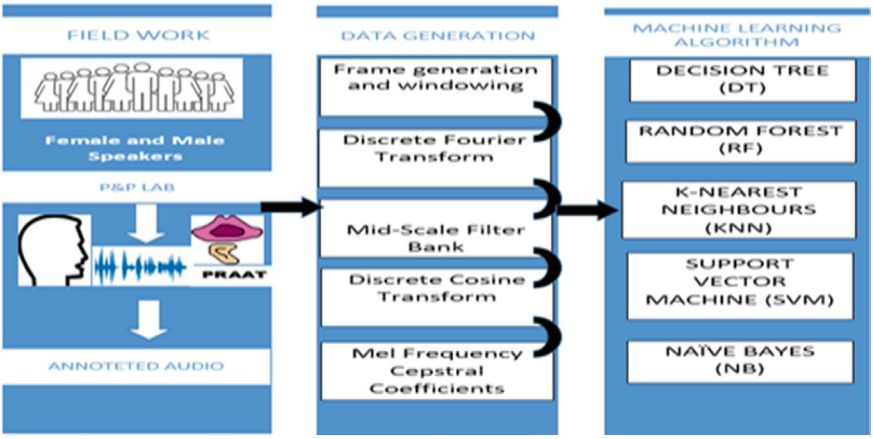


Fig. 6 Workflow in training different machine learning models

"The dashed lines drawn parallel to the separating line mark the distance between the dividing line and the closest vectors to the line. The distance between the dashed lines is called the margin. The vectors (points) that constrain the width of the margin are the support vectors. SVM analysis finds the line (or, in general, hyperplane) that is oriented such that the margin between the support vectors is maximized. Figure 7 illustrates this where the line in the right panel is superior to the line in the left panel" [8, 9].

Decision Tree:

The decision tree classification method consists of two key stages: the creation and the refinement of the tree. The process of tree construction adopts a top-down approach, wherein the tree is progressively divided until all data points belong to the same class label. This phase demands significant computational resources as it entails repeated traversal of the training dataset. In contrast, tree pruning takes a bottom-up approach, aiming to enhance the algorithm's predictive accuracy and classification performance by mitigating the issue of overfitting in the tree that can lead to misclassification errors. "There are many decision tree-based algorithms like ID3, C4.5, C5.0, CART, etc. These algorithms have the merits of high classifying speed, strong learning ability, and simple construction" [10]. Figure 8 illustrates the functioning of the decision tree outlined in this paper.

K-Nearest Neighbor (KNN):

The K-Nearest Neighbor (KNN) algorithm is a straightforward machine learning approach, relying on the idea that similar samples are typically located close to each other [11]. It belongs to the category of instance-based learning methods, also known as lazy learners. Unlike eager learning algorithms (e.g., decision trees, neural networks, and Bayes networks), lazy learners like KNN store all training

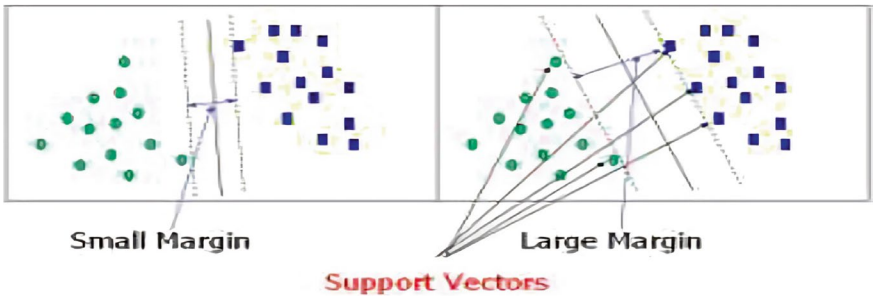


Fig. 7 Two-dimensional classification

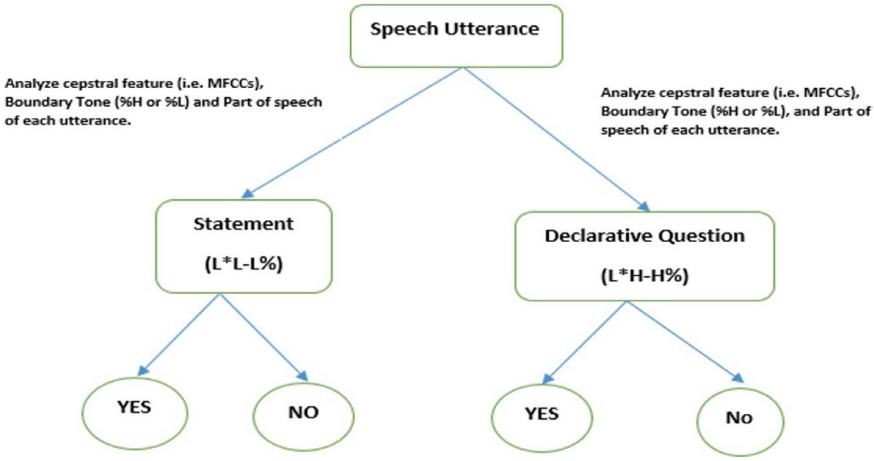


Fig. 8 An example of decision tree induction

samples and delay building a classifier until a new, unlabeled sample needs classification [12]. Although lazy learners require less computation time during training, they involve more computation time in the classification process [13, 14].

K-Nearest-neighbors classifiers rely on learning through similarity, wherein they assess a given test sample by comparing it to the existing training samples that exhibit likeness. To classify a data sample X, the algorithm identifies its K-nearest neighbors and subsequently assigns X to the class label held by the majority of these neighbors. It's worth noting that the selection of the parameter K can significantly influence the performance of the k-nearest neighbor algorithm [14].

Random Forest:
Leo Breiman, a researcher at the University of California, introduced the concept of Random Forest in 2001. This ensemble method consists of multiple individual classifiers, or decision trees, that operate independently. To classify a test sample

using this new classifier, the class label is determined by aggregating the voting results from each classification.

Naive Bayes:

The Naive Bayes Classifier technique relies on the principle of Bayesian Theorem and is employed in situations where the input features exhibit high dimensionality [15]. It is based on "Bayes Theorem". Bayes Theorem posits the following: Consider a data sample, denoted as S, for which the class label is unknown. Additionally, suppose there exists a hypothesis, denoted as H, indicating that this data sample S could potentially belong to a specific class K. It is employed to compute the posterior probability $P(K|S)$, from $P(K)$, $P(S)$, and $P(S|K)$, where the posterior probability of the target class is denoted as $P(K|S)$, $P(K)$ is referred to as the prior probability associated with the class, the likelihood, denoted as $P(S|K)$ represents the probability of a specific predictor belonging to a particular class, $P(S)$ is the prior probability of a predictor of class.

$$P(A|B) = \frac{P(X|C)P(C)}{P(X)} \tag{5}$$

Performance Analysis:

Assessing a model's performance is of utmost importance, and it can be done using different metrics. Among these metrics, accuracy, precision, recall, and the F1 Score are the most essential and frequently employed. Precision measures the number of correctly predicted positive class instances out of all predicted positive instances. Recall, on the other hand, quantifies the ratio of correctly predicted positive instances to all actual positive instances. The F1 Score, which combines Precision and Recall, acts as a weighted average, offering a balanced assessment that takes into account both false positives and false negatives.

Results

Table 1 shows the comparative studies of results based on different matrices on which the machine learning models are evaluated.

The performance of each model is evaluated using Confusion Matrix. The Confusion Matrix provides a visual representation that effectively illustrates the results of a classification model. In this matrix, columns represent predicted classes, while rows signify actual classes. Figure 9 represents the confusion matrices for Random Forest (RF) and Decision Tree (DT).

Table 1 Comparison of the evaluation scores for different MLAs

MLAs	Accuracy	Precision	Recall	F1-score
RF	0.865714	0.865789	0.865714	0.864361
DT	0.718571	0.723022	0.718571	0.719223
KNN	0.721428	0.7814516	0.721428	0.692561
SVM	0.752857	0.753917	0.752857	0.752971
NB	0.688571	0.696501	0.688571	0.689921

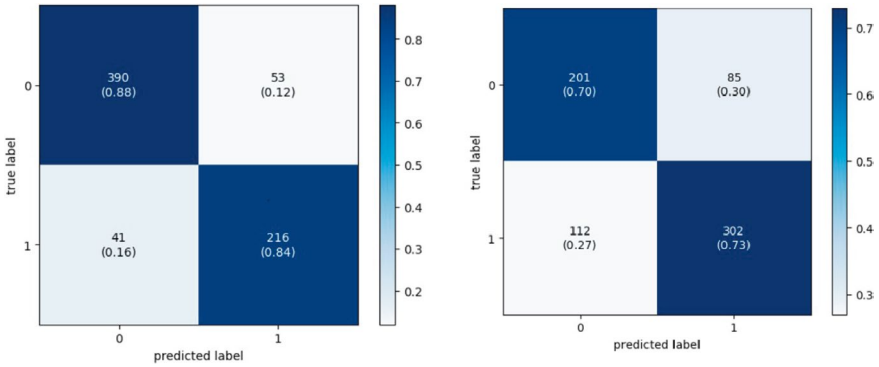


Fig. 9 Confusion matrix for RF and DT

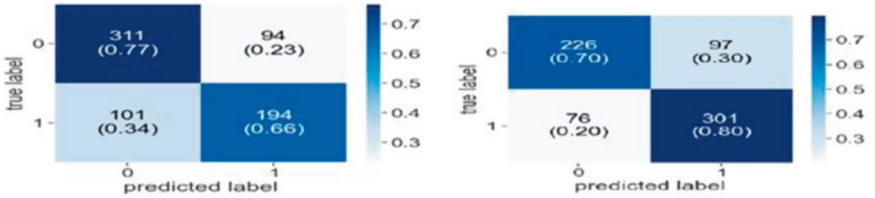


Fig. 10 Confusion matrix for KNN and SVM

Figure 10 represents the confusion matrices for the KNN and Support Vector Machine (SVM).

Figure 11 shows the confusion matrices for Naive Bayes (NB).

MFCCs were used as input features for training different models. It was observed that a significant portion of the data contained non-zero values, leading to improved performance of the Random Forest algorithm. Moreover, the issue of overfitting was substantially reduced, along with mitigating data loss.

5 Conclusion and Future Work

The study unveiled that minimal prior research has been conducted on intonation patterns in the Nepali language. This investigation aims to analyze the intonational patterns of basic declarative and declarative question sentences. Speech synthesis without having intonation may not be natural and meaningful as the same sentence may be a statement, a question, or even an exclamation. With this finding,

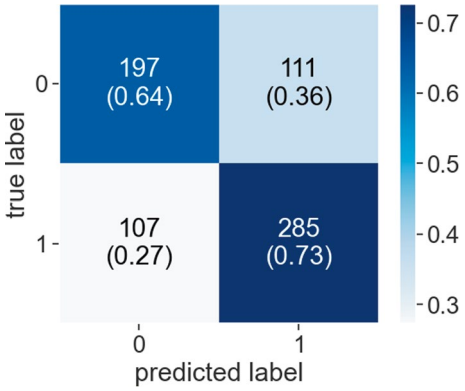


Fig. 11 Confusion matrix for NB

intonation of a declarative statement can be distinguished from that of a declarative question and has the potential to enhance the naturalness of Text-to-Speech (TTS), Automatic-Speech-Recognition (ASR), and Speech-to-Text (STT) systems particularly when dealing with linguistically limited resource language such as Nepali. Training on various supervised machine learning models was conducted, leading to the proposal of machine learning-based question/statement classifiers. Approximately 700 speech utterances are manually annotated and prosodic boundaries for each utterance were done using PRAAT. The MFCCs for all the annotated speech utterances are computed. It is observed that Random Forest performed better with the highest evaluation score of ~ 0.87% and Naïve Bayes with the least ~ 0.69%. The challenge we encountered during the research was the limited availability of annotated speech datasets. Manually extracting pertinent speech audio from a "Large Nepali ASR training dataset", along with preprocessing and annotating using PRAAT, posed significant challenges. Consequently, the decision was made to refrain from training on neural network models due to limited availability of annotated speech dataset. The finding is a preliminary approach for analyzing the intonation pattern, thus it consists of only two classes and is one of the limitations of this research. Additional classes still need to be incorporated, and it is necessary to analyze the performance and efficiency using various features. Furthermore, these outcomes could lay the groundwork for the creation of a state-of-the-art Nepali Text-to-Speech neural network model that excels in naturalness in forthcoming endeavors.

References

1. Laskowski K, Jin Q (2009) Modeling instantaneous intonation for speaker identification using the fundamental frequency variation spectrum. In: IEEE international conference on acoustics, speech and signal processing, IEEE, pp 4541–4544. https://doi.org/10.1109/ICASSP.2009.4960640
2. Jens A, Bhim N (2010) Intonation patterns in Nepali feedback units. In: Proceedings of oriental COCOSDA 2010 held in Kathmandu, Nepal, pp 24–25. Accessed 16 Sep 2023. [Online]. Available https://gup.ub.gu.se/file/112016
3. Banerjee E, Kr Ojha A, Nath Jha G (2021) Prosody labelled dataset for hindi using semi-automated approach, ArXiv, pp 14–19. https://doi.org/10.48550/arXiv.2112.05973
4. Rao KS, Yegnanarayana B (2009) Intonation modeling for Indian languages. Comput Speech Lang 23(2):240–256. https://doi.org/10.1016/j.csl.2008.06.005
5. Kjartansson O, Sarin S, Pipatsrisawat K, Jansche M, Ha L (2018) Crowd-sourced speech corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In: Proceedings the 6th international workshop on spoken language technologies for under-resourced languages (SLTU), 2018, pp 52–55. https://doi.org/10.21437/SLTU.2018-11
6. Jadoul Y, Thompson B, de Boer B (2018) Introducing Parselmouth: a python inter-face to Praat. J Phon 71:1–15. https://doi.org/10.1016/j.wocn.2018.07.001
7. Singh N, Khan RA, Shree R (2012) MFCC and prosodic feature extraction tech-niques: a comparative study. Int J Comput Appl 54(1):9–13. https://doi.org/10.5120/8529-2061
8. Bhavsar H, Panchal MH (2012) A review on support vector machine for data classification
9. Li YS, Peiyi PH (2003) SVM classification : its contents and challenges
10. Brijain M, Patel R, Kushik M, Rana K (2014) A survey on decision tree algorithm for classification. [Online]. Available www.ijedr.org
11. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. https://doi.org/10.1109/TIT.1967.1053964
12. Han J, Kamber M, Pei J (2012) In: Data mining: concepts and techniques. Elsevier.https://doi.org/10.1016/C2009-0-61819-5
13. Kotsiantis SB (2007) Supervised machine learning: a review of classification techniques. Informatica 31:249–268. Accessed 17 Sep 2023 [Online] Available https://api.semanticscholar.org/CorpusID:47128183
14. Bhavsar H, Ganatra A (2023) A comparative study of training algorithms for super-vised machine learning. Int J Soft Comput Eng (IJSCE) 2(4). Accessed 17 Sep 2023 [Online]
15. Nikam SS (2015) A Comparative study of classification techniques in data mining algorithms. Peer Rev Res J Published By 8(1) [Online]
16. Soman KSD, Diwakar S, Ajay V (2005) Insight into data mining theory and practice. Prentice Hall PHI Pvt. Ltd. Accessed 17 Sep 2023 [Online]. Available https://books.google.co.in/books?hl=en&lr=&id=TmvWI-b77AIC&oi=fnd&pg=PR1&dq=Insight+into+-data-+mining:+Theory+and+practice&ots=qOO6csRcCL&sig=Wm3QdgX4nmPHrv-13jaL17IBAtM#v=onepage&q=Insight%20into%20data%20mining%3A%20Theory%20and%20practice&f=false

Optimization of Energy Efficiency in 5G Network Through RAO-I Algorithm

Shubham Pandey and Mitul Kumar Ahirwal

Abstract The rise of 5G technology signifies a pivotal transformation in communication networks, offering unprecedented speed and capacity. However, escalating data demands have exposed the limitations of conventional methods, particularly in managing large data volumes, latency-sensitive applications, and remote connectivity needs. The integration of 5G with Unmanned Aerial Vehicles (UAVs) emerges as a groundbreaking solution, introducing dynamic aerial base stations that can adapt to enhance coverage and efficiency. This paper delves into the intricate task of optimizing energy efficiency within a converged 5G and satellite network, employing the Rao-1 Algorithm to maximize energy efficiency while maintaining optimal device power and association strategies. Our simulations unveil that the zenith of energy efficiency is attained when an equal number of devices connect to both the UAV and the ground station, resulting in a remarkable 65% boost in efficiency compared to the exclusive use of a single communication channel. In summary, the fusion of 5G and UAV technology promises to redefine data transmission, transcending traditional network constraints and exemplifying our ability to leverage technology for societal betterment.

Keywords Optimization · Rao-I · 5G network · UAV · Energy efficiency

1 Introduction

In the realm of technological breakthroughs, the evolution of communication networks has been a pivotal force in shaping the way individuals, businesses, and societies interact and operate. Among the most recent and transformative developments in this domain is the coming of the fifth generation of wireless technology, commonly

S. Pandey · M. K. Ahirwal (✉)
Maulana Azad National Institute of Technology, Bhopal 462003, India
e-mail: mkahirwal@manit.ac.in

S. Pandey
e-mail: s333pandey@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_39

mentioned as 5G. Promising higher speed, connectivity, and capacity, the 5G network is poised to transform the digital landscape, enabling a wide array of applications and services that were previously beyond the scope of conventional wireless systems. 5G technology is a driving force behind innovation, enabling faster, more responsive, and more connected technological solutions across industries. Its impact extends beyond just faster smart phones. It forms the basis for many groundbreaking technologies that will define our future [1].

As our reliance on data continues to grow, conventional data transmission methods face several limitations. Ground-based networks can struggle to cope with the ever-increasing data volume, latency-sensitive applications, and the need for connectivity in remote or disaster-stricken areas. These challenges underscore the necessity of finding more efficient and flexible solutions for data transmission [2].

In an era characterized by booming data needs and the rapid evolution of technology, the combination of 5G (fifth generation) wireless networks and Unmanned Aerial Vehicles (UAVs), also known as drones, holds immense promise in upgrading data transmission. Unmanned Aerial Vehicles (UAVs) present an exceptional prospect for the creation of aerial base stations, which can adaptively relocate to enhance coverage, spectral efficiency, and overall user experience [3]. This innovative synergy addresses critical challenges, justifies its necessity, and offers a glimpse into its potential. The integration of 5G technology with UAVs for data transmission is poised to reshape the way we connect and communicate. By addressing the limitations of traditional networks and offering unparalleled flexibility, mobility, and resilience, this innovative pairing not only meets the demands of our data-driven world but also opens the door to a multitude of applications across industries. As technology continues to advance, the fusion of 5G and UAVs exemplifies our capacity to leverage innovation for the betterment of society [4].

However, there are numerous significant obstacles that continue to exist when it comes to enhancing energy efficiency in scenarios involving the convergence of 5G and satellite networks for power distribution and transmission [5]. Firstly, due to the considerable distance between satellites and ground PDs, direct data uploads from the ground to satellites can result in significant transmission loss. Current research on data transmission in 5G and satellite-converged networks often overlooks the potential advantages of using UAVs. Integrating UAVs into the architecture of a 5G and Converged satellite network for data transmission, allowing UAVs to serve as relays and forward data through designated air-space links, can effectively mitigate transmission losses [5].

Optimizing energy efficiency within the context of a converged 5G and satellite network, where devices have the option to transmit data to both UAVs and terrestrial base stations, presents a critical challenge. The main objective of this research is to address the energy efficiency optimization problem in this integrated network, taking into account various constraints and parameters. Specifically, the study's objective is to determine the optimal device power allocation and association strategy that maximizes energy efficiency while ensuring reliable data transmission. The ultimate goal is to identify the conditions under which the converged 5G and UAV network

achieves its peak energy efficiency, offering a solution that can transform the way data is transmitted in a data-driven world.

In brief, this study explores the challenge of constrained optimization of energy efficiency within a converged network that combines 5G and satellite technologies using the optimization technique that uses the RAO—1 algorithm. The total energy efficiency of the system is maximized while maintaining the optimum power of the devices in an optimized network model. Here in this study, a network model consisting of power devices (PD) with the option and equal chances of transmitting the data to ground Base Station (BS) or Unmanned Aerial Vehicle (UAV) is used. Ultimately all the data has to be transmitted to the cloud server via BS and UAV.

2 Literature Review

Some of the past and ongoing research in the field of 5G is included in this paper. Salah et al. [6] proposed adaptive massive MIMO optimization for 5G networks, dynamically adjusting active antennas based on user demand to maximize energy efficiency while maintaining spectral efficiency. In [7], analysis of concave relationship between the terminal users' number and system energy efficiency in massive MIMO downlink, deriving optimal values through EE function derivatives has been shown. It extends this analysis to uplink and downlink scenarios, exploring various processing schemes and jointly optimizing BS antennas and terminal users for optimal EE, considering both perfect and imperfect CSI. Patcharamaneepa korn et al. [8] explored space modulation (SM) in multiuser massive MIMO systems, enhancing uplink transmission by utilizing multiple antennas per user and a large antenna array at the base station (BS) [9]. They analyzed the antenna combination detection probability using order statistics and applied it to approximate achievable sum rates with common linear processing methods, such as matched filtering (MF) and zero-forcing detection [9]. Abrar et al. [10] in their study, found that UAVs, due to their mobility, affordability, and small size, can serve as edge servers, augmenting Mobile Edge Computing (MEC) capabilities, and offering on-demand resources and services in scenarios where terrestrial MEC networks are unavailable or disrupted by natural disasters, as IoT device applications and mobile traffic continue to grow amid limited resources. The researchers [11] saw that self-sustaining low-power SDs without manual battery replacements were a promising solution, RF-based energy harvesting, complemented by MEC offloading to enhance SD computation capabilities, making MEC-assisted wireless networks with RF-based EH a focus in IoT system design. Ahmed et al. [12] focus on an energy-efficient UAV system, improving UAV trajectory, user scheduling, transmit power, and mobility. It first addresses throughput maximization with an efficient algorithm and then formulates an energy-efficienct problem by enhancing throughput and UAV propulsion energy. Xu et al. [13] explore the energy efficiency of a UAV wireless communication system featuring multiple Ground Terminals (GTs). However, the introduction of multiple GTs poses heightened performance demands and presents challenging issues for the UAV system.

The authors previously addressed energy-efficient communication by optimizing UAV trajectory, specifically developing a mathematical model for fixed-wing UAV propulsion energy consumption in papers [14] and [15]. Notably, their prior work did not address the presence of multiple GTs or Quality of Service (QoS) requirements for these terminals. In the context of the authors [16], management and orchestration entities are responsible for complex decisions, including minimizing energy consumption and costs, determining VNF instances for B.F nodes, and optimizing traffic routing in 5G networks and it introduces a model, problem formulation, and solution concept named OptiLoop, with two implementations, to address these challenges. Prasad et al. [17] concluded that simplifying expressions in massive MIMO systems leads to gains in array and multiplexing, enabling reduced UE transmission power and increased system throughput. These systems also show potential for substantial energy efficiency improvements in 5G networks, despite open research challenges. Mujkic et al. [18] analyzed the energy efficiency of a load-adaptive massive MIMO system, taking into account two user location distribution (ULD) models: boundary-focused (BF) and centre-focused (CF). It analyses the impact of various factors on energy efficiency, highlighting the benefits of adaptively selecting the number of antennas over fixed configurations. The results show that EE is influenced by cell size and the number of users, with adaptive systems achieving superior energy consumption due to antenna optimization.

3 Methodology and System Design

The network setup for the overall distribution involves several components that are Low Earth Orbit (LEO) satellite, an Unmanned Aerial Vehicle (UAV), a terrestrial base station (BS), and a set of ground-based devices. These devices serve different power-related functions producing various data such as Procuring data, Surveillance data, multimedia data, Signal Quality, and Coverage Data. This data needs to be sent to a cloud data processing facility or cloud data center. This model features a terrestrial base station that is equipped with a cloud data center. This allows the devices to transmit their collected data to the station. However, because both the devices and the BS have fixed locations, the transmission of data for devices situated farther away results in significant signal loss and high energy consumption. This leads to lower data transfer rates, hampering data collection from the devices. Moreover, a significant number of devices sending data to terrestrial base station can overload the transmission link. The devices send their data to a Low Earth Orbit (LEO) satellite through an Unmanned Aerial Vehicle (UAV). The UAV follows a predefined path to act as a data relay, forwarding information to the LEO satellite. Similar to a terrestrial base station, the LEO satellite is equipped with a cloud server to store and process the data. The essential part of this system includes a low-orbit satellite orbiting at a fixed height. An Unmanned ariel vehicle moving in a circular trajectory of fixed radius on a fixed height from the ground. A circular area has been considered for random

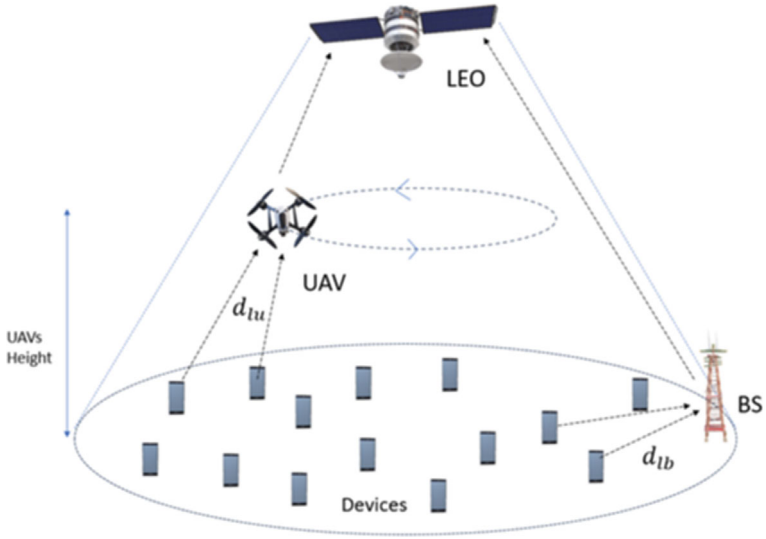


Fig. 1 5G network model consists of LEO, UAV, base station, and devices

distribution of power devices and a terrestrial base station is placed in between the devices. Figure 1, shows the 5G Network Model used for simulation in this study.

The devices can send data either to the Base Station or the Unmanned Aerial Vehicle in the same time slot. It can be given by the Eq. (1),

$$x_{lb} + x_{lu} \leq 1,$$

Here, x_{lb}. symbolizes the data sent to the Base Station, while x_{lu}. symbolizes the data being sent to the UAV. Assuming compensation for the Doppler effect caused by the UAV's motion [19], the channel gain from the devices to the UAV adheres to the free-space path loss model [20], the optimization function is being formulated for this case based on the function given by Meng et al. [5]. Moreover, given the significant distance between the UAV and the LEO satellite, we can disregard horizontal distance. Consequently, the channel gain between the UAV and the LEO satellite still follows the free-space path loss model [5]. The final Eqs. (2–4) as given by Meng et al. [5].

$$Z = \frac{\sum_{l=1}^{L} R_l}{\sum_{l=1}^{L} E_l}, \tag{1}$$

$$R_l = x_{lu} * W \left(1 + \log_2 \frac{c\frac{P_l * h_0}{s * d_{lu}^2} * \frac{P_u * h_0}{s * (H_g - H_u)^2}}{1 + \frac{P_l * h_0}{s * d_{lu}^2} + \frac{P_u * h_0}{s * (H_g - H_u)^2}} \right)$$

$$+ x_{lb} * W * \log_2\left(1 + \frac{l * 10^{\frac{-10*d_{lb}^{-4}}{10}}}{s}\right) \tag{2}$$

$$E_l = P_l * t, \tag{3}$$

In this context, d_{lu} represents the separation distance between the devices and the UAV, while d_{lb} refers to the separation distance between the devices and the Base Station (BS). H_u represents the minimum altitude of the UAV above the ground, and H_g signifies the altitude of the Low Earth Orbit (LEO) satellite. h_0 represents the channel gain per unit distance, and h_u represents the channel gain between the UAV and the LEO satellite. P_u stands for the transmission power of the UAV, while P_l represents the transmission power of the devices. The variability in additive white Gaussian noise (AWGN) for this scenario is represented by the parameter s.

Constraints in constrained optimization pertain to the boundaries or restrictions placed on the variables that determine the outcome of an optimization problem [21]. These limitations establish the realm of possible solutions in which the optimization algorithm seeks the most optimal result. Constraints are pivotal in ensuring that the optimization process adheres to practical constraints and aligns with real-world constraints and requirements. The main aim of the paper is to optimize the function under the given constrains which are as follows:

$$C1: \quad x_{lu} \in \{0, 1\}, \tag{4}$$

$$C2: x_{lb} \in \{0, 1\}, \tag{5}$$

$$C3: x_{lu} + x_{lb} \leq 1, \tag{6}$$

$$C4: \sum_{l=1}^{L} x_{lb} \leq L/2, \tag{7}$$

$$C5: \sum_{l=1}^{L} x_{lu} \leq L/2 \tag{8}$$

$C1$ and $C2$ signify that the devices association strategy is binary in nature, i.e., they can either be 0 or 1. $C3$ ensures that, during each time slot, each device chooses only one data transmission method—either direct forwarding to the terrestrial 5G base station or uploading to the UAV and subsequent forwarding to the LEO satellite. $C4$ and $C5$ impose constraints on the major devices that can be associated with the UAV or the BS in a given time slot. In this context, the half of the total device is taken as the ideal situation.

4 Proposed Energy Efficiency Algorithm Based on Rao-1 Algorithm

Rao-1 Algorithm is an optimization algorithm that is free of metaphor, algorithm-specific and parameter-less algorithm that aims at providing optimized solutions, i.e., best and worst solutions for a given population based on certain criteria. The equation for Rao-1 Algorithm [22] is as follows.

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,\text{best},i} - X_{j,\text{worst},i}\right), \tag{9}$$

The algorithm in use here aims to maximize energy efficiency over a series of iterations denoted by i, each involving m decision variables and n solutions. In this equation, X_j, k, i represents the value of the jth variable for the kth solution in the population during the ith iteration. r is a random number within the [0, 1] range for each variable. The algorithm operates on a population with a specified number of solutions, decision variables, and an objective function to maximize.

The algorithm assesses the most favorable and unfavorable solutions within the group and adjusts them using the Rao-1 Algorithm formula. It checks whether the newly obtained best and worst solutions surpass the previous ones. If they do, the new solutions replace the previous ones; otherwise, the previous solutions persist as the final solutions. The termination criteria are then evaluated, and if satisfied, the best and worst solutions are deemed optimal. If not met, the process repeats until an optimal solution is reached.

Pseudo-code for the optimization:

Initialization:

– *Define problem dimension, upper and lower bounds for each dimension, constants, and parameters*

Define the fitness function:

– *Input: Particle with transmission power and binary values*
– *Calculate the energy efficiency based on Eq.* (2)
– *Calculate penalty factors based on constraints using Eqs.* (5–9)
– *Return the fitness value*

Set algorithm parameters:

– *maximum function evaluations, population_size, maximum_number of iterations, number of runs*

Run the Rao-I algorithm for each run:

For m in range (run)

Initialize positions randomly in the search space
For k in range (Maximum iteration):

Evaluate fitness for each particle and find the best and worst positions
Update particle positions using Rao-I algorithm
Calculate fitness for the updated positions
Apply the competition rule to update positions if necessary.

Store the best fitness value for the run.
Find the index of the best fitness value in finval1 and calculate the number of function evaluations

Calculate statistics for all runs

End

5 Simulation and Result

The area considered for this system model is the radius of 1000 m, where the UAV is revolving in a fixed orbit at a fixed height that is the lowest height considered after taking the physical parameters in the scenario. The constraints that are fixed are shown in the Table 1. While variable parameters are given in Table 2. Table 3 shows the energy efficiency obtained after optimizing the function using Rao-1 Algorithm and the parameters mentioned above in the table.

This shows that maximum energy efficiency is at the slot when equal number of devices are associated with both UAV and BS. After carefully determining the values of Pl and the allocation of devices between the BS and UAV, the optimal solution results in a remarkable energy efficiency score of 200,100.0956. This efficiency peak

Table 1 Constant parameters of 5G network model

Constant parameters	Value	Constant parameters	Value
L	100–400	Hg	200 × 1000 m
T	1 s	W	e^8
HU	90 m	S	$1.99 \times e^{-18}$ W
H0	1.4×10^{-4} m	Pu	39.8 W

Table 2 Variable parameters of 5G network model

Parameters	Value
d_{lb}	1 ~ 1000.0000
d_{lu}	1 ~ 1004.0418
P_l	1 ~ 40
x_{lu}	0,1
x_{lb}	0,1

Table 3 Energy efficiency results after using Rao-I algorithm

UAV/BS	Run_1	Run_ 2	Run_3	Run_4	Run_5
20/180	129,100.07050297	126,100.07579272	126,100.07430173	127,100.07315832	126,100.08284111
60/140	123,100.07713443	122,100.07858956	122,100.07794377	123,100.07690358	122,100.07794377
100/100	200,100.11879055	200,100.12370168	200,100.11627261	200,100.11805966	200,100.11910772
120/80	200,100.09388866	200,100.0944131	200,100.09388861	200,100.0964517	200,100.09918161
180/20	122,100.12089596	126,100.12044404	124,100.12072485	126,100.12191535	125,100.11186513

is achieved by setting the power level of devices at 20.32095, with a total of 98 devices transmitting data to the BS and 102 devices transmitting to the UAV that again points to the obtained result that is the optimized value at the instance of equal association of the devices. If all devices send their data to base station only instead of using this merged model, then the energy efficiency is 120900.07672109, that is comparatively very less when the devices are also sending their data to UAV.

6 Conclusion

The advent of 5G technology promises a visible transformation in communication networks, offering unparalleled speed and capacity. However, as data demands continue to rise, conventional methods face limitations. Integrating 5G with Unmanned Aerial Vehicles (UAVs) holds immense potential, creating dynamic aerial base stations that can enhance coverage and efficiency. This innovation addresses crucial challenges and opens doors to various applications across industries. Nonetheless, optimizing energy efficiency in converged 5G and satellite networks remains a challenge. This study employs the Rao-1 Algorithm to maximize energy efficiency while maintaining optimal device power and association strategies. Simulations reveal that peak energy efficiency occurs when an equal number of devices associate with both the UAV and ground station. The efficiency improves by approximately 65% as compared to the scenario where a single communication channel mode is being utilized exclusively. In summary, the fusion of 5G and UAV technology promises to transform data transmission, overcoming traditional network limitations. This innovative synergy exemplifies our capacity to harness technology for societal improvement.

References

1. Andrews JG, Buzzi S, Choi W, Hanly SV, Lozano A, Soong AC, Zhang JC (2014) What will 5G be? IEEE J Sel Areas Commun 32(6):1065–1082
2. Li B, Fei Z, Zhang Y (2018) UAV communications for 5G and beyond: recent advances and future trends. IEEE Internet Things J 6(2):2241–2263
3. Fotouhi A, Qiang H, Ding M, Hassan M, Giordano LG, Garcia-Rodriguez A, Yuan J (2019) Survey on UAV cellular communications: practical aspects, standardization advancements, regulation, and security challenges. IEEE Commun Surveys Tutorials 21(4):3417–3442
4. Zeng Y, Wu Q, Zhang R (2019) Accessing from the sky: a tutorial on UAV communications for 5G and beyond. Proc IEEE 107(12):2327–2375
5. Meng S, Zhu S, Wang Z, Zhang R, Han J, Liu J, Sun H, Qin P, Zhao X (2022) JDAPCOO: resource scheduling and energy efficiency optimization in 5G and satellite converged networks for power transmission and distribution scenarios. Sensors 22(18):7085
6. Salah I, Mabrook MM, Rahouma KH, Hussein AI (2022) Energy efficiency optimization in adaptive massive MIMO networks for 5G applications using genetic algorithm. Opt Quant Electron 54(2):125

7. Yang J, Zhang L, Zhu C, Guo X, Zhang J (2021) Energy efficiency optimization of massive MIMO systems based on the particle swarm optimization algorithm. Wirel Commun Mob Comput 2021:1–11
8. Patcharamaneepakorn P, Wu S, Wang CX, Alwakeel MM, Ge X, Di Renzo M (2016) Spectral, energy, and economic efficiency of 5G multicell massive MIMO systems with generalized spatial modulation. IEEE Trans Veh Technol 65(12):9715–9731
9. Maleki M, Mohamed-Pour K, Soltanalian M (2019) Large-system mutual information analysis of receive spatial modulation in correlated multi-cell massive MIMO networks. IEEE Trans Commun
10. Abrar M, Ajmal U, Almohaimeed ZM, Gui X, Akram R, Masroor R (2021) Energy efficient UAV-enabled mobile edge computing for IoT devices: a review. IEEE Access 9:127779–127798
11. Liu Y, Xiong K, Ni Q, Fan P, Letaief KB (2019) UAV-assisted wireless powered cooperative mobile edge computing: joint offloading, CPU control, and trajectory optimization. IEEE Internet Things J 7(4):2777–2790
12. Ahmed S, Chowdhury MZ, Jang YM (2020) Energy-efficient UAV-to-user scheduling to maximize throughput in wireless networks. IEEE Access 8:21215–21225
13. Xu Y, Xiao L, Yang D, Cuthbert L, Wang Y (2018) Energy-efficient UAV communication with multiple GTs based on trajectory optimization. Mobile Inform Syst
14. Lyu J, Zeng Y, Zhang R, Lim TJ (2017) Placement optimization of UAV-mounted mobile base stations. IEEECommun Lett 21(3):604–607
15. Chen Y, Zhang S, Xu S, Li GY (2011) Fundamental tradeoffs on green wireless networks. IEEE Commun Magazine 49(6):30–37
16. Malandrino F, Chiasserini CF, Casetti C, Landi G, Capitani M (2019) An optimization-enhanced MANO for energy-efficient 5G networks. IEEE/ACM Trans Network 27(4):1756–1769
17. Prasad KSV, Hossain E, Bhargava VK (2017) Energy efficiency in massive MIMO-based 5G networks: opportunities and challenges. IEEE Wirel Commun 24(3):86–94
18. Mujkic S, Kasapovic S, Abuibaid M (2022) Completed tasks number maximization in UVA-assisted mobile relay communication system. Comput Mater Continua 71(1)
19. Wu Q, Zeng Y, Zhang R (2018) Joint trajectory and communication design for multi-UAV enabled wireless networks. IEEE Trans Wirel Commun 17:2109–2121
20. Liu Y, Xiong K, Ni Q, Fan P, Letaief KB (2020) UAV-assisted wireless powered cooperative mobile edge computing: joint offloading, CPU control, and trajectory optimization. IEEE Internet Things J 7:2777–2790
21. Rao RV, Pawar RB (2020) Constrained design optimization of selected mechanical system components using Rao algorithms. Appl Soft Comput 89:106141
22. Rao R (2020) Rao algorithms: three metaphor-less simple algorithms for solving optimization problems. Int J Indus Eng Comput 11(1):107–130.4

Identification of Sounds from Construction Sites: A Machine Learning Approach

Sayantan Maji, Himadri Mukherjee, Ankita Dhar, Matteo Marciano, and Kaushik Roy

Abstract The rate at which the world is evolving is astonishing with cutting-edge technologies being introduced every day. There have been developments in every field ranging from constructing gigantic architectures to enhancing the ease and comfort with which we can commute daily. Nevertheless, this has given rise to different types of pollution, sound pollution being one of them. Although the impact of sound pollution is not instantaneously noticed, it has a distinct effect on humans and nature which is observed over a period of time. There are multiple sources that cause sound pollution. Sounds from ongoing construction sites are one of them. Such sounds disrupt the daily lifestyle of people, their health, and daily productivity. Mostly, these sounds start very early in the day thereby making it difficult for people to manually take precautions like shutting doors and windows or deploying mobile soundproofing solutions. Machine learning has advanced significantly over the years leading to the refinement of artificial intelligence. This has aided in the development of multifarious solutions to ease and simplify our daily lifestyle. This sets the stage for an autonomous system, that can detect sounds from construction sites very early to automatically deploy precautionary measures so that the effect of these sounds on close by residents can be minimized. In this paper, a system is presented that models audio clips using MFCC-based features for the identification of sounds from construction sites in the Urban environment. Experiments were performed on a dataset of over $6K$ clips composed of the varied construction sounds as well as ambient urban sounds and a maximum accuracy of 98.02% was achieved using random forest-based classification.

S. Maji (✉) · A. Dhar
Department of Computer Science and Engineering, Sister Nivedita University, Kolkata, India
e-mail: sayantanmaji2015@gmail.com

H. Mukherjee · K. Roy
TISA Lab, Department of Computer Science, West Bengal State University, Kolkata, India

M. Marciano
Gazelien Records Lab, Department of Arts and Humanities, Music Program, New York University Abu Dhabi, Abu Dhabi, UAE
e-mail: matteo.marciano@nyu.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_40

Keywords Sound pollution · Urbanized landscape · Construction sounds · Machine learning

1 Introduction

Over the years, there have been remarkable advancements across all dimensions which has reduced hassle in the daily life of humans. Such advances span from medical treatment to communication as well as running regular errands. This evolution of technology has also contributed to the introduction of new modalities of pollution which is bringing about significant harm to our planet as well as to the health of all living creatures at an alarming pace. Among the varied types of pollution, sound/noise pollution is of high concern. While it may not exhibit any immediate consequence but acts as a camouflaged peril. It disrupts both the mental and physical health of humans [1]. There are various sources of sound pollution like vehicle sounds, loud microphones, firecrackers, etc. Sounds from construction sites are one such source. This is a global problem considering the fast-paced Worldwide urbanization, that has been catalyzed by advanced machines. Construction sounds are not just incidental sounds that are a result of development but they also exemplify the kernel of revolution and progression. The impact of construction sounds on the working productivity and contentment of places like schools, colleges, hospitals, libraries, and research centers cannot be undervalued. High levels of construction sounds near educational institutions disturb in-progress lectures and limit the overall academic performance of a student. Medical care institutions require a peaceful environment surrounding their location so high levels of construction sounds not only disturb the recuperation of the patient but also significantly compromise the efficacy of surgeries that demand undivided concentration. Systems that can automatically detect such sounds can be very helpful in this context. Such systems can aid in the automatic deployment of countermeasures like alarms, automatic shutting of windows, and deployment of mobile soundproofing solutions to minimize the effect of noise on use. Such systems can also help in assessing the extent of construction noise in a region prior to procurement of properties. This can be very helpful for people who have serious health problems where such sounds can amplify their ailments.

2 Related Works

There have been several initiatives for differentiating and identifying construction sounds. Moniruzzaman et al. [2] presented a system for evaluating the noise pollution levels of rush-hour hotspots in Khulna City, Bangladesh. The dataset encompassed location assessments of noise levels which were recorded at different times and

gathered by means of a sound level meter (Multi-function Environmental Meter, Model: RH87). Various noise pollution indices were calculated such as Noise Climate (NC) and Traffic Noise Index (TNI) by the use of Gaussian percentiles. Inverse Distance Weighted (IDW) was used to develop noise maps that illustrated the noise level distributions in the study area. Shibbari intersection, North Kashipur, and New market recorded the highest noise levels of 83.2 dBA, 76.2 dBA and 79.6 dBA, respectively. Surveys carried out among locals disclosed that noise levels in the industrial areas had harmful effects such as disturbance of sleep(36%), headaches (23%), hearing problems (21%), and irregular heartbeat (20%). Sojol et al. [3] built a two-tiered arrangement having the final goal of reducing the harmful effects of noise pollution on human health. The used dataset comprised data obtained from the Department of Environment in Bangladesh. The arrangement made use of Arduino Uno, Wi-Fi modules, sound sensors, and microcontrollers. The first phase of the arrangement displayed sound levels and suitable memos to create awareness among the masses, whereas the second phase dealt with regulating the noise levels by the use of a controlled honking system reducing unnecessary honking.

Angulo et al. [4] crafted an architecture aimed at evaluating the capabilities of the latest urban sound event detection systems and analyzed the effect on the system's performance by factors like polyphony and sound proximity. The datasets used included SONYC-UST-v2 [5], SINGA:PURA [6] labeled set, and SINGA:PURA unlabelled set. Convolutional recurrent neural network (CRNN) and embedding classifiers with Patchout faSt Spectrogram Transformer (PaSST) and openL3 were utilized for the experimentation purpose. Best accuracy scores of 80.8% and 60.1% were achieved in the case of the CRNN system with the per-channel energy normalization (PCEN) [7] front-end. Ezhilarasi et al. [8] have proposed a system of monitoring air and noise pollution by the usage of embedded computing and sensory devices. The measured sound level was notified to the online server through the Internet of Things, and the obtained data was made available to the end user via Wi-Fi.

Deshmukh et al. [9] put forward a system aimed at providing a versatile and cost-efficient system that specifically monitored air and sound pollution by means of the Internet of Things (IoT). The data was collected in real-time by sensors and was transmitted over the Internet with a Wi-Fi module (ESP8266). A microcontroller was also used for data processing. The system was tested at M.G Road in Pune where observed values of carbon monoxide, carbon dioxide, and sound levels were recorded as 44%, 48%, and 36%, respectively. The results achieved were near to that obtained by standard devices exhibiting the efficiency of the system. Tsao et al. [10] presented a system to measure and better envision noise pollution in expanded environments by applying a distributed sound sensing system (DSSS) employing the Internet of Things. An Ameba RTL 8195AM as well as a microphone module was used for the construction of the DSSS while C programming language was used for the development of the system's firmware. Also, PHP was used for the development of a clouding platform that would receive data from several DSSS devices. The system was been successfully set up in the library of National Quemoy University in Kimmen.

Kumbalkar et al. [11] created a system that provided real-time data on air and sound quality. The system included components such as a Node MCU controller, air sensors, Wi-Fi and sound detector module, LCD display, and a mobile application allowing users to track pollution data on cell phones. Alerts are initiated when pollution surpassed preset levels. Saha et al. [12] created a framework that could monitor temperature, humidity as well and air and sound pollution levels. The data was collected in different locations through the LM393 sound sensor, MQ135 gas sensor, and the DHT11 temperature humidity sensor. The sensors were integrated with the Raspberry Pi acting as a core processing unit from where the collected data was transferred for deeper analysis and supervision. The framework successfully presented a consistent tracking of the parameters and identified any unusual or hazardous conditions. Manglani et al. [13] developed a system that consistently kept track of the air and noise pollution levels in an area. The sensors collected the data measurements of gases such as NH_3 and CO_2 and the sound intensity levels in decibels. Air sensor MQ135 and noise sensor were integrated with an Arduino microcontroller. An alert was activated, indicated by a buzzer when the predefined limits of pollution were crossed. Rafi et al. [14] presented a system for the assessment of sound levels in a city. The sound level data were collected spanning over 30 distinct locations in Dhaka, Bangladesh, by the use of the Lutron SL-4012 sound level meter during peak traffic hours. Further, a sound pollution heatmap was provided which portrayed a categorization of the recorded sound levels as per the World Health Organization's standards over Dhaka. Additionally, a device-based sound metering and awareness messaging system was proposed which could contribute to raising public awareness and reducing sound pollution.

3 Dataset Description

The dataset encompasses diverse sounds that have been sorted into two categories. These categories aid in illustrating and depicting the sophisticated array of noises that are present in urban. The first category consists of construction sounds that have become an integral part of our daily lives and are an essential part of the Urban fabric. This category consisted of different sounds like heavy machinery sounds, drilling sound, concrete mixing sound, backup generator sound, safety whistles sound, compaction and hydraulic equipment sounds, welding sounds, hammering and nailing sounds, clanging of metal sound, sawing sound, demolition sounds, road paving sound, tunneling, and boring sounds. The second category consisted of an extensive selection of additional ambient sounds ordinarily experienced in a metropolitan atmosphere. These sounds consisted of both indoor and outdoor sounds like car honks, animal sounds, kitchen sounds, conversation, rain, and thunder sounds. Both the classes (Construction and Rest) were composed of these sounds with clips of 5 seconds. The construction class consisted of 3179 clips while the rest class consisted of 2896 clips. It was noted that both classes had commonalities in terms of ambient noise, conversations, airplane sounds, etc. Such characteristics were not removed

Fig. 1 Wordcloud of constituent sounds of **a** construction class and **b** rest class

to ensure real-world conditions. The word cloud of the constituting sounds of both classes is presented in Fig. 1.

4 Proposed Method

4.1 Feature Extraction

An audio demonstrates high variation throughout its length which poses difficulty in analysis. To tackle this problem, the audio is split into shorter sub-clips (frames) to ensure pseudo-stationarity. At times, framing leads to loss of continuity which leads to jitters during analysis in the frequency domain. Thus, the frames are subjected to windowing to address this issue. In this experiment, Hanning window [15] was used for this purpose which is illustrated as follows:

$$w(n) = 0.5 * \left(1 - cos\left(2\pi \frac{n}{N}\right)\right) \tag{1}$$

where length of window = $N + 1$ and $0 \leq n \leq N$.

The audio clips were split into frames of different lengths from 512 samples to 1792 samples. Standard Mel Frequency Cepstral Coefficients (MFCC) were extracted for each of these clips at frame level. MFCC involves the use of the Mel frequency scale that is based on the idea of how humans recognize and understand the variances in frequencies of sound [16]. The audio signal is converted from the Hertz (Hz) scale to the Mel scale. The Mel scale corresponds to how humans perceive sound. The relationship between the Hertz and Mel scale is represented as:

$$m = 1127 * \log\left(1 + \frac{f}{700}\right) \tag{2}$$

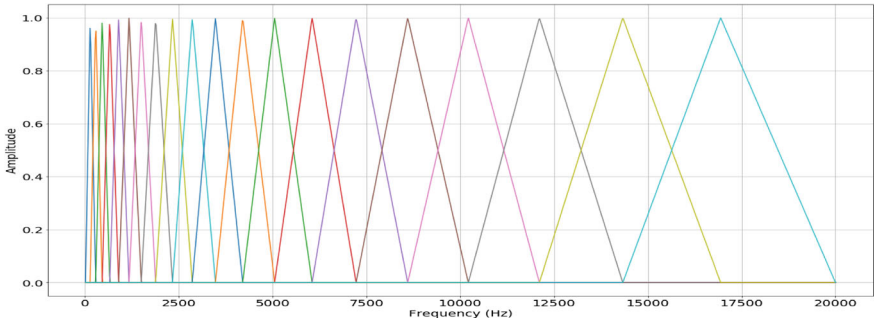


Fig. 2 Filter bank of 20 Mel filters

where m symbolizes the value on the Mel scale and f expresses the frequency in Hertz. A filter bank of 20 Mel filters in the range of 20 Hz to 20, 000 Hz is presented in Fig. 2.

4.2 Machine Learning-Based Classification

Machine learning can be referred to as the branch of artificial intelligence (AI) that is particularly inspired by the human brain to learn and evolve from experiences. It is concerned with the development and deployment of algorithms that make use of pre-existing data also known as training data to generate output. The output thus obtained helps in making predictions and decisions on test data. An increase in the amount of training data leads to an increase in the output efficiency of the machine learning model in most cases.

Ensemble learning is a type of machine learning that merges the results obtained from several individual models commonly of identical or diverse types to develop a more potent and enhanced predictive model. The random forest is a classifier that is based on this technique. Random forests are used in both regression and classification tasks. The root concept of random forest lies in the fact of building up several decision trees and merging the predictions obtained to make the results more accurate and efficient. It can be used to calculate the significance of each feature and is capable of handling absent values in the dataset. Over-fitting is minimized in random forests as irregularity/randomness is incorporated at the time of data sampling as well as during feature selection.

The random forest algorithm includes the following steps:

- **Ensemble of decision trees**: A group of decision trees is created that work in collaboration with each other. Each of these trees contributes toward making predictions.

- **Random sampling**: In the course of training, different random samples of the data are taken which is then followed by choosing certain data points and utilizing them to build a tree. Each tree is trained on a set of samples.
- **Random selection of feature**: During the construction of each decision tree only, a number of features are chosen from all the available features. This contributes to making the constructed trees distinct from each other.
- **Classification and Regression**: In classification, the final predicted outcome is the most typical prediction from all the trees. In regression, the final predicted outcome is the average of all the tree predictions.
- **Enhanced accuracy**: The generated decision trees are combined, using random samples and random features being selected. This reduces over-fitting and contributes to a random forest's ability to handle noisy data.

5 Results and Analysis

Each of the feature sets was subjected to the random forest-based classifier. 70% of the data was used for training while the remaining was used for testing. The obtained accuracies for the disparate feature sets are presented in Table 1.

It is noted that the best performance was obtained for the 25 dimensional features computed using a window length of 1280 samples. The confusion matrix for this setup is presented in Table 2.

Table 1 Performance of different feature sets

	512	768	1024	1280	1536	1792
5	95.2	95.09	95.2	95.37	95.64	95.64
10	96.97	97.24	97.52	97.68	97.46	97.79
15	96.91	96.97	97.41	97.68	97.74	97.74
20	96.91	96.91	97.02	97.35	97.68	97.68
25	96.69	96.86	97.46	**97.85**	97.63	97.57
30	96.8	97.3	97.35	97.52	97.57	97.68

The columns represent the window lengths while the rows represent the feature dimensions The best performance is highlighted with bold

Table 2 Confusion matrix for random forest with 50 estimators on the 25 dimensional features computed using a window length of 1280 samples

	Construction	Rest
Construction	978	17
Rest	22	797

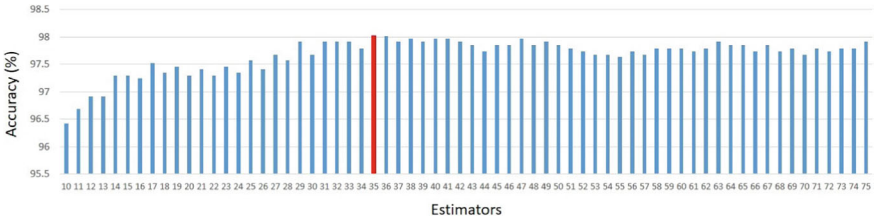


Fig. 3 Performance for the different number of estimators

Table 3 Confusion matrix for random forest with 50 estimators on the 25 dimensional features computed using a window length of 1280 samples with 35 estimators

	Construction	Rest
Construction	979	16
Rest	20	799

Next, the number of estimators varied from 10 to 100, and a higher accuracy of 98.01% was obtained for both 35 and 36 estimators. The variation of accuracy for the different estimators is presented in Fig. 3.

The confusion matrix for this setup is presented in Table 3. In this setup, an improvement of 0.17% was observed in the overall accuracy. The number of misclassifications was reduced by 1 and 2 for construction sounds and other sounds, respectively. The misclassified instances were analyzed in detail, and it was noted that some of the clips in construction sound had similarities with different indoor sounds like mixer grinder, oven alarms, and forceful door closing. Certain construction sounds also had distinct conversations which were also misclassified.

The performance of several other classification techniques, namely Naive Bayes [17], SVM [18], KNN [19], decision tree [20], and AdaBoost [21], was also compared on the best-performing features whose results are presented in Fig. 4. Both in terms of the average accuracy and the other sound class accuracies, random forest held the top spot. In terms of the construction sound class, KNN produced a slightly higher accuracy of 98.79% which was 0.0.41% higher than random forest. However, in the case of other sounds, the recognition rate significantly dropped by 5.14% as compared to random forest. This was followed by SVM, where a reduction of 0.1% and 0.99% was observed for the construction and other sounds, respectively. The lowest performance was obtained for Naive Bayes which was almost 16.71% less than that of random forest in terms of the average accuracy. The reduction of recognition rate was higher for the other sound class as compared to the construction sound class.

Fig. 4 Performance for the different classifiers

6 Conclusion

In this paper, a system is presented to distinguish construction site sounds from acoustic surroundings in the urbanized landscape. Experimentation was conducted with construction sounds along with an assortment of familiar sounds occurring in the urbanized landscape. The system was successful in distinguishing the 2 classes with an accuracy of 98.02% using MFCC-based features coupled with random forest-based classification. In subsequent time, the framework will be evaluated on a dataset of increased size and variety in terms of the type of sounds. The dataset will also be exposed to additional noise to test the reliability of the framework. Tests will be carried out with manually designed attributes for enhanced productivity in modeling the audios to build a handcrafted feature-based architecture. The architecture will be developed further to analyze as well as identify subtle distinctions among various kinds of construction sounds. Eventually, the machine learning-based architecture will be enhanced to develop an airier framework to ease the deployment process in settings with a limited number of resources which will additionally contribute to instantaneous and uninterrupted computation. Finally, deep learning-based approaches will also be explored to enhance the efficacy of the proposed system.

Acknowledgements This research was carried out on the High-Performance Computing resources at New York University Abu Dhabi. This work was supported by the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001.

References

1. https://www.medicalnewstoday.com/articles/noise-pollution-health-effects#mental health: Effects of noise pollution on health. Online. Accessed on (01012024)
2. Moniruzzaman S, Rahman MM, Arifuzzaman M, Gazder U (2021) Study on noise pollution in some selected areas of khulna city area. In: 2021 International conference on decision aid sciences and application (DASA). pp. 73–79. IEEE

3. Sojol JI, Shihab MH, Ahamed T, Siam MAS, Siddique S (2019) Nirob: a next generation green earth technology based innovative system for metropolitan sound pollution management. In: 2019 21st international conference on advanced communication technology (ICACT). pp. 722–727. IEEE
4. Angulo F, Essid S, Peeters G, Mietlicki C (2023) Cosmopolite sound monitoring (cosmo): a study of urban sound event detection systems generalizing to multiple cities. In: ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 1–5. IEEE
5. Cartwright M, Cramer J, Mendez AEM, Wang Y, Wu HH, Lostanlen V, Fuentes M, Dove G, Mydlarz C, Salamon J et al (2020) Sonyc-ust-v2: an urban sound tagging dataset with spatiotemporal context. arXiv preprint arXiv:2009.05188
6. Ooi K, Watcharasupat KN, Peksi S, Karnapi FA, Ong ZT, Chua D, Leow HW, Kwok LL, Ng XL, Loh ZA et al (2021) A strongly-labelled polyphonic dataset of urban sounds with spatiotemporal context. In: 2021 Asia-pacific signal and information processing association annual summit and conference (APSIPA ASC). pp 982–988. IEEE
7. Wang Y, Getreuer P, Hughes T, Lyon RF, Saurous RA (2017) Trainable frontend for robust and far-field keyword spotting. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5670–5674. IEEE
8. Ezhilarasi L, Sripriya K, Suganya A, Vinodhini K (2017) A system for monitoring air and sound pollution using arduino controller with iot technology. Int Res J Adv Eng Technol (IRJAET) 3(2):1781–1785
9. Ioit A, Aissms I (2017) Air and sound pollution monitoring system using iot. Int J Recent Innov Trends Comput Commun 5(6):175–178
10. Tsao YC, Su BR, Lee CT, Wu CC (2017) An implementation of a distributed sound sensing system to visualize the noise pollution. In: 2017 International conference on applied system innovation (ICASI). pp. 625–628. IEEE
11. Kumbalkar G, Joshi N, Lonkar R, Darne S, Latare Y (2022) Iot based whether reporting with air and sound pollution monitoring system. Int J Res Appl Sci Eng Technol 10:372–375
12. Saha AK, Sircar S, Chatterjee P, Dutta S, Mitra A, Chatterjee A, Chattopadhyay, SP, Saha HN (2018) A raspberry pi controlled cloud based air and sound pollution monitoring system with temperature and humidity sensing. In: 2018 IEEE 8th Annual computing and communication workshop and conference (CCWC). pp. 607–611. IEEE
13. Manglani T, Srivastava A, Kumar A, Sharma R (2022) Iot based air and sound pollution monitoring system for smart environment. In: 2022 International conference on electronics and renewable systems (ICEARS). pp. 604–607. IEEE (2022)
14. Rafi MAI, Sohan MR, Jawad A, Sourid HR, Hossain MM (2023) Sound pollution monitoring system and awareness creation in modern cities: a case study. In: 2023 International conference on advances in electronics, communication, computing and intelligent information systems (ICAECIS). pp 637–642. IEEE
15. Testa A, Gallo D, Langella R (2004) On the processing of harmonics and interharmonics: using hanning window in standard framework. IEEE Trans Power Delivery 19(1):28–34
16. Mukherjee H, Halder C, Phadikar S, Roy K (2017) Read-a bangla phoneme recognition system. In: Proceedings of the 5th international conference on frontiers in intelligent computing: theory and applications: FICTA 2016, Volume 1. pp. 599–607. Springer
17. Webb GI, Keogh E, Miikkulainen R (2010) Naïve bayes. Encyclopedia of machine learning 15(1):713–714
18. Wang H, Hu D (2005) Comparison of svm and ls-svm for regression. In: 2005 International conference on neural networks and brain. vol 1, pp 279–283. IEEE (2005)
19. Guo G, Wang H, Bell D, Bi Y, Greer K (2003) Knn model-based approach in classification. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy. Proceedings. pp. 986–996. Springer (2003)

20. Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD (2004) An introduction to decision tree modeling. J Chemometr J Chemometr Soc 18(6):275–285
21. Ying C, Qi-Guang M, Jia-Chen L, Lin G (2013) Advance and prospects of adaboost algorithm. Acta Automatica Sinica 39(6):745–758

IoT Meal Master Using Augmented Reality

Mayank Chadha, Richa Gupta, Jean Chu, Lakshika Vaishnav, and Aung Chan

Abstract The Internet of Things (IoT) is becoming more and more ubiquitous as its scope is expanding each day. Over the years we have seen many interesting implementations of this technology ranging from robotics to smart wearables and homes. It is interesting to note that despite having a multitude of IoT applications in the kitchen, there are not many when it comes to cooking and guidance. Traditionally, we scroll through a ton of recipes on the internet, and it gets confusing and boring to read through the list. Also, it becomes difficult to follow along as there is a lot of dependency on the weight of the ingredients and the quantities that we want to prepare. Our solution proposes the use of IoT and Augmented Reality (AR) to assist individuals while cooking and guide them during each step of the process. This paper will design a smart weighing scale based on load cells and an AR application that communicates with each other in real time. This allows the user to follow along the recipe and prepare accurate proportions in an immersive way as opposed to the traditional methods.

Keywords Internet of Things · Augmented Reality · ESP32

M. Chadha (✉)
Northeastern University, Boston, MA 02115, USA
e-mail: chadha.may@northeastern.edu

R. Gupta
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

J. Chu
Carnegie Mellon University, Pittsburg, PA 15213, USA

L. Vaishnav
State University of New York, Albany, NY 12207, USA

A. Chan
Strategy First University, Yangon 11111, Myanmar

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_41

1 Introduction

In our ever-evolving society, constant advancements in technology have paved the way for enhanced efficiency and safety. As we explore new horizons, interoperability have become a key focus and technologies like internet of things and immersive technologies such as augmented reality are at the forefront, bridging the gap between physical and virtual environments. These groundbreaking technologies transcend the limitations of traditional screens, offering users an unprecedented level of interaction with advanced graphics and integration with the physical world. In fields such as healthcare, AR proves to be a game-changer by providing real-time information during surgeries from sensors embedded on the devices and offering innovative training methods for medical students. In education, interactive learning experiences and virtual field trips are revolutionizing the way students absorb information. For architecture and construction, AR facilitates visualization and project planning, contributing to business growth.

Even in our daily lives, we come across many emerging applications such as home automation or smart kitchen. By making our homes smarter, we increase our comfort and quality of life [1]. Papers [1, 2] proposes the use of low power wireless communication protocols like ZigBee, Wi-Fi, or Bluetooth. Leveraging these components, we embark on a revolutionary journey to transform the heart of our homes—the kitchen.

No longer are the days of flipping through recipe books or scrolling through various websites to master the culinary arts. As of now, most systems are focused on safety aspects, such as detecting gas leakage inside kitchens or during cooking [3, 4] but a very few focus on the actual process of cooking. Papers [5, 6] presents the use of IoT for pet feeder and augmented reality for education respectively. Majil et al. [7] focuses on the use of augmented reality with computer vision technology to identify food ingredients and guide at every step of cooking the solution is a great implementation but a major concern while cooking is the measurement of ingredients which cannot be done without any external hardware. Our initiative pioneers the use of IoT-based load cell and augmented reality to create a unique cooking experience that connects friends and family. In this innovative study, we delve into the possibilities of connecting individuals through the virtual and real realms, providing a platform for real-time interaction.

2 Literature Review

Augmented Reality is a user interface metaphor that weaves a connection between digital world and physical spaces [8]. Over the last few years, augmented reality started using internet, playing with the physical world and its integration with IoTs have enhanced its application for health care, education, training, architecture, game development, and many more other fields [9]. This review of literature will take you around the different aspects of augmented reality and its applications. Also, the

review will talk about the integration of IoT's that has led to developments of smart houses, smart kitchens, and smart cities and enhanced the use of AR in the real world.

Connection and need for communication between families and food culture necessitates the need for modern technological means such as IoT and augmented reality technology. In the study [10] designed a prototype that in-corporates culinary elements and remote home control technology for better Intergenerational communication; further author discusses about incorporating IoT technologies for immersive and interactive cooking experiences enhancing family bonds and communications. Intelligent-Assistive Technologies supporting people with mild cognitive impairments in their day-to-day activities, expanding the research by using potential of artificial intelligence [11] and augmented reality that will further enhance the capabilities of assisting people. Use of modified artificial methods integrated with mathematical model that consists of image & sound processing including interface construction; this aims to increase extract selected features and improve human–machine interaction where they use AR visualization as a human–machine communication interface and built in sensors and data exchange between applications to locate and enhance system functionality, model includes applications such as looking for objects in the kitchen, small children to find their parents displaying the location on their screens and smart technologies in education and learning [12].

As mentioned, augmented reality can enhance the learning experience with visualization of complex concepts and interaction with virtual objects, another high-raising use of augmented reality is into gamification and game design [13]. 'Pokemon go' is an augmented reality smartphone game that superimposes characters of Pokemon into real-world and gives the user the ability to experience real-world wildlife watching and natural history. Grubert et al. [14] delves into the notion of "Pervasive AR," highlighting its significant strides within the augmented reality realm. They emphasize how this advancement fosters more engaging and personalized user interactions by seamlessly integrating contextual cues such as location, identity, activity, and time. Their approach involves crafting a nuanced classification system for context awareness, which not only enhances the user experience but also prioritizes social and privacy considerations.

In the paper, we will show the experiment where users can use augmented reality with IoT appliances. AR demonstrates the easy way to quickly debug the devices and a seamless way to interact with IoT services in local environment such as home or buildings. From D. Jo's study in 2016, we glean insights into creating an optimal environment for assigning objects and integrating augmentations, particularly focusing on controlling home appliances and enhancing interactivity within the Internet of Things (IoT) framework [15].

3 System Architecture

Figure 1 denotes the complete architecture proposed for the application. Using the nomenclature from the book [16], the access side refers to where the sensing takes place, while the core side refers to where the application runs. Between the two, there is an edge that acts as a gateway for communication between both systems. The system comprises of components and subcomponents and changes the state over time. Below are the descriptions of the access side as well as the core side of the system.

3.1 Access-Side Components

The access-side system consists of two major components. One is the sensing component, and the other one is the smart device. The sensing component is responsible to detect the measurement and the smart device is responsible to calibrate and send data to different devices. Below is the detailed information on each of the component.

The load cell is used as the sensing element which is a strain gauge that generates current when put on mechanical loading. The current is minimum and hence an external amplifier is required. For our purposes, we used the Hx711 amplifier which is a popular selection for making load cell. After that, the values are calibrated using known weights as shown in [17]. Also, an interesting aspect of the sensor is it does not support any standard protocols like I2C or SPI which are often used to integrate various hardware devices. We followed the official documentation for the sensor for communicating with the smart device.

The smart device is an ESP32 Microcontroller that consists of a low powered WIFI Enabled device [18]. The device calibrates the raw data received from the sensor and transmits them. Over WIFI with the help of IoT protocols. Telemetry

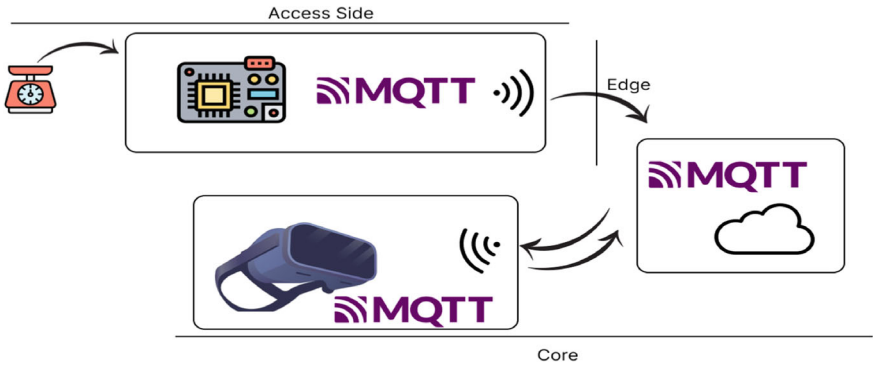


Fig. 1 System architecture

of the data is done using MQTT protocol, which is based on Publish–Subscribe architecture [19]. MQTT is an application layer protocol based on the 5-layer Open Systems Interconnection Model (OSI). The publishers publish on specific node called as topic and if the subscriber is subscribed to the topic, it will receive any messages. The smart device that acts as a MQTT publisher that publishes data to the broker on specific topic.

3.2 Core-Side Components

The edge sits right between the access and the Core side. The core side consists of an MQTT broker, the augmented reality application. We use the Mosquito broker which is an implementation of MQTT protocol [20] and hosted on cloud. Here the MQTT broker is hosted on the cloud that acts as a medium to communicate between the broker and its subscribers/publishers.

For implementation of augmented reality, we are using Oculus Quest 3 which provides the developers with a lot of features for both augmented and virtual reality like. We utilized features like pass-through and hand gesture tracking. For our purpose, we developed the application using the meta presence platform [21], a set of SDK tools provided by meta for making development more accessible. The Oculus Quest 3 also comes with the controllers to navigate inside the AR application and can also supports gestures. The AR application consisted of a MQTT subscriber that subscribes to the topics from the smart device.

4 Methodology

Software and hardware are a key to evolution of enhanced augmented reality user experience. There are variety of API that makes it possible to create a fascinating AR user experience, making use of mobile technologies. The new APIs introduced by Apple and Google pave a way to increase in augmented reality; head-mounted displays such as Google Glass, Microsoft HoloLens and Magic Leap Light wear allows hands free interaction with IoT services as well as products. Recent work provides a scalable augmented reality framework enables by an extension to IoT infrastructure: AR IoT. The study implemented a guided recognition and tracking tool for AR using methods like feature-based tracking and model-based tracking; furthermore, study highlights use of AR as an interface for interacting with IoT appliances, and hence, confirms that it is preferred by the users.

The user uses the augmented reality application on a virtual reality headset. Figure 2 displays the user interface of the application that contains a recipe selection panel along with the difficulty level of the recipe. Next, the user selects the recipe from the panel of options that display the recipe name, image, and the difficulty

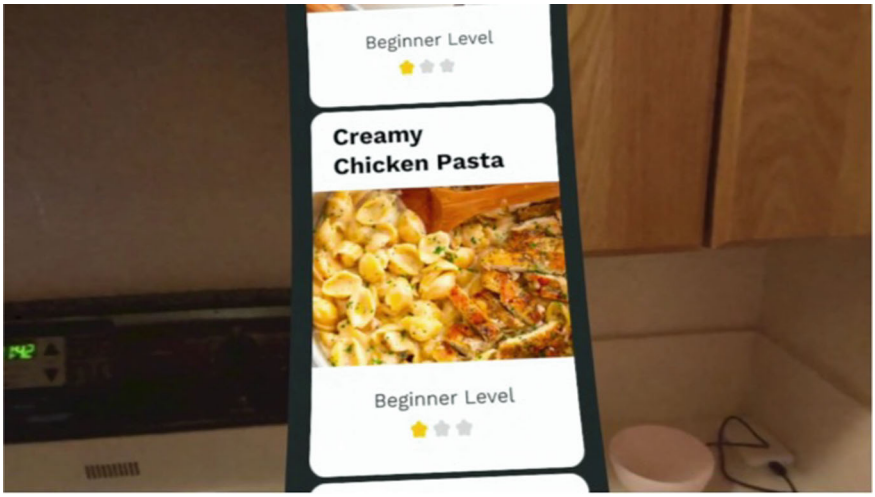


Fig. 2 Augmented Reality dashboard displaying different recipes

level. Once the recipe is selected, the user then collects all the ingredients necessary as provided on the virtual dashboard and select the quantity it needs.

The next step is to measure the ingredients. Once the ingredients are collected and marked as ready, the next step is to measure the weight of each ingredient that is provided on the dashboard. For this, we will use the load cell as discussed previously. The load cell and the smart device gets the weight of each ingredient and sends it using MQTT protocol. Figure 3 depicts the ingredients, the smart device and the load cell to measure the weight of ingredients. The smart device is connected to the internet using WIFI and acts as a publisher that published values on a specific topic, i.e., /device/loadcell/value. The values then are published on the cloud hosted broker which handles all the information about the topics.

The device that acts as another client, subscribes to the same topic that is publishing the data, i.e.,/device/loadcell/values on the broker. The subscriber then starts listening to the data published by the load cell. When the data is received by the headset, it compares the weight received from the load cell with the weight instructed on the recipe. It then instructs the user to reduce or increase the weight based on the requirements. When the augmented reality application receives the information from the access side, it displays the weight of the ingredients and suggests adjusting the quantities based on the recipe requirements. Furthermore, it provides the user with the option to increase the proportions, such as when cooking for many people, quantities of ingredients are adjusted automatically for ease of use. The application then guides the user with necessary resources or video references to follow each step until the recipe is prepared.

The Oculus Quest 3 comes with the controllers to navigate through options inside the augmented reality application. However, it is not feasible to use the controllers while cooking as hands might be greasy or busy with chopping or cooking. As a

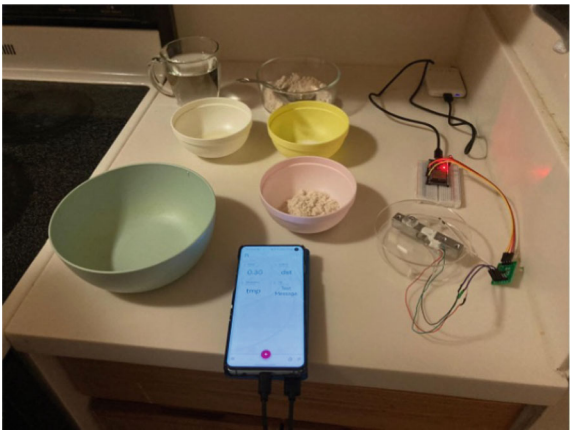


Fig. 3 The load cell and ESP32 microcontroller measuring weight of the ingredient

result, we integrated the speech to text options using whisper APIs to convert speech to text for searching the web. Whisper is a language model developed by Open AI for speech to text conversion. Open AI has provided APIs to integrate with other applications. Figure 4 presents the user screen to select the microphone from a panel of options and starts recording the voice. The virtual reality application records the voice and then sends the audio file to the whisper API's which returns the text for the same.

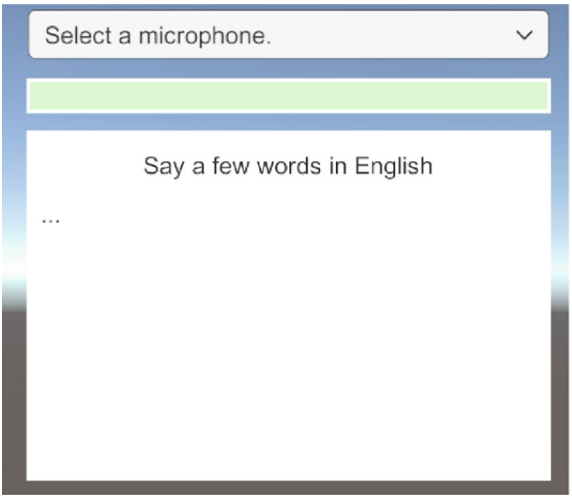


Fig. 4 User interface for speech-to-text conversion using whisper API's

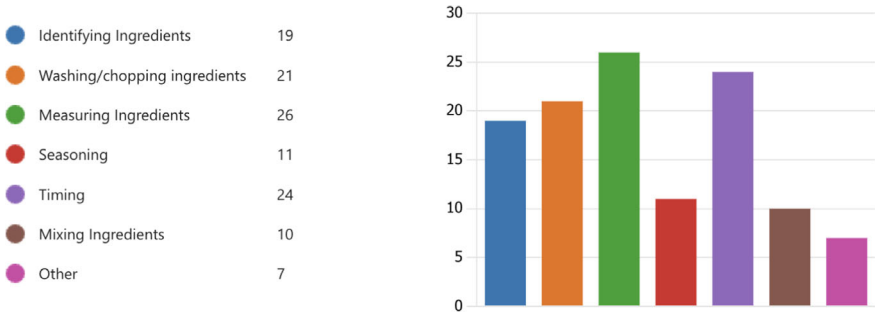


Fig. 5 Survey graphs

5 Result

Although there were a multitude of solutions based on augmented reality and IoT, there were very limited when it comes to cooking. An augmented reality application was also inefficient independently as cooking dependence is with the weights and traditional weighing scales are incapable of being used for this application.

We also conducted a small survey among a small group of participants, for which a significant majority expressed challenges in one of the key processes of cooking. The primary difficulty reported by the respondents was related to measuring ingredients, followed by difficulty in scrolling through recipes on the internet or watching video. Traditional ways of scrolling through blogs of recipes can get confusing especially when clicking our phones with greasy fingers. The application has kept all the features in mind required to make the cooking process as immersive and easy as possible.

Figure 5 depicts the survey conducted among participants of different age groups highlighting the challenges faced by them during the cooking process. Also, in Fig. 6, we can get the idea of the environment the application will be used in. This contains the weighing scale measuring the weight of ingredients and the user using the application deployed on the headset utilizing the features for cooking assistance. Furthermore, with the help of the augmented reality application and the device, participants showed enhanced proficiency and confident in cooking a variety of new dishes in a fun and immersive way. The application followed all the standards for communication, hardware, and measurements. This allowed the user to use the application with low latency, hands free, and in a fun way.

Fig. 6 A person wearing the headset with the weighing scale being used for measurement

6 Conclusion

Augmented Reality (AR) stands on the pillars of seamless integration between the virtual and real worlds, real-time interaction, and advanced 3D-based tracking. This project envisions a kitchen where the boundaries between the physical and virtual dissolve, allowing friends and relatives to share a cooking adventure in real time. Our application becomes a dynamic teacher, guiding users through each step of the cooking process. Gone are the isolated cooking experiences; now, the kitchen becomes a communal space where loved ones can come together, despite physical distances. IoT transforms cooking into an engaging and collaborative activity. Users can follow along with the virtual guide, receiving instructions, tips, and tricks in real time. This project not only simplifies the cooking process but also adds a layer of fun and togetherness, turning each culinary endeavor into a shared experience. Furthermore, the project was presented at the Massachusetts Institute of Technology for the Reality hackathon that took place in the month of January 2024 and was tested for a variety of recipes and proportions.

In summary, harnessing the power of augmented reality to redefine how we approach cooking in our homes. Through the seamless connection of virtual and real worlds, real-time interaction, and 3D-based tracking, this project introduces a new era in culinary exploration, bringing friends and family together around the kitchen, guided by the whimsical charm of our virtual guide.

7 Limitations and Future Work

There are certain challenges of the project that were either due to the limitations of the hardware or some features that would complement and further enhance the usability of the project.

- Currently, the virtual reality application requires the need of a virtual reality headset that supports augmented reality. Alternate approach will be to use the augmented reality glasses which are light weight and perfectly fits the needs.

However, certain limitation with the hardware prevents the UI elements of the application from rendering.

- The recipes in the virtual dashboard are static and stored inside the application. These can be stored outside the application using a database that would also allow adding many recipes.
- It would also be interesting to see how we can build a recommendation algorithm to recommend recipes based on user preference or based on the ingredients available with the help of machine-learning techniques.
- Allowing the application to provide nutrient information to monitor nutrient information like protein or calorie intake can enable a wide range of usability and further enhance user experience.

References

1. Asadullah M, Raza A (2016) An overview of home automation systems. In: 2016 2nd international conference on robotics and artificial intelligence (ICRAI), Rawalpindi, Pakistan, pp 27–31. https://doi.org/10.1109/ICRAI.2016.7791223. keywords: {Home automation; Home appliances; Bluetooth; GSM; ZigBee; Speech recognition; Home Automation System; Bluetooth; GSM; ZigBee; EnOcean; Internet of things (IoT)}
2. Pavithra D, Balakrishnan R (2015) IoT based monitoring and control system for home automation. In: 2015 Global conference on communication technologies (GCCT), Thuckalay, India, pp 169–173. https://doi.org/10.1109/GCCT.2015.7342646. keywords: {Home automation; Smart phones; Servers; Fires; Monitoring; Home appliances; IEEE 802.11 Standard; Raspberry pi; Internet of Things; Web Server; Home Automation}
3. Palandurkar VR, Mascarenhas SJ, Nadaf ND, Kunwar RA (2020) Smart kitchen system using IOT. Int J Eng Appl Sci Technol (IJEAST) 04:378–383. https://doi.org/10.33564/IJEAST.2020.v04i11.067
4. Nugroho F, Pantjawati A (2018) Automation and monitoring smart kitchen based on internet of things (IoT). In: IOP conference series material science engineering, vol 384. pp 012007. https://doi.org/10.1088/1757-899X/384/1/012007
5. Wu W-C, Cheng K-C, Lin P-Y (2018) A remote pet feeder control system via MQTT protocol. In: 2018 IEEE international conference on applied system invention (ICASI), Chiba, Japan, 2018, pp 487–489. https://doi.org/10.1109/ICASI.2018.8394292. keywords: {Cameras; DC motors; Automobiles; Feeds; Protocols; IP networks; Raspberry Pi; MQTT protocol; GPIO; IP camera}
6. Chen X, Wang S (2011) The virtual reality technology application in education. Software Tribune (education technology) 12:76–78
7. Majil I, Yang MT, Yang S (2022) Augmented reality based interactive cooking guide. Sensors (Basel) 22(21):8290. https://doi.org/10.3390/s22218290. PMID:36365988;PMCID:PMC9655470
8. Feiner S (1994) Redefining the user interface: augmented reality. In: Proceedings ACM SIGGRAPH, course notes, vol 2. pp 1–18
9. Dorward LJ, Mittermeier JC, Sandbrook C, Spooner F (2017) Pokémon go: benefits, costs, and lessons for the conservation movement. Conservat Lett 10(1):160–165. https://doi.org/10.1111/conl.12326
10. Huang Y, Chen Q, Xu L, Wu Z, Fan M (2023) Toward leveraging augmented reality (AR) for enhancing remote intergenerational communication in cooking scenarios. Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, Computational Media and Arts Thrust

11. D'Agostini J et al. (2018) An augmented reality virtual assistant to help mild cognitive impaired users in cooking a system able to recognize the user status and personalize the support. In: 2018 Workshop on metrology for industry 4.0 and IoT, Brescia, Italy, pp 12–17. https://doi.org/10.1109/METROI4.2018.8428314. keywords: {Wastewater; Sensor systems; Gas detectors; Sensor arrays; Principal component analysis; Temperature sensors; Intelligent Assistive Technologies; Augmented Reality; Action Recognition; Object Localization; Sensor Fusion}
12. Połap D (2018) Human-machine interaction in intelligent technologies using the augmented reality. J Inform Technol Control 47(4):691–703. https://doi.org/10.5755/j01.itc.47.4.21602
13. Alexander M, Sandra T, Artur L, Manfred T (2017) Pokémon WALK: persuasive effects of Pokémon GO game-design elements. pp 241–252. https://doi.org/10.1007/978-3-319-55134-0_19
14. Grubert J, Langlotz T, Zollmann S, Regenbrecht H (2017) Towards pervasive augmented reality: context-awareness in augmented reality. IEEE Trans Visualization and Comput Graphics 23(6):1706–1724. https://doi.org/10.1109/TVCG.2016.2543720
15. Jo D, Kim GJ (2016) ARIoT: scalable augmented reality framework for interacting with internet of things appliances everywhere. IEEE Trans Consum Electron 62(3):334–340. https://doi.org/10.1109/TCE.2016.7613201
16. "Fundamentals of IoT Communication Technologies" (2021) Herrero, 1th edn. Springer-Nature
17. https://randomnerdtutorials.com/esp32-load-cell-hx711/. Last Accessed 22 Dec 2023
18. Espressif ESP32 (2024). https://www.espressif.com/en/products/socs/esp32. Last Accessed 03 Feb 2024
19. MQTT Homepage (2024). https://mqtt.org/. Last Accessed 03 Feb 2024
20. Mosquitto Homepage (2024). https://mosquitto.org/. Last Accessed 03 Feb 2024
21. Meta Quest Developer (2024). https://developer.oculus.com/presence-platform/. Last Accessed 06 Apr 2024

PQS-AKA: Post-Quantum Secure Lightweight Authenticated Lattice-Based Key Agreement for IoT Communications

Ajeet Singh, Appala Naidu Tentu, and K. Gangadhara Rao

Abstract As the Internet of Things (IoT) ecosystem expands, the need for secure and private communication over public networks becomes increasingly crucial. A fundamental component of this security paradigm is the establishment of a secret session key among IoT devices, enabling the secure transmission of confidential data over the Internet. In this paper, we introduce a Post-Quantum Secure Lightweight Authenticated Lattice-based Key Agreement (PQS-AKA) protocol, specifically developed for the rapidly evolving IoT ecosystem. Traditional Authenticated Key Agreement (AKA) protocols, predominantly based on concepts like Diffie–Hellman or prime factorization, are becoming obsolete due to their vulnerability to quantum computing advancements and their inefficiency in the resource-constrained IoT environment. Current quantum-resistant methods, although leveraging lattice cryptography for security, are impeded by issues such as high certificate management overheads and substantial communication and computational costs. PQS-AKA not only offers a defense against the threats posed by quantum computing through the inherent hardness of lattice problems but also simplifies the process by removing the need for extensive certificate management. The core security strength of our protocol is thoroughly evaluated under the random-natured oracle model, demonstrating its robustness against both present and future quantum computing attacks. Performance evaluations reveal that the proposed protocol offers a significant improvement over existing solutions in terms of efficiency and practicality, making it a viable and secure choice for industrial IoT applications.

Keywords Key agreement · Authentication · Lattice-based cryptographic security · Internet of things · Probable and provable security · Correctness proof

A. Singh · K. G. Rao
Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh 522 510, India

A. Singh (✉) · A. N. Tentu
C.R. Rao Advanced Institute of Mathematics, Statistics, and Computer Science, University of Hyderabad Campus, Hyderabad, Telangana 500 046, India
e-mail: ajeetcs@uohyd.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_42

1 Introduction

The rapid expansion of the Internet of Things (IoT) has been a driving force in the evolution of global communication networks, leading to an unprecedented interconnectedness of devices and systems. This growth, however, brings forth significant challenges in ensuring secure and private communication, particularly over public networks. A cornerstone of addressing these security concerns is the establishment of secret session keys among IoT devices, which are crucial for the secure transmission of confidential data over the Internet. Traditional Authenticated Key Agreement (AKA) protocols [1, 2], largely based on Diffie–Hellman and prime factorization methods, are increasingly found to be inadequate in the face of emerging quantum computing threats. These traditional methods, while foundational in the past, now fall short in providing the necessary security for the resource-constrained and highly dynamic IoT environment. To bridge this gap, our study introduces the Post-Quantum Secure Lightweight Authenticated Lattice-based Key Agreement (PQS-AKA) protocol. This protocol is specifically designed to counter the vulnerabilities posed by quantum computing, utilizing the complexity of lattice-based problems to offer a robust security solution. Moreover, PQS-AKA addresses the practical limitations of existing quantum-resistant methods [3, 4] by reducing certificate management overheads and cutting down on communication and computational expenses. This paper presents a comprehensive analysis of PQS-AKA, focusing on its security strengths under the random oracle model and its efficiency in real-world IoT applications, thereby marking a significant step forward in the realm of IoT security.

1.1 Varieties of Cryptographic Key Establishment Protocols

Cryptographic key establishment protocols encompass various methods used to securely create shared keys between entities. These protocols play a crucial role in securing communications by enabling parties to establish keys while ensuring confidentiality, integrity, and authenticity. One notable method is the Diffie–Hellman Key Exchange, which allows specific two parties to generate a shared secret-natured key over an insecure communication channel without directly exchanging the key. RSA Encryption, another widely used protocol [5], involves generating public and private key pairs for secure data transmission. Elliptic Curve Cryptography (ECC) leverages elliptic curves to create key pairs [6], offering robust security with shorter key lengths. Additionally, key exchange protocols within TLS/SSL, such as RSA, Diffie–Hellman, and Elliptic Curve Diffie–Hellman, are crucial in securing internet communications. Quantum Key Distribution (QKD) uses quantum mechanics to produce secure shared keys, while password-based protocols derive encryption keys from user-provided secrets. Symmetric cryptography, exemplified by algorithms like

AES, also employs key exchange methods [7] to establish secure channels using pre-shared secret keys. These diverse protocols serve as foundational elements in safeguarding sensitive data and ensuring secure communications across various systems and applications.

1.2 Contribution Highlights of The Paper

The core contribution highlights of this paper are as follows:

- This research work constructs a Post-Quantum Secure Lightweight Authenticated Lattice-based Key Agreement (PQS-AKA) protocol for IoT, offering enhanced security against quantum computing threats.
- Streamlined operational efficiency by eliminating extensive certificate management, reducing communication and computational overhead in IoT environments.
- PQS-AKA is validated under the random oracle model, demonstrating strong resilience against quantum attacks, and proved to be more efficient than existing key agreement solutions for IoT applications.

1.3 Related Work

Every lattice-based cryptographic system developed to date has inherent imperfections, notably a slight chance of decryption failure. This issue, particularly in the case of NTRU, has raised security concerns [8]. Since its introduction, various attacks targeting NTRU have been formulated [9], with the most recent ones exploiting its vulnerability linked to decryption failures. Nonetheless, the developers of NTRU have continuously enhanced the system to protect against such attacks [10]. NTRU's effectiveness and practicality have inspired numerous subsequent research projects. In 2002, an NTRU variant named CTRU was introduced in the work [11], using a polynomial ring instead of the base ring Z in NTRU. However, this system was quickly compromised [12]. Despite its shortcoming, CTRU paved the way for the concept of adapting NTRU to different rings. Pursuing this concept, another NTRU variant was proposed in [12], where Z was substituted with Gaussian integers.

Authors, in their work [13], have conducted a thorough examination of the impact of post-quantum cryptographic strategies in protecting IoT networks and investigate current research endeavors within this domain. Authors [14] have developed a two-party authenticated key agreement protocol based on identity, utilizing lattice-based cryptography. Authors, in their work [15] introduced an identity-based, two-party authenticated key agreement (ID-2PAKA) mechanism that enables secure communication and the exchange of confidential information between two parties within

IoT-enabled areas. Work by [16] presented an identity-based anonymous authentication and key agreement (IBAAKA) protocol tailored for Wireless Body Area Networks (WBAN) in a cloud-assisted setting, ensuring both mutual authentication and user anonymity. In the security analysis section, it is demonstrated that, based on the widely recognized computational Diffie–Hellman assumption and the random oracle model, the proposed IBAAKA scheme can be proven secure, effectively meeting the necessary security criteria. Zaminkar et al. [17], in their work, employed a strong hybrid technique, incorporating encryption, as an effective solution to address the challenges of Low-Power and Lossy Networks (RPL) protocol, ensuring secure connectivity of devices.

1.4 Organization of the Paper

The structure of remaining of this paper is organized as follows: Sect. 2 introduces the necessary technical preliminaries. Section 3 explains the proposed protocol framework. An experimental evaluation and comprehensive performance analysis are detailed in Sect. 4. Finally, Sect. 5 offers a conclusive discussion along with consolidated observations.

2 Technical Preliminaries

This section presents the significant technical preliminaries in this research domain.

2.1 Lattice

In 1996, Miklós Ajtai [18] pioneered a lattice-based cryptographic framework that guaranteed security through the robustness of a well-analyzed lattice issue. Cynthia Dwork noted the complexity of a specific average-case lattice issue known as the short integer solution (SIS), which is at least as challenging as the most difficult lattice problems. She proposed a cryptographic hash function with security reliant on the computational difficulty of SIS. In 1998, Jeffrey Hofstein, Jill Piefer, and Joseph H. Silverman developed the NTRU, a lattice-based public-key cryptosystem, which, however, did not match the complexity of solving the hardest lattice problems. Oded Regev introduced the first lattice-based public-key encryption system in 2005, which was secure under the worst-case hardness assumptions, using the learning-from-error problem (LWE). Subsequent research has concentrated on enhancing Regev's security proof and the efficiency of the initial scheme. Further work has been

undertaken to create more cryptographic primitives based on LWE and associated problems. For instance, Craig Gentry, in 2009, unveiled the first fully homomorphic encryption scheme rooted in the lattice problem.

A lattice in $\mathbb{R}^m$ is a set of all integer combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m (m \geq n)$

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\} \tag{1}$$

2.1.1 Lattice-Based Cryptographic Primitives

In the field of linear algebra, the shortest vector problem (SVP), occasionally referred to as GapSVP, stands as the most critical computational challenge associated with lattices. It involves estimating the smallest Euclidean length of a nonzero vector within the lattice. Even using polynomial approximation coefficients, this problem is considered difficult to solve efficiently or even a quantum computer. It is known that many (but not all) lattice-based cryptographic structures are secure under this mechanism, even though SVP is difficult.

2.2 Shortest Vector Problem (SVP)

The shortest vector problem (SVP) holds significant importance in lattice-based cryptography, as it revolves around identifying the shortest nonzero vector in a lattice. SVP is widely recognized as one of the prevalent challenges in lattice cryptography. The definition of the problem is as follows: given a basis B Lattice $L(B)$, find a lattice point composed of this basis $\mathbf{B_x}$: $\mathbf{x}$ makes this point the closest distance to the 0 coordinate point.

$$\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^k$$
$$\|\mathbf{Bx}\| \leq \lambda_1$$

Observation found that because λ_1 is already the shortest distance between the point and the point between the middle point and the point, the distance of the $\mathbf{Bx}$ distance is not less than λ_1 at 0 o'clock, at most it is equal to the remarkability.

Table 1 Notations used

$\mathcal{N}$otation	$\mathcal{D}$escription
s	Security constant
v, p	An integer and a prime number
Y	A matrix from $\mathbb{Z}_p^{v \times v}$
e	Master secret-key of KGC
Q	Master public-key of KGC
$G_i(\cdot)$	Cryptographic hash functions, $G_i : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, for $i = 1, 2, 3$
E_j	jth Edge node
sk_j	Private-key of E_j
v_j	Random vector chosen by E_j
C_j, D_j	Computed values by E_j in Session-key Generation phase
T_j	Timestamp used in Session-key Generation phase
U_j	Intermediate key computation by E_j
K_j	Session key computed by E_j

3 Proposed PQS-AKA Protocol

This section introduces the PQS-AKA protocol by detailing the system model and the methodological framework, accompanied by a thorough description of the technical algorithms employed.

3.1 System Model and Methodology Blueprint

Our system model outlines a secure framework for communication between nodes in a network, emphasizing simplicity and security. At the start, a central authority sets up the necessary security parameters, including a master key. Then, each node receives a unique private key through a secure process, ensuring that only legitimate nodes can participate in the network. Finally, nodes use these keys to generate a shared secret key for communication, allowing them to exchange information securely. This process is designed to be both straightforward and secure, ensuring that even with basic knowledge of the system, the integrity and privacy of the communication are maintained. For the sake of clarity and understanding, the nomenclatures exploited are mentioned in Table 1.

Setup Phase During the initial Setup phase, the Key Generation Center (KGC) orchestrates the creation of the primary system parameters. This phase is instigated with a security constant s. The steps performed by the KGC in the Setup phase to yield a set of principal parameters are as follows:

- The KGC elects an integer v adhering to the condition $v \geq 2s \ln p$, where p is established as a prime number greater than $\beta \cdot \sqrt{s(\ln s)}$. Subsequently, a matrix Y from $\mathbb{Z}_p^{v \times v}$ is formulated.
- A random vector e from $\mathbb{Z}_p^v$ is selected. The KGC then computes the primary public key $Q \longleftarrow e^T \cdot Y$, where e^T denotes the transpose of e.
- Three cryptographic hash functions G_i are chosen by the KGC, mapping binary strings to the set $\mathbb{Z}_p^*$ for i in the set $\{1, 2, 3\}$.
- The KGC keeps e as the master secret key and releases the primary parameters $B = \{s, v, p, Y, Q, G_i(\cdot)\}$ publicly.

3.2 Symmetric Key Extraction Phase

During the Private Key Extraction phase, the Key Generation Center (KGC) distributes the private keys to individual edge nodes. The procedure unfolds as follows:

- Edge nodes E_j, for $j = 1, 2$, initiate the process by sending their respective identifiers ID_j to the KGC.
- The KGC, upon receipt of ID_j, authenticates the identity of each edge node E_j using an offline verification method.
- Following a successful verification, the KGC:

1. Generates a random vector v_j from the set $\mathbb{Z}_q^n$.
2. Computes the parameter R_j as the transpose of v_j multiplied by the matrix Y, i.e., $R_j = v_j^T \cdot Y$.
3. Derives a hash value $h_j = G_1(ID_j \parallel R_j)$.
4. Calculates the private key for edge node E_j as $sk_j = (v_j + h_j \cdot e) \mod q$.
5. Securely transmits the private key sk_j and the parameter R_j to the edge node E_j.

3.3 Authenticated Session-Key Generation

Session-Key Generation phase is a crucial procedure executed collaboratively by the edge nodes E_1 and E_2. This phase facilitates the negotiation of a shared session key. The steps are delineated as follows:

Computational Procedure:
$\mathcal{I}$**nput** $\rightarrow Y, B_2, P, P_1, P_2, h_1, h_2, sk_1, sk_2, q, n, \sigma, G_2, G_3$
$\mathcal{O}$**utput** $\rightarrow$ Session keys K_1, K_2

1. Edge node E_1 selects a vector $v_1 \in \mathbb{Z}_q^n$ randomly, ensuring $||v_1|| \leq \sigma$ for a given $\sigma \in \mathbb{Z}^+$.
2. E_1 computes $C_1 = Y \cdot v_1$ and $D_1 = v_1^\top \cdot Y$.
3. E_1 calculates $\epsilon_1 = G_2(B_2 \| C_1 \| T_3)$, where T_3 represents the timestamp.

4. E_1 computes the intermediate key $U_1 = (sk_1 + \epsilon_1 \cdot v_1) \mod q$.
5. E_1 sends the tuple $\{C_1, D_1, T_3, U_1, P_1\}$ to E_2 over a secure channel.
6. Upon receipt, E_2 verifies if $D_1 \cdot Y \stackrel{?}{=} P_1 + h_1 \cdot P + \epsilon_1 \cdot D_1$.
7. If verification is successful, E_2 selects $v_2 \in \mathbb{Z}_q^n$ and calculates $k_2 = v_2^\top \cdot C_1$ and $D_2 = v_2^\top \cdot Y$.
8. E_2 then derives $\epsilon_2 = G_2(B_2 \| C_2 \| T_4)$ and computes $U_2 = (sk_2 + \epsilon_2 \cdot v_2) \mod q$.
9. E_2 computes the session key $K_2 = G_3(k_2 \| D_1 \| D_2 \| T_3 \| T_4 \| P_1 \| P_2)$ and sends $\{D_2, T_4, U_2, P_2\}$ back to E_1.
10. E_1 checks the received tuple and if $U_2 \cdot Y \stackrel{?}{=} P_2 + h_2 \cdot P + \epsilon_2 \cdot D_2$.
11. Following a successful check, E_1 computes $k_1 = D_2 \cdot v_1$ and finalizes the session key $K_1 = G_3(k_1 \| D_1 \| D_2 \| T_3 \| T_4 \| P_1 \| P_2)$.

3.4 Computational Complexity, Security and Robustness Analysis

The computational complexity of the PQS-AKA protocol is a critical factor in assessing its efficiency and practicality. We define the complexity in terms of Big O notation, which provides an asymptotic upper bound on the growth of these operations relative to the input size. The primary operations considered are:

- Hash function computations, denoted as $O(h(n))$, where h represents the complexity of the hash function and n the size of the input.
- Encryption and decryption operations, denoted as $O(e(n))$, where e represents the encryption/decryption function and n the size of the key or plaintext/ciphertext.
- Generation of random vectors within a mathematical space, denoted as $O(r(n))$, where r represents the random vector generation function and n the dimension of the space.

The total computational complexity $T(n)$ of the protocol can be expressed as a combination of these individual complexities:

$$T(n) = O(h(n)) + O(e(n)) + O(r(n)). \tag{2}$$

The resilience of the PQS-AKA protocol against quantum attacks is a crucial aspect of its security analysis. We define the resilience of the protocol against a quantum adversary $\mathcal{Q}$ as follows:

1. Let $T_Q(n)$ denote the time complexity of the most efficient known quantum attack on the protocol, where n represents the size of the cryptographic keys. This time complexity is a function that quantifies the computational effort required by a quantum adversary to compromise the protocol.

2. The protocol is deemed quantum-resilient if $T_Q(n)$ grows super-polynomially with respect to n, which is to say, $T_Q(n)$ exceeds any polynomial time complexity $p(n)$ for sufficiently large n. Formally, the protocol retains its security against quantum-based attacks if, for every polynomial $p(n)$, there exists an n_0 such that for all $n > n_0$, $T_Q(n) > p(n)$. This ensures that the protocol remains secure for all practical sizes of n, making it infeasible for quantum adversaries to break the protocol within a reasonable timeframe.
3. This resilience is quantified by the measure $R_Q = 1 - \epsilon_Q(n)$, where $\epsilon_Q(n)$ represents a negligible function that denotes the probability of a successful quantum attack against the protocol. A function $\epsilon_Q(n)$ is considered negligible if for every positive polynomial $p(n)$, there exists an n_0 such that for all $n > n_0, \epsilon_Q(n) < \frac{1}{p(n)}$. This quantification ensures that the probability of breach by quantum computation remains insignificant, underpinning the protocol's robustness against quantum attacks.

Given the current state of quantum algorithms, the PQS-AKA protocol integrates cryptographic primitives that are conjectured to be resistant to quantum attacks. These include but are not limited to, lattice-based cryptography, hash-based cryptography, and multivariate polynomial equations. The choice of these primitives is guided by their presumed hardness against quantum computational attacks, ensuring the protocol's security even as quantum computing continues to evolve. The protocol's efficacy in safeguarding against quantum threats thus hinges on the continued intractability of solving these cryptographic problems using quantum computers.

4 Performance Analysis

This section provides a detailed performance analysis of the developed system, including benchmarking against a range of computational factors. Considering the factors provided in Table 2 and corresponding graphical representation (depicted as Fig. 1), the proposed PQS-AKA protocol appears to offer a good balance between computational efficiency, storage, and communication overhead, making it a potentially strong candidate among the protocols considered.

In the comparative performance analysis table, several variables are employed to evaluate the protocols. The variable n denotes the dimensionality of the lattice, a criti-

Table 2 Performance analysis comparison: benchmarking

Protocol	Computational complexity	Storage (bits)	Communication (bits)
Gupta et al. [19]	$O(n^3 \cdot \log^2 q)$	$n^2 \cdot \log q$	$3n \cdot \log q$
Rana et al. [20]	$O(n^3 \cdot \log^2 q)$	$(2n^2) \cdot \log q$	$(2n^2 + 3) \cdot \log q$
$\mathcal{P}$roposed PQS-AKA	$O(n^2 \cdot \log^2 q)$	$(n^2 + n) \cdot \log q$	$(4n + 1) \cdot \log q$

cal parameter that impacts both the security level and computational complexity of the protocols. The notation $\log q$ represents the logarithmic scale of the prime number q, which is integral to the modular arithmetic operations central to lattice-based cryptographic mechanisms. Computational complexity is expressed using Big O notation, providing an asymptotic upper limit to the computational resources required as a function of the input size, characterized by the variables n and q. The storage and communication costs are denoted in bits, indicated by the term $\cdot \log q$, which is a direct measure of the bit-length required to represent numbers in the system. These variables collectively serve to quantify and contrast the computational demand, storage needs, and communication overhead across the various protocols.

5 Conclusion

In conclusion, this research work presents the Post-Quantum Secure Lightweight Authenticated Lattice-based Key Agreement (PQS-AKA) protocol as a robust solution for securing communications within the IoT ecosystem. The protocol emerges as a pivotal innovation, particularly relevant in an era where quantum computing poses significant threats to traditional cryptographic methods. The PQS-AKA protocol leverages the complexity of lattice problems to offer quantum resistance, thereby ensuring the longevity and security of IoT communications. Our evaluations under the random oracle model affirm the protocol's resilience against quantum attacks and highlight its superior performance in terms of computational and communicative efficiency.

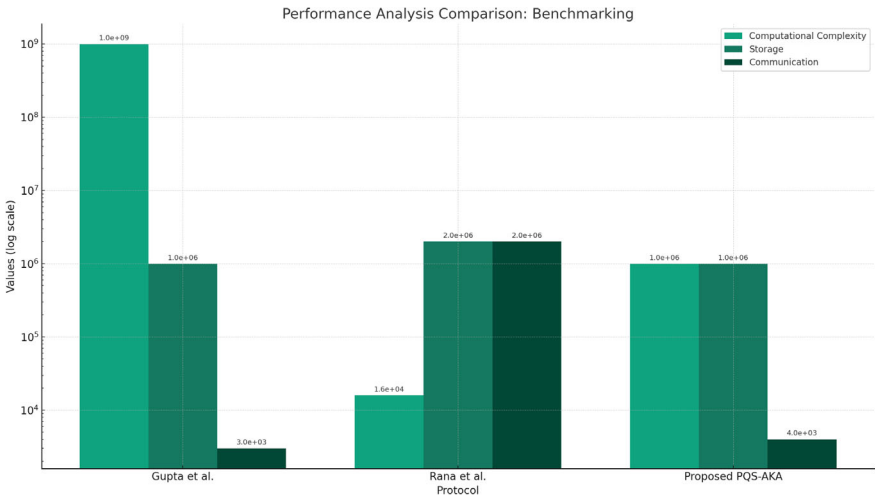


Fig. 1 Benchmarking and comparative analysis

5.1 Future Scope

Looking ahead, the future scope of this work includes several promising directions. While the protocol excels in theoretical robustness and efficiency, its practical implementation in real-world IoT environments remains to be explored. Future research could also focus on integrating the PQS-AKA protocol with emerging IoT technologies and standards, potentially enhancing interoperability and ease of deployment. Additionally, as quantum computing continues to advance, ongoing refinement of the protocol may be necessary to address any newfound vulnerabilities. There is also an opportunity to explore the protocol's application in other areas that require lightweight and secure communications, such as edge computing and smart devices. The PQS-AKA protocol, with its post-quantum security and practical efficiency, is well-positioned to become a standard for the next generation of secure IoT communications.

Funding Information Funding is not applicable for this work.
Competing Interests Authors have no conflict of interest in this research work.
Author Contribution All the authors have contributed equally in this research work.
Data Availability Statement No data is available with this work.

References

1. Guo C, Chang C (2014) An authenticated group key distribution protocol based on the generalized Chinese remainder theorem. Int J Commun Syst 27:126–134
2. Gupta D, Biswas G (2017) An ECC-based authenticated group key exchange protocol in IBE framework. Int J Commun Syst 30:e3363
3. Gupta D, Biswas G, Nandan R (2018) Security weakness of a lattice-based key exchange protocol. 2018 4th International conference on recent advances in information technology (RAIT). pp 1–5
4. Li Q, Hsu C, Raymond Choo K, He D (2019) A provably secure and lightweight identity-based two-party authenticated key agreement protocol for vehicular ad hoc networks. Security and communication networks. 2019:1–13
5. Jamgekar R, Joshi G (2013) File encryption and decryption using secure RSA. International Journal Of Emerging Science And Engineering (IJESE). 1:11–14
6. Bos J, Halderman J, Heninger N, Moore J, Naehrig M, Wustrow E (2014) Elliptic curve cryptography in practice. Financial cryptography and data security: 18th international conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18. pp 157–175
7. Gupta D (2020) An IBE-based authenticated key transfer protocol on elliptic curves. Encyclopedia of criminal activities and the deep web. pp 1112–1122
8. Howgrave-Graham N, Nguyen P, Pointcheval D, Proos J, Silverman J, Singer A, Whyte W (2003) The impact of decryption failures on the security of NTRU encryption. Annual international cryptology conference. pp 226–246
9. Coppersmith D, Shamir A (1997) Lattice attacks on NTRU. International conference on the theory and applications of cryptographic techniques. pp 52-61
10. Gaithuru J, Salleh M (2017) Itru: Ntru-based cryptosystem using ring of integers. In J Innov Comput 7
11. Gaborit P, Ohler J, Solé P (2022) CTRU, a polynomial analogue of NTRU. (Inria,2002)

12. Hashim H, Molnár A, Tengely S (2021) Cryptanalysis of ITRU. Rad Hrvatske Akademije Znanosti I Umjetnosti: Matematičke Znanosti 181-193
13. Lohachab A, Lohachab A, Jangra A (2020) A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum IoT networks. Int Thing 9:100174
14. Islam S, Zeadally S (2020) Provably secure identity-based two-party authenticated key agreement protocol based on CBi-ISIS and Bi-ISIS problems on lattices. J Inf Secur Appl 54:102540
15. Gupta D, Islam S, Obaidat M, Vijayakumar P, Kumar N, Park Y (2020) A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments. IEEE Syst J 15:1732–1741
16. Kumar M, Chand S (2020) A lightweight cloud-assisted identity-based anonymous authentication and key agreement protocol for secure wireless body area network. IEEE Syst J 15:2779–2786
17. Zaminkar M, Sarkohaki F, Fotohi R (2021) A method based on encryption and node rating for securing the RPL protocol communications in the IoT ecosystem. Int J Commun Syst 34:e4693
18. Ajtai M (1996) Generating hard instances of lattice problems. Proceedings of the twenty-eighth annual ACM symposium on theory of computing. pp 99-108
19. Gupta D, Biswas G (2018) A novel and efficient lattice-based authenticated key exchange protocol in C-K model. Int J Commun Syst 31:e3473
20. Rana S, Mishra D (2021) Lattice-based key agreement protocol under ring-LWE problem for IoT-enabled smart devices. Sādhanā. 46:84

"Transforming Emotions: A Comprehensive Review of Text Emotion Detection with Transformer Models"

Brajesh Kumar Khare and Imran Khan

Abstract Emotions are a key aspect of being human. When people write essays or reports, they naturally express their own feelings. Recently, the way we identify emotions in text has been completely transformed by using transformer-based models in natural language processing (NLP). Transformer-based models have revolutionized text emotion detection through Contextual Understanding, Long-range Dependencies, Transfer Learning, Multi-modal Inputs, and Fine-tuning. This review paper thoroughly investigates the present status of investigations using transformers to recognize and understand emotions conveyed in written text. We examine several studies that have utilized transformer architectures, starting from the influential Transformer model to more recent variations like BERT, GPT, XLM, and their derived models. This review paper begins by introducing various emotion models. Following that, it discusses the process of detecting emotions in text. The paper explores different types of embeddings and provides a detailed discussion on transformers and their variations. The literature review section critically assesses existing methodologies, identifying trends, advancements, and gaps in current knowledge. Results from various studies are synthesized and compared. This comprehensive review aims to serve as a valuable asset for scholars, practitioners, and enthusiasts interested in the intersection of transformers and text emotion detection. Moreover, the paper examines the challenges faced in the domain of text emotion detection.

Keywords Transformers · Emotions · Embeddings · BERT · Machine learning

B. K. Khare (✉) · I. Khan
Department of Computer Science and Engineering, Harcourt Butler Technical University, Kanpur, Uttar Pradesh, India
e-mail: brajeshkhare234@gmail.com

I. Khan
e-mail: imran.k@hbtu.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_43

1 Introduction

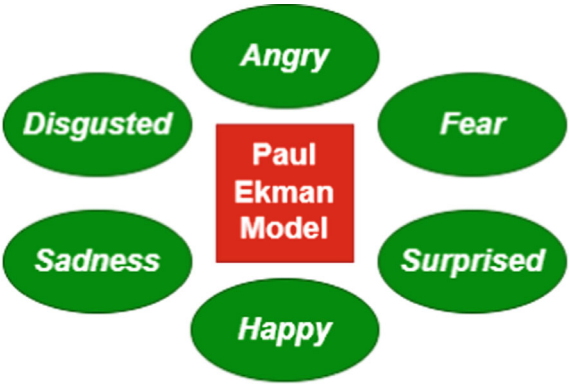
NLP is a specialized field within the realms of Computer Science and AI. It is centered around the computational manipulation of human language, aiming to enable machines to comprehend and generate human languages effectively. The utilization of NLP in favored domains like search engines, sentiment analysis, translation systems, and opinion assessment is addressing societal challenges with an unmatched swiftness [1]. Textual data are inherently sequential, meaning that the arrangement of words and their contextual connections within a sentence holds a crucial role in extracting the full intended meaning. In this context, RNN serves as a specialized sequential model with the ability to effectively manage and process such sequential data [2]. The effectiveness of RNNs is constrained by their slow training process and the challenges posed by capturing dependencies across lengthy sequences. Nevertheless, a derivative of the RNN, namely LSTM, emerges as a crucial remedy to counter some of these drawbacks. While LSTM offers a viable solution to the challenge of long-range sequence dependencies, it does come at a trade-off. It sacrifices parallel computations and, notably, exhibits a slower performance compared to the conventional RNN [3].

Attention networks were introduced as a means to address certain limitations linked to LSTMs and RNNs. The primary objective of an attention network is to concentrate on specific segments within the data that hold significant relevance. The inclusion of attention mechanisms resulted in a significant improvement, as demonstrated in the study conducted by [4]. In this research, the identification of emotion intensities was successfully carried out by employing a stacked ensemble that included a CNN, LSTM, and GRU. The collaborative strength of these components was further enhanced by incorporating attention mechanisms, contributing to a more effective and synergistic emotion detection process.

The layout of the sections in this review paper are detailed and outlined as follows:

- Segment 2 delves into an exploration of various models related to emotions.
- Segment 3 delves into the explanation of the text emotion detection process.
- Segment 4 offers a brief overview of the Transformers.
- In Segment 5 different transformer-based models are thoroughly analyzed.
- In Segment 6, an analysis is conducted on previous studies within the domain of text emotion detection that made use of pre-trained models based on transformers.
- Finally, Segment 7 serves as the conclusion and summarizing the encountered challenges

Fig. 1 Paul Ekman model



2 Emotion Models

Emotion models are computational frameworks or algorithms designed to identify, analyze, and understand human emotions expressed in various forms of data, such as text, speech, images, and videos. Psychological or emotional models can be broadly divided into two groups: Categorical and Dimensional.

2.1 Categorical Emotion Model

The categorical model for emotions defines specific emotional states such as anger, happiness, sadness, and fear. Following this model, emotions are grouped into four, six, or eight distinct categories. This model is also known as a discrete emotion model. Six basic categories of emotions are distinguished by the *Paul Ekman model*: happiness, sorrow, disgust, surprise, wrath, and fear. However, more complex emotions such as pride, guilt, lust, greed, shame, and others may arise from the combination or synthesis of these basic emotions. Figure 1 describes Paul Ekman model.

2.2 Dimensional Emotion Model

This approach employs three factors—"Valence", "Arousal", and "Power"—to characterize emotions. Arousal quantifies the level of excitement associated with a sentiment, while Valence pertains to its positive or negative nature. For example, joy is more exciting than happiness and contentment. Power or dominance signifies restriction over emotion [5]. Figure 2 describes dimensional emotion model.

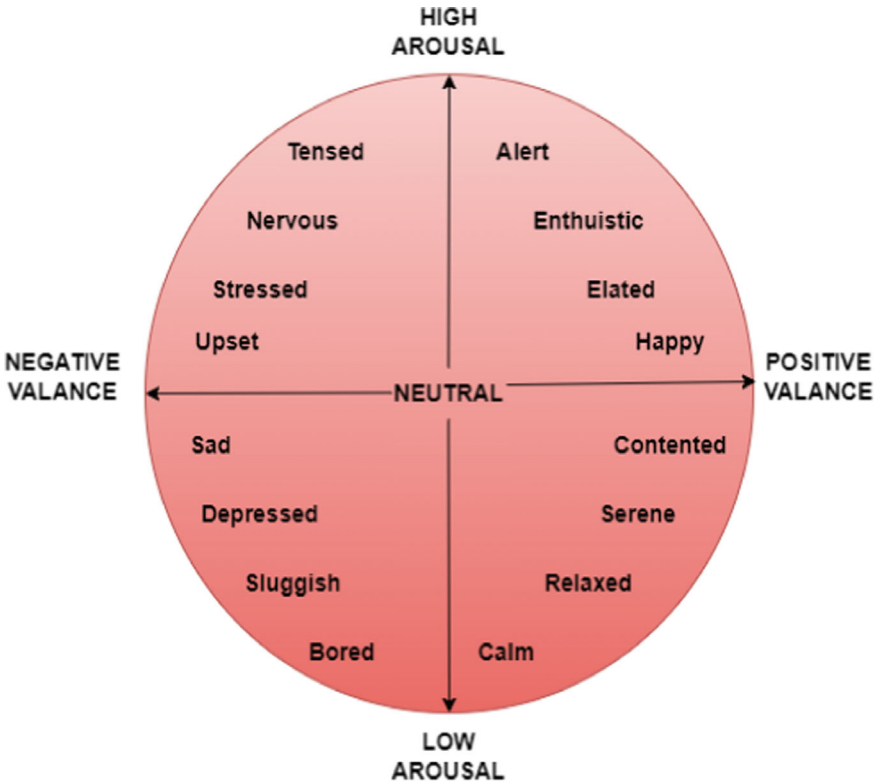


Fig. 2 Dimensional emotion model [6]

3 Text Emotion Detection Process

Text emotion detection involves the automatic identification and categorization of emotions expressed within a given piece of text. It involves applying NLP and ML techniques to analyze text and determine the underlying emotions, such as happiness, sadness, anger, fear, etc. This is a summary of the steps included in the process of identifying emotions in text, as depicted in Fig. 3.

3.1 Data Collection

Collecting data for text emotion detection involves gathering a diverse and representative dataset of text samples along with their corresponding emotion labels. There are several publicly available datasets that we can use for text emotion detection tasks. These datasets come from various sources like social media, news articles, and other written communication forms. The annotations and labels assigned to

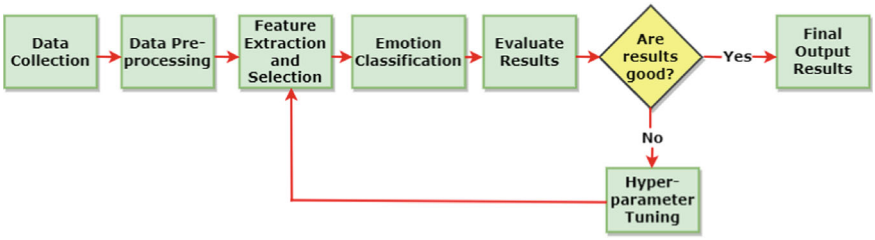


Fig. 3 Text emotion detection process

each text sample, carried out by human annotators, are pivotal in training the model to recognize different emotions accurately. Some popular datasets for text emotion detection includes EmoReact, ISEAR, SemEval, Aman's dataset, EmoInt, MELD, Smile dataset, etc.

3.2 Data Preprocessing

It is an essential stage in preparing text data for tasks like text emotion detection. Proper preprocessing ensures that the data is clean, standardized, and suitable for feeding into machine learning models. It includes Text Cleaning, Tokenization, stop word Removal, Lower casing, Stemming and Lemmatization, Removing Non-Alphanumeric Characters, etc.

After preprocessing, we'll have a cleaned and standardized dataset that can be used to train and evaluate our text emotion detection model.

3.3 Feature Extraction

It entails the conversion of unprocessed textual data into a structure suitable for input into machine learning algorithms. In the context of text emotion detection, features are numerical representations of the text that capture its underlying characteristics [7]. These features empower machine-learning models to acquire patterns and establish relationships within the text data. Some common techniques for feature extraction in text emotion detection are TF-IDF, one hot encoding, n-grams, Bag-of-Words, etc.

3.4 Embeddings

In NLP, an "embedding" refers to a numerical representation of words, phrases, sentences, or other textual elements in a continuous vector space. Embeddings are

used to convert text data, which is inherently discrete and categorical, into a format that can be processed by ML algorithms that work with numerical data [8]. The main idea behind embeddings is to capture semantic relationships and contextual information between words or tokens. Words that are semantically similar or related in meaning should be represented as vectors that are closer together in the embedding space.

There are several types of embeddings used in NLP, each with its own characteristics and applications. Different categories of embeddings are explained below.

3.4.1 Word Embeddings

In the context of a paragraph or larger text, the term "word embeddings" refers to the representation of words as continuous vector spaces, capturing the semantic relationships between words based on their contextual usage within the given text [9]

- ***Word2Vec***: This method involves representing words as compact vectors by predicting context words from target words or vice versa. It adeptly captures semantic connections among words.
- ***GloVe***: GloVe generates embeddings by utilizing word co-occurrence statistics, encompassing both global and local connections among words.
- ***FastText***: FastText extends word embeddings to subword-level by considering character n-grams. It is useful for handling morphologically rich languages and rare words.

3.4.2 Contextualized Word Embeddings

Contextualized word embeddings represent words in a way that captures their context-dependent meanings within a given text. Unlike traditional static word embeddings, which assign a fixed vector to each word irrespective of context, contextualized embeddings take into consideration the surrounding words and their order in a sentence [10]

- ***BERT***: BERT generates contextualized word embeddings by considering surrounding words in both directions. It's widely used for various NLP tasks, capturing intricate contextual meanings.
- ***ELMo (Embeddings from Language Models)***: ELMo produces word embeddings by training a bi-directional LSTM model on a language modeling task. These embeddings capture word meanings in different contexts.

3.4.3 Sentence/Paragraph Embeddings

Sentence or paragraph embeddings refer to vector representations that capture the overall meaning or semantics of a sentence or paragraph. Unlike word embeddings that represent individual words as vectors, sentence embeddings aim to encode the entire context and semantic content of a sequence of words [11]

- ***Doc2Vec***: Similar to Word2Vec, Doc2Vec generates embeddings for entire documents or paragraphs. It captures the semantic meaning of larger text segments.
- ***Universal Sentence Encoder***: This model generates sentence embeddings that can be used for semantic similarity and classification tasks. It's trained on a large amount of data and supports multiple languages.
- ***InferSent***: Designed specifically for sentence embeddings, InferSent uses a bi-directional LSTM to capture sentence semantics and is used for various NLP tasks.

3.4.4 Cross-Lingual Embeddings

Cross-lingual embeddings are word or sentence representations that capture the semantic connections between languages. The objective is to establish a common embedding space where comparable meanings are in close proximity, irrespective of the language in which the words or sentences are expressed. Such embeddings prove especially beneficial for tasks involving multiple languages, including machine translation, cross-lingual information retrieval, and multilingual sentiment analysis [12]

- ***Multilingual BERT (mBERT)***: A variant of BERT trained on multiple languages, providing embeddings that work well across different languages.
- ***Word Translation Embeddings***: These embeddings strive to represent words from diverse languages in a unified space, facilitating cross-lingual applications.

3.5 Model Development

It may include Model Selection, Data Splitting, Model Building and Training, and Hyperparameter Tuning.

Model Selection

- Select a model architecture suitable for text classification. Common choices include RNNs, CNNs, LSTM, GRU, or Transformer-based models.
- Configure the architecture to handle the number of emotions and text sequence lengths.

Data Splitting

We can split the entire dataset into three segments: training, testing, and validation data. The distribution would allocate 80% for training, 10% for validation, and 10% for testing purposes.

Model Building and Training

- Initialize the chosen model with appropriate hyperparameters.
- Train the model using the training set and track its performance during training using the validation set.
- Select an appropriate loss function like categorical cross-entropy for tasks involving multi-class classification.
- Apply optimization techniques like Adam, SGD, or more advanced optimizers.

Hyperparameter Tuning

- Adjust hyperparameters such the number of hidden units in layers, batch size, dropout rate, and learning rate.
- Use techniques like grid search or random search for optimization.

3.6 Evaluation of the Model

- A model's performance may be obtained from evaluation measures in a variety of ways, including how well it predicts outcomes, manages unbalanced classes, generalizes to new data, and more.
- Frequently used evaluation metrics for classification includes precision, recall, AUC-ROC, F1-score, accuracy, Confusion Matrix, and various additional metrics.
- We can employ metrics like the Silhouette Score and the Calinski–Harabasz Index for the purpose of clustering analysis.

4 Transformers

Transformers constitute a kind of deep learning framework that has drastically transformed a variety of tasks within NLP. The primary breakthrough introduced by transformers is the self-attention mechanism, enabling the model to assess the significance of distinct words in a sentence while handling each individual word [13]. This mechanism empowers transformers to grasp extensive dependencies and context in a text, rendering them exceptionally proficient for tasks that require comprehension of sequential data, such as language-related tasks. Figure 4 describes the architecture of transformer.

A transformer in deep learning transforms a string of words or tokens into numerical representations. It uses a softmax function to modify the output probabilities

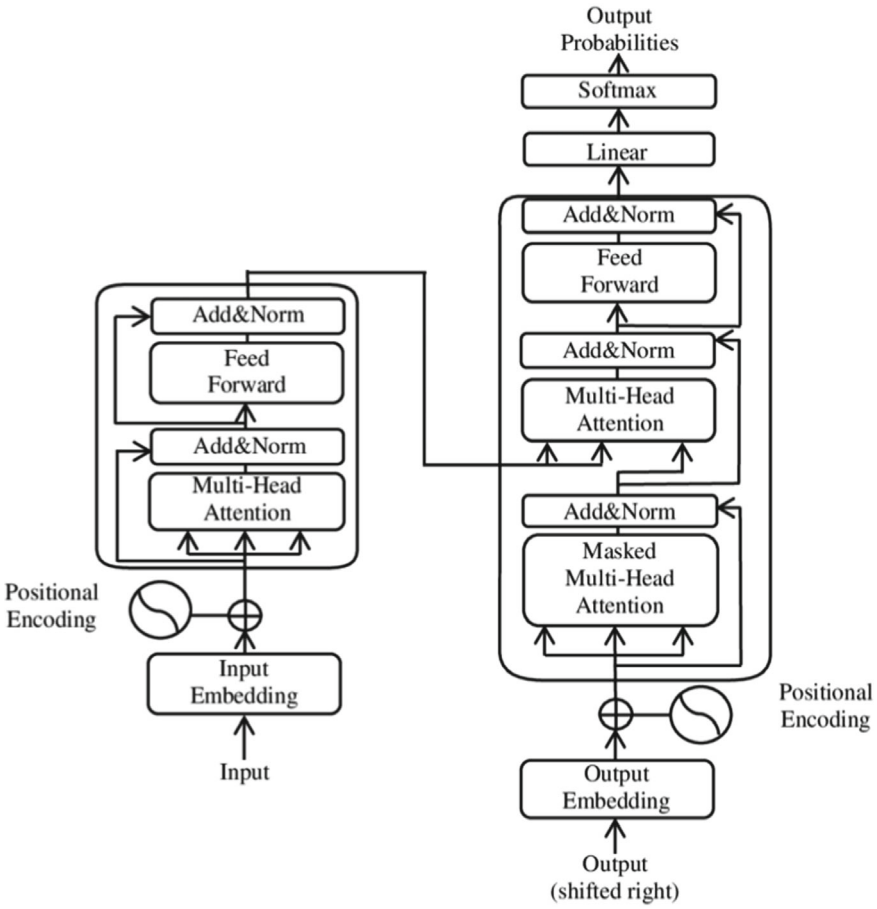


Fig. 4 Transformer architecture [14]

and is composed of encoder and decoder components. The encoder blocks are feed-forward networks with multi-head attention that receive embeddings. By calculating attention vectors for every input, the multi-head attention layers are able to identify the connections between words in a phrase. Next, another multi-head attention layer receives the input of the attention vectors produced by the encoder block and by masked multi-head attention layers. These vectors then pass via a linear layer, a feed-forward network, and a softmax activation function [15]. The features and their descriptions are provided in Table 1.

Table 1 Features of transformer

Feature	Description
Self-attention mechanism	Allows the model to weigh different input positions differently, capturing contextual information
Multi-head attention	Uses multiple attention heads in parallel, enabling the model to learn diverse relationships
Scaled dot-product attention	Attention scores are scaled, preventing vanishing/exploding gradients during backpropagation
Positional encoding	Embeds positional information to convey the token's position in the input sequence
Encoder–decoder structure	Employs separate encoder and decoder stacks, each composed of multiple layers
Feed-forward neural network	Position-wise feed-forward network processes information after attention layers
Layer normalization	Normalizes the output of sub-layers, contributing to training stability and faster convergence
Residual connections	Includes residual connections around each sub-layer, aiding gradient flow during training
Encoder–decoder attention	Incorporates an attention mechanism in the decoder, attending to the encoder's output
Masked self-attention (decoding)	Prevents attending to future tokens during decoding, maintaining causality in sequence generation
Position-wise feed-forward network	Utilizes fully connected layers and ReLU activation for token-level information processing
Final linear and softmax layer	Produces the final probability distribution over the target vocabulary in the decoder

5 Transformer-Based Models

This section provides further details about specific models that have arisen from the transformer architecture, emphasizing their strengths.

5.1 Transformer-XL

Transformer-XL expands upon the original Transformer architecture to tackle the limitation of context in the vanilla Transformer model. The context limitation refers to the fact that the standard Transformer model has a fixed-length context window, meaning it can only consider a limited number of tokens in the input sequence. This limitation makes it challenging to capture long-range dependencies effectively [16]. To overcome this limitation, Transformer-XL [17], introduces two main innovations: a recurrence mechanism and a segment-level recurrence. Transformer-XL excels at capturing extended dependencies compared to traditional Recurrent Neural Networks

(RNNs). Moreover, the model retains the ability to capture short-term dependencies effectively.

5.2 Generative Pre-training (GPT)

GPT, form a series of neural network models built on the transformer architecture, representing a significant breakthrough in artificial intelligence (AI) that drives generative AI applications like ChatGPT. GPT models empower applications to generate text and content resembling human-like quality, spanning various formats such as images and music. Additionally, they enable responses to questions in a conversational style [18]. Various sectors are utilizing GPT models and generative AI for a range of purposes, including developing Q&A chatbots, summarizing text, generating content, and improving search functionalities. Predicting the subsequent token in the sequence is the only job the GPT has. It does this with the design in Fig. 5.

The input passes through a feed-forward layer, the multi-head attention layer in 12 stacked transformer decoder blocks, and ends with the softmax producing a probability distribution. This may be stated mathematically as (Eqs. 1–3)

$$y_0 = UQ_e + Q_p \tag{1}$$

$$y_l = \boldsymbol{Transforme_Block}(y_{l-1}) \forall i \in [1, n] \tag{2}$$

$$S_{(u)} = \boldsymbol{Softmax}\left(y_n Q_e^T\right) \tag{3}$$

In this case, the number of layers is n, the context vector of tokens is $U = (u - k, \ldots, u - 1)$, the token embedding matrix is Q_e and the position embedding matrix is Q_p.

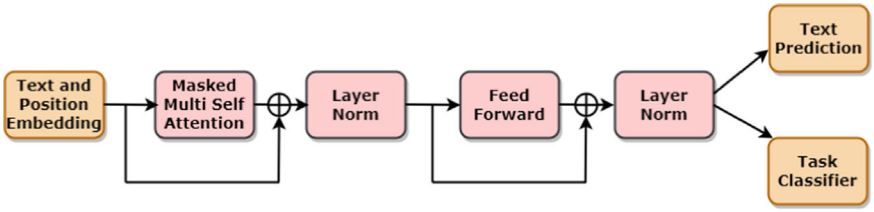


Fig. 5 GPT2 architecture

5.3 Bert

BERT, created by Google AI in 2018, is an advanced NLP model that brought significant changes to the field. It introduced bidirectional context understanding, allowing it to better understand word meanings in sentences. Built on the transformer architecture with self-attention mechanisms, BERT reads text bidirectionally during training, capturing a more complete context. In pre-training, it learns contextualized word embeddings by predicting masked words and establishing sentence relationships. This helps it comprehend context and meaning. In fine-tuning, BERT's pre-trained representations are adapted with task-specific layers for various NLP applications like question answering, sentiment analysis, and more. BERT's strengths include bidirectional context, contextualized embeddings, transfer learning, and versatility across NLP tasks [19]. Figure 6 illustrates the overall architecture of the BERT model.

5.4 Cross-Lingual Language Models

Cross-lingual language models [20] are NLP models that have the ability to understand and process text in multiple languages. These models are designed to handle multilingual data, making them useful for tasks that involve different languages. These models are trained on a diverse range of languages and can effectively bridge language barriers by enabling translation, text generation, and other language-related tasks across different languages.

5.5 XLNet

A powerful NLP model called XLNet [21] combines the ideas of the Transformer-XL model with the Permutation Language Model. It improves bidirectional context understanding and robustness, making it a powerful tool for various NLP tasks such as Summarization, Machine Translation, and Text Generation. It stands for "eXtreme Language understanding" and is designed to overcome some limitations of previous language models like BERT. As a derivative of BERT, the key distinction between BERT and XLNet lies in their training objectives. XLNet utilizes the permutation objective, in contrast to BERT, which employs data masking and aims to predict the masked data using bidirectional context. Key assets of XLNet involve its ability to capture bidirectional context, robustness due to permutation modeling, and achieving cutting-edge performance on diverse tasks within the field of NLP.

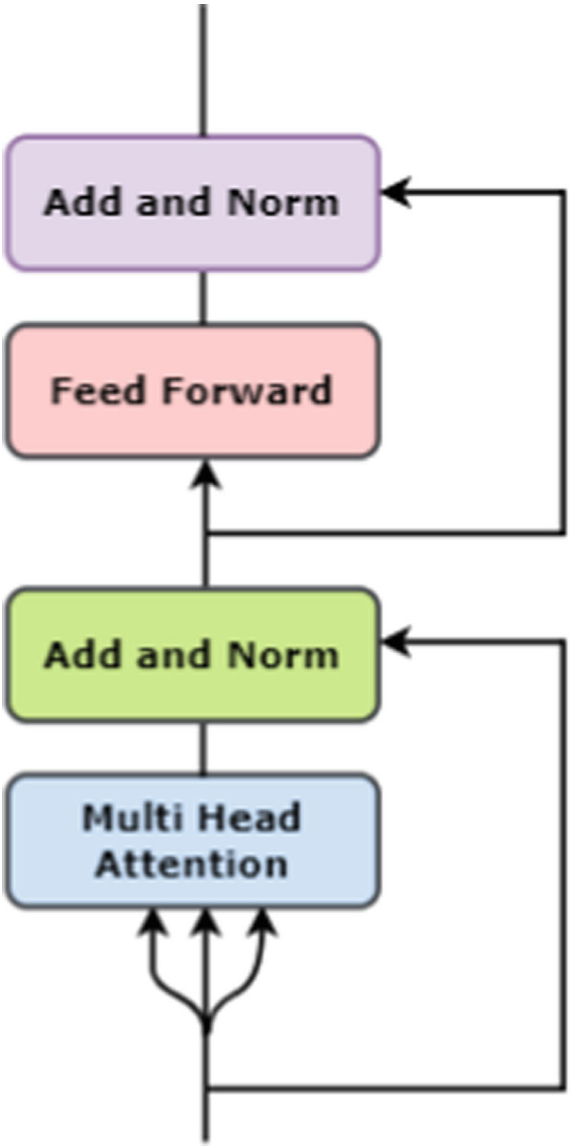


Fig. 6 Architecture of the BERT model [15]

5.6 RoBERTa

RoBERTa [22] is a modified version of the Transformer-based language model architecture, renowned for its impressive performance across different natural language processing tasks. While it builds upon the BERT architecture, RoBERTa incorporates various training enhancements that contribute to its superior results. RoBERTa uses

larger batch sizes, more training data, and longer training times to improve its understanding of context and semantics in text. In RoBERTa, the masking is performed dynamically for each batch, rather than using a fixed set of masked tokens throughout the training. After pretraining, RoBERTa can be fine-tuned on different tasks, such as text summarization, machine translation, question answering, and more.

5.7 DistilBERT

DistilBERT [23] is a compressed and distilled version of the BERT model, created to reduce the model's size and computational requirements while still maintaining much of its performance. It possesses fewer parameters in contrast to BERT, making it a more compact and memory-efficient model. Due to its smaller size, DistilBERT offers faster inference times compared to BERT, which can be useful for real-time applications. The DistilBERT model provides pre-trained models and tools for easy integration into different NLP tasks. Table 2 provides an overview of specific Transformer-Based Models designed for the task of detecting emotions in textual content.

6 Literature Survey

Yen-Hao Huang et al. [24] explore how BERT, a pre-trained language model, performs in emotion recognition. They adapt BERT to tasks requiring continuous dialogue emotion prediction, leveraging its two-sentence structure. Through experiments, they demonstrate that mapping continuous dialogue into a causal utterance pair, comprising the utterance and the reply utterance, enhances emotion capture. The proposed method achieves high micro F1 scores of 0.815 and 0.885 in the testing datasets of Friends and EmotionPush, respectively. Malte and Ratadiya [25] suggested a new method for detecting cyber abuse using deep learning. They used a bidirectional transformer-based BERT architecture to analyze text in English, Hindi, and a mix of Hindi and English (Hinglish). Their approach outperformed other methods on identifying aggression in code-mixed Hindi text, meeting the TRAC-1 standard. Interestingly, their model also performed well on the English task, without needing multiple models or ensemble-based methods. This suggests that their proposed approach is a promising and efficient alternative to current methods for cyber abuse detection. Sungjoon Park et al. [26] presented a model to predict detailed emotions using three dimensions: valence, arousal, and dominance (VAD). They trained the model by minimizing the Earth Mover's Distance (EMD) loss, aligning predicted VAD scores with emotion categories sorted along the VAD dimensions. This model can both classify emotion categories and estimate VAD scores for a given sentence. The authors used a pre-trained RoBERTa-Large, adjusted it on three datasets with emotion labels, and tested its performance on the EmoBank dataset

Table 2 Transformer-based text emotion detection models

Transformer model	Architecture	Pre-training task	Number of layers	Attention heads	Hidden size	Parameters
BERT	Bidirectional transformer	Masked language modeling	12 or 24	12	768	110 million (BERT-base)
GPT-3	Autoregressive transformer	Language modeling	175 billion	96	1256	175 billion
T5	Encoder–decoder transformer	Text-to-text transfer learning	12	12	1024	220 million (T5-small)
RoBERTa	Bidirectional transformer	Masked language modeling	12	12	768	125 million (RoBERTa-base)
DistilBERT	Bidirectional transformer	Masked language modeling	6	6	768	66 million (DistilBERT-base)
ALBERT	Bidirectional transformer	Masked language modeling	Variable	Variable	Variable	Variable (ALBERT-base)
XLNet	Autoregressive transformer	Permutation language modeling	12 or 24	16	1024	177 million (XLNet-base)
ELECTRA	Bidirectional transformer	Replaced token detection	12	12	768	110 million (ELECTRA-base)
XLM	Bidirectional transformer	Masked language modeling	Variable	Variable	768	270 million (XLM-R Base)

with VAD scores. Jwa et al. [27] are working on ways to automatically find fake news using data-driven methods. They use a model called BERT to figure out if news is fake by looking at how the headline and the body of the news relate. To make the model better, they get more news data and use it to train the model before. They find that BERT's deep understanding works well for this job and boosts the F-score by 0.14 compared to older top-performing models. A single neural network model that integrates BERT, BiLSTM, and Capsule modules was created by George Vlad et al. [28] to handle the task of identifying propaganda at the sentence level. Compared to starting from scratch, they achieved higher results using a pre-training strategy on a similar problem (emotion categorization). Their solution achieved the 12th rank out of 26 participating teams in the NLP4IF-2019 Task SLC, with an F1-score of 0.5868 on the official test data. In Social NLP 2019, Luo and Wang [29] describe how they approached the EmotionX-2019 shared task. Their main focus is on identifying emotions from spoken expressions in two datasets: EmotionPush, a Facebook conversation log, and Friends, a TV program. They use a two-step deep

learning technique where each phrase is translated into a series of vectors that indicate its meaning. Then, they anticipate one of four potential moods using a basic softmax classifier. They employ a pre-trained model (BERT) to transfer information from a huge corpus and refine their model on the particular dataset because they lack labeled data. On the Friends and EmotionPush test sets, the model obtains micro-F1 scores of 79.1% and 86.2%, respectively. Kazameini et al. [30] presented a novel deep learning approach for automatically detecting personality using text. They utilized the BERT language model to capture contextualized word representations from textual information. Their method included inputting these contextualized representations, along with psycholinguistic features, into a Bagged-SVM classifier to predict personality traits. This innovative model outperformed the previous best method by 1.04%, reaching an accuracy of 59.03%. Mehta et al. [31] presented a novel deep learning model merging traditional psycholinguistic features with language model embeddings for predicting personality traits. They applied this methodology to both the Essays dataset, focusing on Big-Five personality traits, and the Kaggle dataset for MBTI. The model demonstrated cutting-edge performance. Additionally, they employed interpretable machine-learning methods to visually illustrate and quantify the impact of various language features in the personality prediction models.

Table 3 provides a summary of the latest developments that have been addressed.

7 Challenges and Future Directions

Utilizing Transformers for detecting emotions in text presents a hopeful avenue, yet it presents several hurdles. The key obstacles in utilizing transformers for detecting emotions in text include the following:

- Limited data for certain emotions, data imbalance.
- Lack of labeled data.
- Detecting emotions in text becomes difficult when there is an absence of emotion-related keywords or phrases.
- Inadequate information, typing errors, informal language, brief texts, emojis, sarcasm, irony, harmony, and more constitute various text-related challenges.
- The presence of multiple emotions and perspectives within a single sentence presents a challenge in detecting multiple emotions.
- Language dependency poses a substantial challenge as well.
- Models trained on certain types of text like social media posts might not generalize well to other domains like formal documents.
- Distinguishing between closely related emotions like Sadness and Grief, Fear and Anxiety, Anger and Frustration can be difficult.
- Highly complex transformer models might require substantial fine-tuning and optimization for emotion detection, which can be resource-intensive.

Table 3 Different approaches proposed by researchers

References	Dataset(s) Used	Contribution	Outcomes
Yen-Hao Huang et al. [24]	Friends and EmotionPush dataset	Introduce FriendsBERT and ChatBERT as solutions for the task of recognizing emotions in multi-utterance conversations	Achieved a micro F1 score of 0.815 in the Friends testing dataset and 0.885 in the EmotionPush testing dataset
Malte and Ratadiya [25]	TRAC-1	Identified instances of cyber abuse in texts written in English and Hindi through the utilization of the BERT model	Attained F1 scores of 0.6244 for English texts and 0.6596 for Hindi texts
Sungjoon Park et al. [26]	ISEAR SemEval Emobank	Conducted emotion classification and forecasted VAD scores for dimensional emotions utilizing the RoBERTa model	Attained micro F1 scores of 0.668 (ISEAR) and 0.725 (SemEval), along with VAD scores of 0.659, 0.327, and 0.287
Heejung Jwa et al. [27]	News Headlines	Employed BERT for the identification of fake news	Attained a macro-averaged F1 score of 0.746
George Vlad et al. [28]	SemEval 2019 task 3 and SLC's Daily dialogue task	Merged BERT, bi-LSTM, and Capsule models into an ensemble for efficient detection of emotions and propaganda	Achieved a micro F1 score of 0.5868
Luo and Wang [29]	Friends EmotionPush	Employed fine-tuning on BERT for the objective of emotion detection	Achieved a micro F1 score of 79.1% for the "Friends" dataset and 86.2% for the "EmotionPush" dataset
Kazameini et al. [30]	Essays	Utilized BERT word embeddings and employed a bagged SVM approach to predict authors' personality traits	Attained 59.03% accuracy on average
Mehta et al. [31]	Datasets for essays and Kaggle competitions	applied a mix of psycholinguistic characteristics and deep learning to identify personality traits	Attained an average score of 60.6 for the Essays dataset and 77.1 for the Kaggle dataset

The future directions of text emotion detection using Transformers are likely to involve several key areas of exploration and improvement. Some potential future directions include:

- **Fine-Tuning and Transfer Learning Techniques**: Enhancing the adaptability of pre-trained Transformer models for specific emotion detection tasks through advanced fine-tuning strategies and transfer learning approaches.
- **Context-Aware Emotion Understanding**: Improving models to better understand nuanced contextual information, including sarcasm, humor, and cultural nuances, to enhance the accuracy of emotion detection.
- **Domain-Specific Customization**: Tailoring Transformer models for specific domains, such as healthcare, education, or social media, to improve their performance in understanding domain-specific emotional expressions.
- **Creation of Benchmark Datasets**: Establishing standardized benchmark datasets and evaluation metrics specific to text emotion detection tasks to facilitate fair comparisons and advancements in model performance.
- **Continual Learning**: Enabling Transformer models to continually learn and adapt to evolving language and emotional expressions over time, ensuring their relevance in dynamic and changing environments.

8 Conclusion

This paper primarily explores the process of detecting emotions in text, delves into the idea of embeddings, and examines model development process. It outlines the details of transformers and their diverse variations along with their benefits. Moreover, the paper offers valuable perspectives on the prevailing strategies for text-based emotion detection using transformers. We also explore distinct challenges related to employing transformers for the detection of emotions in textual content. In conclusion, the utilization of transformer-based models for emotion detection represents a significant advancement in the field. These models offer remarkable capabilities in capturing contextual nuances and intricate patterns within text, enabling more accurate and nuanced emotion classification. By leveraging attention mechanisms and large-scale pretraining, transformers excel at comprehending the intricate interplay of emotions, resulting in improved performance compared to traditional methods. This summary discusses the recent advancements, challenges, and future directions in understanding how written text conveys emotions, especially with the use of Transformer models. It serves as a useful guide for researchers and practitioners who are exploring the intersection of language comprehension and emotions in technology.

References

1. Yue L et al (2019) A survey of sentiment analysis in social media. Knowl Inf Syst 60:617–663
2. Du K-L, Swamy MNS (2013) Neural networks and statistical learning. Springer Science and Business Media
3. Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association
4. Akhtar MS, Ekbal A, Cambria E (2020) How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes. IEEE Comput Intell Magaz 15(1):64–75
5. Bruna O, Avetisyan H, Holub J (2016) Emotion models for textual emotion classification. J Phys: Conf Ser 772(1), IOP Publishing.
6. Akbulut O (2020) Feature normalization effect in emotion classification based on EEG signals. Sakarya University J Sci 24(1):60–66
7. Lewis DD (1992) Feature selection and feature extraction for text categorization. Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23–26, 1992
8. Pilehvar MT, Camacho-Collados J (2020) Embeddings in natural language processing: Theory and advances in vector representations of meaning. Morgan & Claypool Publishers
9. Almeida F, Xexéo G (2019) Word Embeddings: a survey, Accessed: Aug. 09, 2023. [Online]. Available: http://arxiv.org/abs/1901.09069
10. Hofmann V, Pierrehumbert JB, Schütze H (2020) Dynamic contextualized word embeddings. ArXiv preprint arXiv:2010.12684
11. Lin Z et al (2017) A structured self-attentive sentence embedding. ArXiv preprint arXiv:1703.03130
12. Ruder S, Vulić I, Søgaard A (2019) A survey of cross-lingual word embedding models. J Artif Intell Res 65:569–631
13. Vaswani A et al (2017) Attention is all you need. Adv Neural Inf Proc Syst 30
14. Vaswani A et al (2021) Attention is all you need. (Nips), 2017. http://arxiv.org/abs/1706.03762
15. Acheampong FA, Nunoo-Mensah H, Chen W (2021) Transformer models for text-based emotion detection: a review of BERT-based approaches. Artif Intell Rev 1–41
16. Al-Rfou R et al (2019) Character-level language modeling with deeper self-attention. Proc AAAI Conf Artif Intell 33(01)
17. Dai Z et al (2019) Transformer-xl: attentive language models beyond a fixed-length context. ArXiv preprint arXiv:1901.02860
18. Rathje S et al (2023) GPT is an effective tool for multilingual psychological text analysis
19. Devlin J et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. ArXiv preprint arXiv:1810.04805
20. Conneau A, Lample G (2019) Cross-lingual language model pretraining. Adv Neural Inf Proc Syst 32
21. Yang Z et al (2019) Xlnet: generalized autoregressive pretraining for language understanding. Advances in neural information processing systems 32
22. Liu Y et al (2019) Roberta: a robustly optimized bert pretraining approach. ArXiv preprint arXiv:1907.11692
23. Sanh V et al (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv preprint arXiv:1910.01108
24. Huang Y-H et al (2019) EmotionX-IDEA: emotion BERT—an Affectional model for conversation. ArXiv preprint arXiv:1908.06264
25. Malte A, Ratadiya P (2019) Multilingual cyber abuse detection using advanced transformer architecture. TENCON 2019–2019 IEEE Region 10 Conference (TENCON). IEEE
26. Park S et al (2019) Dimensional emotion detection from categorical emotion. ArXiv preprint arXiv:1911.02499
27. Jwa H et al (2019) Exbake: automatic fake news detection model based on bidirectional encoder representations from transformers (bert). Appl Sci 9(19):4062

28. Vlad, G-A et al (2019) Sentence-level propaganda detection in news articles with transfer learning and BERT-BiLSTM-capsule model. Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda
29. Luo L, Wang Y (2019) Emotionx-hsu: adopting pre-trained bert for emotion classification. ArXiv preprint arXiv:1907.09669
30. Kazameini A et al (2020) Personality trait detection using bagged svm over bert word embedding ensembles. ArXiv preprint arXiv:2010.01309
31. Mehta Y et al (2020) Bottom-up and top-down: Predicting personality with psycholinguistic and language model features. 2020 IEEE International Conference on Data Mining (ICDM). IEEE.

Pose Estimation: Human Keypoint Prediction

Madhumati C Mailarada, R. U. Rachanna, Sonal Sheth, Prabha C Nissimagoudar, M. Kaushik, and Nalini Iyer

Abstract Real-time Pose Estimation is a very crucial element in comprehending human poses. It has a significant role in many areas of our day-to-day life. In this work, we describe an instantaneous technique to determine the individual's stance in a picture. Here, we use confidence maps, which tell us about how confident the keypoint is located at that position and also our approach learns to associate the body parts with a non-parametric representation that preserves both limb alignment and position information, the term for this is Part Affinity Fields (PAFs). The bottom-up approach performs well in real time with great accuracy. Previous studies found that across training, PAFs and body part localization estimation both enhanced concurrently. We show that there is a significant improvement in runtime performance and accuracy when PAF-only refinement is used instead of integrated optimization of the body component location and PAF.

Keywords Human pose estimation · Confidence maps · Part affinity fields

M. C. Mailarada (✉) · R. U. Rachanna · S. Sheth · P. C. Nissimagoudar · M. Kaushik · N. Iyer
KLE Technological University, Hubballi, India
e-mail: madhumaticmailarada@gmail.com

R. U. Rachanna
e-mail: rachannaom595@gmail.com

S. Sheth
e-mail: sonalsheth107@gmail.com

P. C. Nissimagoudar
e-mail: pcngoudar@gmail.com

M. Kaushik
e-mail: kaushik@kletech.in.ac

N. Iyer
e-mail: nalinic@kletech.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_44

1 Introduction

Human keypoint prediction and pose estimation have many applications in our daily lives, contributing to various fields. It is primarily concerned with locating individual body parts, also called keypoints. We deal with such kinds of machines in day-to-day life where pose estimation plays a key role, for example, it helps us to correct our exercise, yoga poses, and gym applications. Pose estimation plays a critical role in the upcoming autonomous vehicles through understanding the pose of pedestrians and road agents, which directs the self-driving car and it also has a significant role in gesture-based control systems which use pose information to interpret hand movements and gestures.

Touchless interfaces in public spaces, such as shopping centers, utilize this pose estimation to enable touch-free interactions, which played a crucial role in the COVID-19 pandemic. We can also see the importance of pose estimation in gaming where it allows players to use their body movements to control in-game characters or elements, which gives the players a thrilling gaming experience. So human keypoint prediction and pose estimation play a crucial role in many applications, so it is very important to work on them.

A single-person pose estimation method usually involves a person detector for estimating pose. Most of the existing solutions have used a top-down approach, which has the following disadvantages:

- This approach faces the challenge of early commitment, where the success of the entire process depends on the first step's accuracy. If this fails, it leads to significant errors in the subsequent pose estimation.
- It generally fails in crowded situations.

So to overcome these challenges, in this work, the authors have used a bottom-up approach that avoids early commitment issues and also used confidence maps which tell us about how confident the keypoint is located at that position, and the part affinity fields (PAFs) which tell us about both location and orientation information across the limb.

The following are the objectives of this paper:

- Access and visualize keypoint annotations in the COCO dataset.
- Extract the features using the VGG19 model for the CNN model.
- Feed the obtained features to multi-stage CNNs for joint prediction of

(a) Confidence maps to understand how sure the model is about the location of body parts.
(b) PAFs to associate respective body parts (Fig. 1).

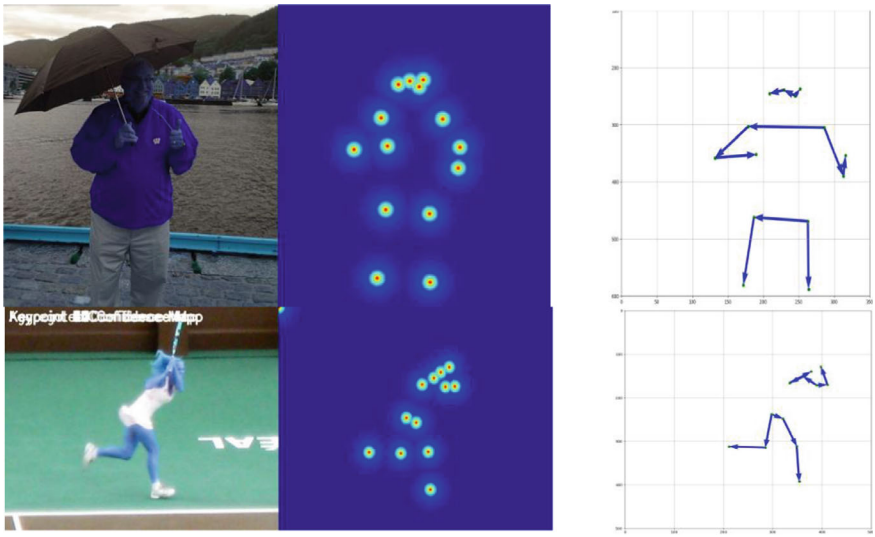


Fig. 1 Keypoints of the individual are recognized using confidence maps and part affinity fields are used to associate the body parts and estimate the pose

2 Literature Survey

Advances have been made in the field of estimating pose by employing cameras, such as the ZED camera, to capture 2D pictures [1]. Pose estimate data is extracted from RGB pictures. In the realm of 2D human pose estimation, there exists a distinguished study that is considered the pinnacle of excellence. The primary aim of this study is to apprehend and comprehend one's surroundings and anticipate the movements of various agents, deep learning models for autonomous driving vehicles (ADV) have been constructed utilizing data from a variety of photos [2]. Accidents occurred due to ADV's difficulty in determining when people intended to cross the road. Consequently, trajectory prediction and human stance became important.

As per the survey, we found the available datasets (Fig. 2) which can be used for prediction of the keypoints. We chose to use COCO dataset for its versatility. Out of the 330,000 photos in the COCO dataset [3], over 200,000 have been named. Bounding boxes are used to provide extensive information in the annotations for 1.5 million object instances. Additionally, each image has five captions attached to it which advances the field of image captioning study. Keypoints annotations are available for 250,000 individuals for a personal sample, which improves the dataset's needs for applications like posture estimation. Predicting three-dimensional keypoints correctly led to improved accuracy and precision in human pose estimation [4]. This allowed ADV to use an encoder-decoder architecture to evaluate individual trajectories. Multiple decoding heads were assigned to major tasks and auxiliary supervisions, while two encoding branches retrieved characteristics from context and

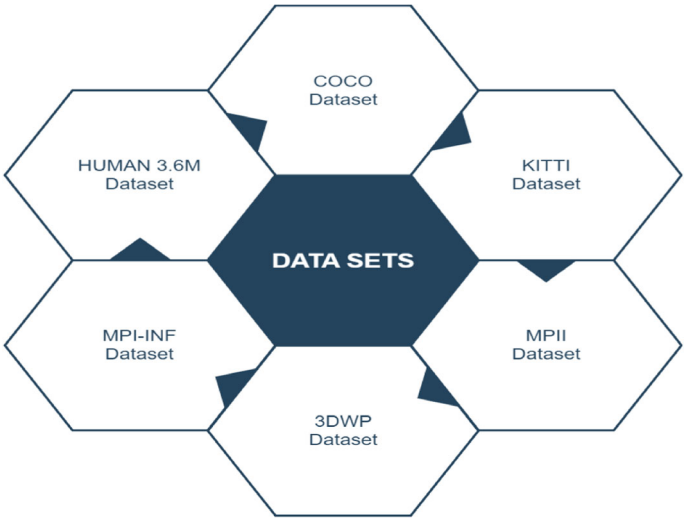


Fig. 2 Available datasets for pose estimation

keypoints. The problem arose when pedestrians walked parallel to the lane ADV had to anticipate whether they would obstruct it or slow down [1]. CNNs were employed for this purpose, and their ability to discern a human's direction of movement was based on four critical points including the nose, arms, elbows, left and right shoulders, and neck.

Pose Former, a unique transformer-based model for 3-dimensional human position prediction, is in the work [1]. To represent human joint interactions inside each frame and temporal correlations across frames, it makes use of spatial and temporal transformer modules [5]. Both Human3.6 and MMPI-INF-3DHP datasets are used. While the temporal transformer obtains the global dependency, the spatial transformer encodes the local interactions between 2-dimensional joints, demonstrating efficacy without the need for significant data segmentation. The model's performance is at the cutting edge putting a mini PC and 26 fps (frames per second) on a standard CPU while maintaining nearly the same accuracy as the baseline. The optimization involves a dilated MobileNet v1 feature extractor and a lightweight refinement stage. The OpenVINO Toolkit includes a network that has been enhanced to significantly improve the ratio of accuracy to network complexity. It is suitable for real-time applications on edge devices because of this improvement.

The Efficient Pose model family uses mobile inverted bottleneck convolutions [6] to provide computationally efficient detection blocks and an effective multi-scale feature extractor, all while maintaining increased pose configuration precision [7]. OpenPose's original implementation requires substantial computational resources, limiting its real-time applicability, especially on devices with CPU constraints. By optimizing the architecture and algorithms, the lightweight OpenPose [8] approach

enables real-time pose estimation even on devices with limited processing power, such as CPUs.

A 3D human posture estimation technique based on relational networks and the relationships between body components is presented in an article [9]. It uses relational networks to learn relations, group joints, and average features to estimate 3D poses. During training, a unique "relational dropout" strategy improves robustness to missing points. On the Human 3.6 M dataset, the suggested method achieves state-of-the-art 3D posture estimation, demonstrating efficacy even in the presence of missing joints. The relational dropout and the relational network's simplicity both lead to better performance and provide flexibility for different setups.

Building on CornerNet, the paper [10] presents CenterNet, a one-stage keypoint-based detection of the object framework. To improve the accuracy and memory, CenterNet identifies objects as triplets of keypoints—the center and corners. By examining core regions, the suggested cascade corner pooling and center pooling improve information enrichment and object detection. CenterNet performs comparably to the top-ranked two-stage detectors at faster inference rates, and when evaluated on MS-COCO [3], it obtains an AP of 57.0%, surpassing the performance of the current one-stage detectors by at least 5.9%. This method effectively investigates visual patterns inside suggested zones, resolving issues with current one-stage detectors.

3 Methodology

Figure 4 illustrates the main flow of our model. The person's pose in an image is estimated by the algorithm using input in the form of an image having a size of $w * h$. Initially, the feedforward network forecasts a set of 2D confidence maps S representing the body part positions. Next, a set comprising 2D vector fields is provided by the subsequent network sequence. A set of 2D vector fields 'L' of the part affinity fields (PAFs) mentioned in [8], which encode the degree of association between parts. There are j confidence maps for the collection $S = S_{1,\ldots}S_j$ where $n \in \{1\ldots. N\}$ for each component. There are C vector fields in the set $L = L_1, L_2 \ldots L_c$ one for each limb C where $L \in Rw \times h \times 2$ as well as $c \in \{1 \ldots C\}$.

In this paper, limb is referred to in a more generalized way to all the part pairs and it is not just for human limbs (e.g., the face). After the confidence maps and PAF prediction, the pose is estimated (Fig. 1).

3.1 Architecture

Figure 4 displays the architecture of the model, which learns about part affinity fields and gives the information about the part association. In the latter stage, the confidence map predicts the keypoints' presence at that location. This architecture

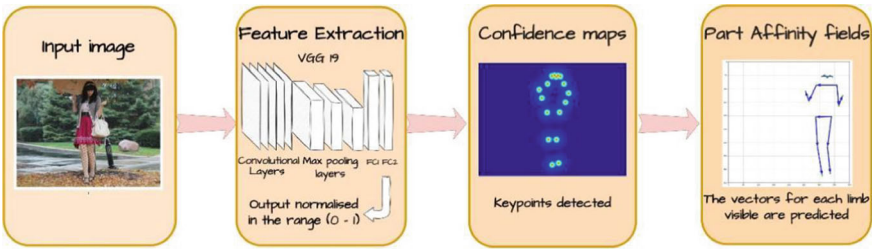


Fig. 3 Proposed pipeline

uses multiple-stage CNN to refine the predictions of PAFs and confidence maps and this played an important role in increasing our model's performance.

Proposed pipeline is shown in Fig. 3 where the input is given in the form of image to the model to predict the keypoint presence using confidence maps and part affinity field for body part location and orientation. As per the previous research [11], the model has 7 * 7 convolution layers, but in the current architecture, we have replaced the 7 * 7 kernel with 3 * 3 kernels. The number of operations for this architecture is 51. The output of the kernels is concatenated which is similar to the DenseNet model [12] (Fig. 4).

3.2 Prediction and Association

A convolutional neural network of the VGG-19 model, initialized by its first ten layers, analyzes the input image and creates feature maps F which are subsequently given to the initial stage as input. Network now generates a collection of part affinity fields (PAFs), such as L^1 = ϕ^1 (F), where ϕ^1 denotes the CNNs used for Stage 1 inference. Firm predictions are produced at each level by concatenating the input picture features F and the estimates from the earlier stage.

$$L^t = \phi^t\left(F, L^{t-1}\right),\ \forall 2 \leq t \leq T_P \tag{1}$$

where T_P denotes the total number of PAF stages, ϕ^t stands for the CNNs utilized at Stage t, and process for detecting confidence maps is the same and begins with the most recent PAF prediction.

$$S^{T_P} = \rho^t\left(F, L^{T_P}\right),\ \forall t = T_P, \tag{2}$$

$$S^t = \rho^t\left(F, L^{T_P}, S^{t-1}\right),\ \forall T_P < t \leq T_P + T_c \tag{3}$$

where T_C represents all of the stages in a confidence map and ρ^t denotes Convolutional Neural Networks for inference at Stage t. To allow the network to repeatedly

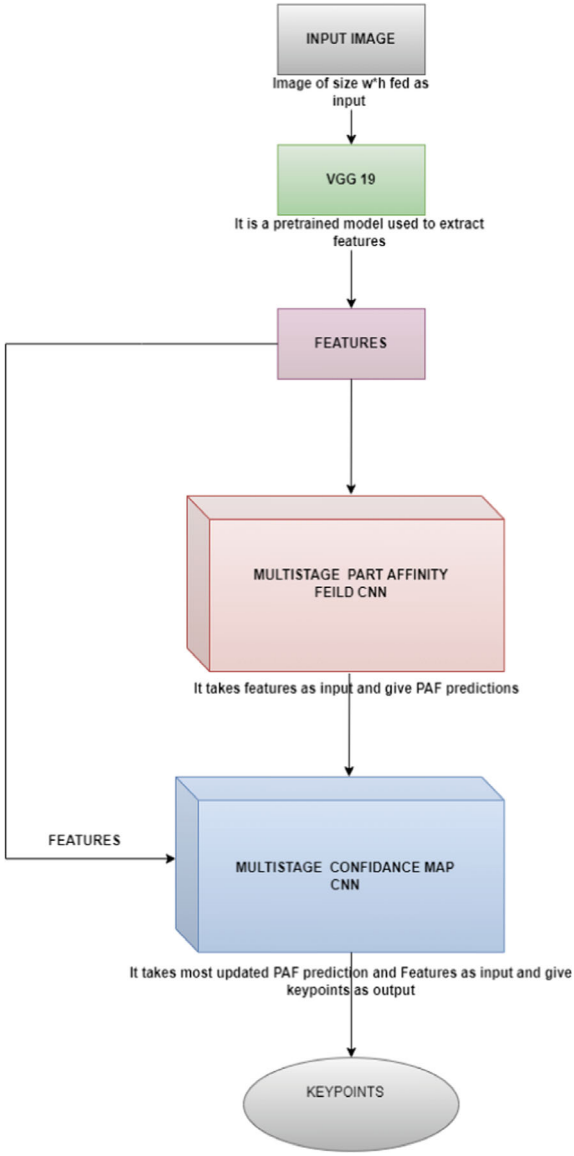


Fig. 4 Overall flow of the model

forecast confidence maps in the second branch and part affinity fields (PAFs) of body limbs in the first branch as done by Osokin [8], the authors used a loss function at the conclusion of every stage and have employed $L2$ loss among the fields and ground truth maps, and, in this instance, weighed the loss functions globally, the loss function of the PAF at stage t_i and the confidence maps at stage t_k,

$$f_L^{t_i} = \sum_{c=1}^{C} \sum_{p} W(p) \cdot \left\| L_C^{t_i}(p) - L_C^*(p) \right\|_2^2 \tag{4}$$

$$f_s^{t_k} = \sum_{j=1}^{J} \sum_{p} W(p) \cdot \left\| S_j^{t_k}(p) - S_j^*(p) \right\|_2^2 \tag{5}$$

where S_j^* is the ground truth part confidence map and L_c^* is the ground truth PAF and when there is no annotation at pixel p, W is a binary mask with $W(P) = 0$. In order to avoid penalizing the real predicted outcomes during the training phase, the mask is used. In this way, the vanishing gradient problem can be resolved by including checkpoints at every training phase. The main goal is

$$f = \sum_{t=1}^{Tp} f_L^t + \sum_{t=T_{P+1}}^{T_P+Tc} f_S^t \tag{6}$$

Confidence Maps

In the time of training, ground truth "Confidence Maps" 'S^*', which are created from the annotated two-dimensional keypoints, are used to solve the above given equation. Each confidence map is a two-dimensional representation of the belief that a body part is present in any of the given pixels. If only one person is depicted in the picture, each confidence map should show a single peak if the relevant portion is visible. If there is more than one person in the picture, there will be a peak for each person k and every viewable portion j. For every individual k, we produce unique confidence maps $S_{j,k}^*$. The value in $S_{j,k}^*$ at position $p \in R2$ is provided below.

$$S_{j,k}^*(p) = \exp\left(-\frac{\left\| p - x_{j,k} \right\|_2^2}{\sigma^2} \right) \tag{7}$$

where the σ controls the peak's spread. The network's expected ground truth confidence map is a sum of the individual confidence maps using a max operator, where

$$S_j^*(P) = S_{j,k}^*(P) \tag{8}$$

The ground truth confidence map that the network predicts is the sum of the individual confidence maps. To maintain the distinctness of the precision of close peaks, we need to obtain the highest value of the confidence maps rather than taking the average [11]. In Fig. 5, it is clearly shown how does the area of presence of keypoint can be predicted.

Fig. 5 This is the output of confidence maps, it tells us how confidently the keypoint is present at that location

Part Affinity Fields

After predicting the presence of the keypoints through the confidence map, it is very essential to associate the related body parts to build the whole-body posture. One technique to measure the relationship is to form an extra halfway for every two parts. Since these halfway points merely convey the part's position and not its direction of travel, they are vulnerable to establishing false associations in situations when an overwhelming number of people are present. PAFs overcome this restriction by maintaining the position and orientation data. The whole PAF is a 2D vector field. A two-dimensional vector records the axis that connects one portion of an appendage to another for every pixel. Consider a limb of the person in an image. Let $x_{j_1,k}$ and $x_{j_2,k}$ be the ground truth positions of the connecting parts j_1 and j_2. A unit vector that points from j_1 to j_2 is the value at $L^*_{c,k}(p)$ if a point p is on the portion, for all other points, the paf value is zero. It is defined as the ground truth PAF $L^*_{c,k}(p)$ in an image p as follows:

$$L^*_{c,k}(p) = \{v \quad \text{if } p \text{ is on limb } c, k \,||0 \text{ otherwise}$$

Here,

$$v = \frac{\left(x_{j_2,k} - x_{j_1,k}\right)}{\left\|x_{j_2,k} - x_{j_1,k}\right\|_2} \tag{9}$$

is a unit vector pointing in the limb's direction. The set of points on the limb is those that fall under the line segment's distance requirement, or those points p for which

$$0 \leq v \cdot \left(p - x_{j_1,k}\right) \leq l_{c,k} \text{ and } \left|v_\perp \cdot \left(p - x_{j_1,k}\right)\right| \leq \sigma_l \tag{10}$$

where the limb width is σ_l and the limb length is a distance in pixels, i.e.,

$$l_{C,k} = \left\|x_{j2,k} - x_{j_1,k}\right\|_2 \tag{11}$$

and $v_{\perp}$ is a vector which is 90° to v. The actual truth PAF takes the average of each person's affinity fields in the image,

$$L_c^*(p) = \frac{1}{\eta_c(p)}(p) \sum_k L_{c,k}^*(p) \tag{12}$$

where $\eta_c(p)$ is the whole count of vectors that are not zero among all k individuals at position p.

By calculating the line integral over the corresponding PAF along the line segment connecting the candidate part locations, we are able to measure the association between candidate part detections during testing. In essence, this quantifies how well the connected limb that would result from connecting the identified body components aligns with the projected PAF. More precisely, to determine the degree of confidence in the link between two candidate part locations d_{j1} and d_{j2}, a sample is drawn from the anticipated part affinity field, L_c along the line segment:

$$E = \int_{u=0}^{u=1} L_c(P(u)) \cdot \frac{d_{j2} - d_{j_1}}{\left\| d_{j_2} - d_{j_1} \right\|_2} du \tag{13}$$

where $p(u)$ determines the two body components' positions d_{j1} and d_{j2},

$$p(u) = (1 - u)d_{j1} + ud_{j2} \tag{14}$$

In practice, we get close to the integral by taking a sample and adding uniformly u spaced values. In Fig. 6, the example of how the limbs can be connected using PAF is visually shown.

Implementation Details

The model is trained for 100 epochs by using Adam optimizer [13] by setting the dropout ratio as 0.5 for VGG 19 layers. Normalization layer is added after convolutional layer except the last 19th layer. In the process of training of the COCO dataset,

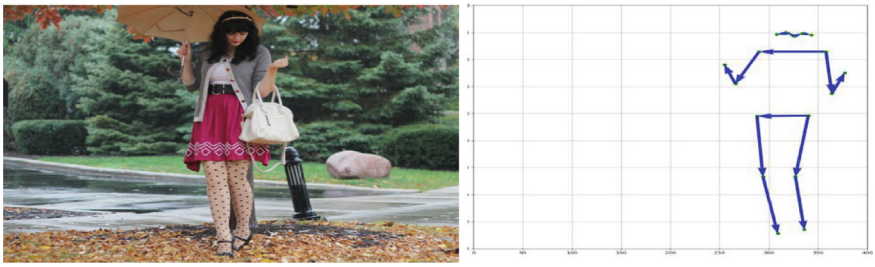


Fig. 6 This is the output of Part affinity fields; it does the part association and gives the information about body part location and orientation

we used Nvidia A6000 GPU with 128 GB RAM. Batch size is set to 65 while the learning rate and the decay rate is set as $1 * 10^{-4}$. Up to convergence, the network architecture used in the evaluation section is trained at the optimal learning rate. The results are shown in the below tables.

4 Results

The authors have evaluated the model on the COCO dataset (2016) [3]. This dataset provides the images which are wildly possible. The images are diverse and include an enormous number of practical difficulties, including contact, obstruction, congestion, and scale variation.

We have compared our method (bottom-up approach) with the other studies using PCK as a performance indicator, which is defined as the ratio of properly identified keypoints to the total number of keypoints, expressed as a percentage. Table 1 displays the PCK at 0.5 score, which indicates the proportion of accurately estimated keypoints when the difference between the estimated and ground truth keypoints is 50% or less of the reference distance, which is generally the torso length.

In this way, the PAF and confidence maps stand apart from the existing methods in single-person pose estimation and the uniqueness of the method is reflected in the results as shown in Tables 1 and 2. We also compared our method with the paper by Li et al. [4] that employs a flexible combination of the templates represented in linear SVMs. The accuracy values for every keypoint at error thresholds of 0.1 (which means that the predicted keypoint is within 10% distance of the ground truth annotation) and 0.15 (which means that the predicted keypoint is within 15% distance of the ground truth annotation) are compared in Table 1 with [17].

Table 1 Comparison of our proposed method (bottom-up approach) and existing methods for single-person pose estimation in terms of PCK (percentage of correct keypoints) at 0.5 score

Method	Shoulder	Knee	Elbow	Head	Wrist	Hip	Ankle	Mean
[14]	95.2	85.7	89.9	97.3	85.3	89.4	81.9	88.2
[15]	96.3	87.4	91.2	98.2	87.1	90.1	83.6	90.9
[12]	96.3	88	91.9	98.5	88.1	90.6	85	91.5
[16]	96.7	88.6	92.5	98.5	88.7	91.1	86	**92**
Proposed	**96.8**	**89**	**92.7**	**98.7**	**89**	**91.5**	**86.2**	91.9

Table 2 Accuracy at error thresholds of 0.10 and 0.15 for each of the 17 keypoints

Keypoints	Accuracy for 0.10 threshold		Accuracy for 0.15 threshold	
	[17]	Ours	[17]	Ours
Nose	Head 56.16	**84.5**	Head 77.76	**92.1**
L eye		**77.4**		**87.6**
R eye		**75.7**		**86.9**
L ear		**79.0**		**89.6**
R ear		**75.2**		**90.1**
Left hip	38.43	**77.7**	58.61	**85.8**
Right hip	34.96	**72.5**	58.09	**86.7**
Left shoulder	53.21	**65.9**	72.75	**86.3**
Right shoulder	55.39	**67.8**	74.03	**86.1**
Left wrist	24.29	**81.2**	38.17	**94.3**
Right wrist	21.72	**70.9**	33.16	**89.8**
Left knee	39.07	**69.8**	56.94	**83.8**
Right knee	38.43	**69.9**	57.32	**90.1**
Left ankle	32.26	**78.9**	49.61	**91.2**
Right ankle	31.10	**79.8**	48.20	**85.3**

The comparison demonstrates that our approach outperforms [18] on all keypoints. This provides a clear overview of the authors' contribution

4.1 Visuals of the Results

Combining the output of all stages where the confidence maps and part affinity fields depict the keypoints of the human in the image is shown in Fig. 7 along with the visual outputs at the end.

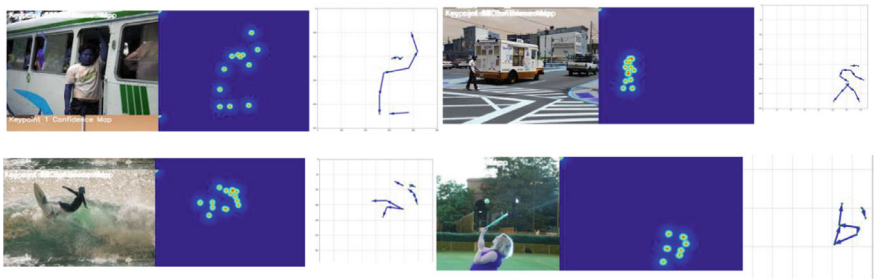


Fig. 7 Input images along with their respective confidence maps and part affinity fields which resemble the pose of the individual in the image

5 Conclusion and Future Scope

Real-time 2D pose estimation has a crucial importance for many machines around us which have to visually understand and interpret human poses. For single-person pose estimation, many of the authors have used the spatial model [5], which is alternatively based on non-tree modeling that adds further edges to the tree architecture to represent obstruction or trees-structured graphic representations which store the spatial connection among neighboring pieces along a kinematic chain, and also symmetry, and other relationships. Many other researchers have also done the single-person posture estimations, which have improved precision, primarily using Convolutional Neural Networks, to perform the body's immediate component monitoring. Authors convey in the current research an explicit non-parametric explanation of the keypoint association that encodes the human body's limbs' position and location. Secondly, the model develops an architecture that integrates portion verification and association synchronously. The refinement of Part Affinity Fields (PAF) using a multi-stage CNN is significantly more important than the combined process of merging PAF with physique component placement. This refinement improves both execution efficiency and accuracy in multi-period scenarios, where there are numerous potential limb combinations. By accurately identifying the correct pairings of body parts, we ensure better recognition of connected components. To achieve this, the PAF scores are utilized to evaluate linear associations, capturing contextual background information effectively. This approach is essential for extending our work to multi-person pose estimation, allowing robots to learn a variety of poses and adapt to different applications, thereby becoming more versatile.

References

1. Zheng C, Zhu S, Mendieta M, Yang T, Chen C, Ding Z (2021) 3d human pose estimation with spatial and temporal transformers. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 11656–11665
2. Papaioannidis C, Mademlis I, Pitas I (2022) Fast CNN-based single-person 2D human pose estimation for autonomous systems. IEEE Trans Circuits Syst Video Technol 33(3):1262–1275
3. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Zitnick CL (2014) Microsoft coco: Common objects in context. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. Springer International Publishing, pp 740–755
4. Li J, Shi X, Chen F, Stroud J, Zhang Z, Lan T, Li C (2023) Pedestrian crossing action recognition and trajectory prediction with 3D human keypoints. arXiv preprint arXiv:2306.01075
5. Cheng Y, Yang B, Wang B, Tan RT (2020) 3d human pose estimation using spatio-temporal networks with explicit occlusion training. Proc AAAI Conf Artif Intell 34(07):10631–10638
6. Groos D, Ramampiaro H, Ihlen EA (2021) EfficientPose: scalable single-person pose estimation. Appl Intell 51:2518–2533
7. Zhang F, Zhu X, Ye M (2019) Fast human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3517–3526
8. Osokin D (2018) Real-time 2d multi-person pose estimation on cpu: lightweight openpose. arXiv preprint arXiv:1811.12004

9. Park S, Kwak N (2018) 3D human pose estimation with relational networks. arXiv preprint arXiv:1805.08961
10. Abu Ghalieh KM, Alawneh SG (2020) Predicting pedestrian intention to cross the road. IEEE Access 8:72558–72569
11. Cao Z, Simon T, Wei SE, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7291–7299
12. Chu X, Yang W, Ouyang W, Ma C, Yuille AL, Wang X (2017) Multi-context attention for human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1831–1840
13. Kingma D, Adam JB (2015) A method for stochastic optimization. CoRR, abs/1412.6980
14. Bulat A, Tzimiropoulos G (2016) Human pose estimation via convolutional part heatmap regression. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. Springer International Publishing, pp 717–732
15. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14. Springer International Publishing, pp 483–499
16. Yang W, Li S, Ouyang W, Li H, Wang X (2017) Learning feature pyramids for human pose estimation. In: Proceedings of the IEEE international conference on computer vision, pp. 1281–1290
17. Yang Y, Ramanan D (2011) Articulated pose estimation with flexible mixtures-of-parts. In: CVPR 2011. IEEE, pp 1385–1392
18. Dantone M, Gall J, Leistner C, Van Gool L (2013) Human pose estimation using body parts dependent joint regressors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3041–3048

Disease Classes Identification and Association Rules Extraction Using Medical Records

L. R. Aishwarya, V. R. Sai Pooja, and Swetha Rameswaran

Abstract Massive volume of data is generated in the healthcare industry that includes numerous unstructured information and texts about the hospitals, patients, diseases, medical equipment, drugs, etc. Deployment of extensive data mining and data treatments will refine the data and provide more actionable insights than what is observed in the raw medical data and documents. This paper proposes a methodology of handling such medical records as a semi-supervised learning problem to formulate an assistive technology for predicting the topographic classes (Digestive system, Cardiovascular, Nervous, General Pathological Conditions, Neoplasms) of the diseases discussed in the records by leveraging natural language processing techniques. A topographic class classifier model has been built that predicts the class based on the contents described in the text. Models such as Support Vector Machine, Random Forest, XGBoost, and a few LLMs were tested for this purpose. Unified Medical Language Systems (UMLS) is a repository of medical terms and vocabulary that helps in tagging the type of a text based on the predefined classes. Using UMLS, the text is analyzed for identifying underlying diseases, symptoms, treatment, and drug-related information and tagged. This forms the base data to perform rule mining on the text to identify the core connections between diseases, symptoms, drug, and so on to find their co-occurrence. The reasoning behind this analysis is to uncover the hidden co-occurrences of diseases, symptoms, treatments that otherwise go unnoticed. This will provide more insights about drug behaviors, adverse events, disease-drug reactions, and so on that would help in refining the clinical trials.

Keywords Topographic classifier models · Rule mining · UMLS tagging · Diseases association

L. R. Aishwarya (✉) · V. R. Sai Pooja · S. Rameswaran
Pfizer Healthcare India Private Limited, Chennai, India
e-mail: aishwarya.lr@pfizer.com

V. R. Sai Pooja
e-mail: saipooja.vr@pfizer.com

S. Rameswaran
e-mail: swetha.rameswaran@pfizer.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_45

1 Introduction

The technological advancements in healthcare industry have led to the generation of enormous amount of data over the recent years and that is increasing exponentially with every passing year. With healthcare industry contributing to almost 30% of the world's total data volume (Fig. 1) as pointed out by RBC (Royal Bank of Canada) Capital [1], there are abundant scopes and possibilities presented to analyze and transform the data into actionable, meaningful insights for the betterment of the human welfare.

One such potential source of information is the medical health records that contain information about patients, symptoms, drugs, dosages, and so on. Developing an automated system that could assist in reading through complex medical health records and interpreting the key insights would play a pivotal role to make it consumable. Natural language processing has been a boon in the field of machine learning that converts qualitative text data to a quantifiable format. By implementing the NLP procedures, one can transform several millions of text data to powerful insights that capture the core of its underlying principles. Medical health records captured historically along with the class of diseases serve as a foundation for this approach. This is framed as a semi-supervised learning problem as the classes of diseases are identified only for a fraction of the total health records. This is used to train a machine learning model using natural language processing algorithms that would aid in predicting the class of disease a particular medical text belongs to. Formulating an assistive technological algorithm that could predict this would be extremely helpful for drug developers and healthcare professionals as they can focus more on identifying the specific disease given the class of the disease. Once the class of disease is identified,

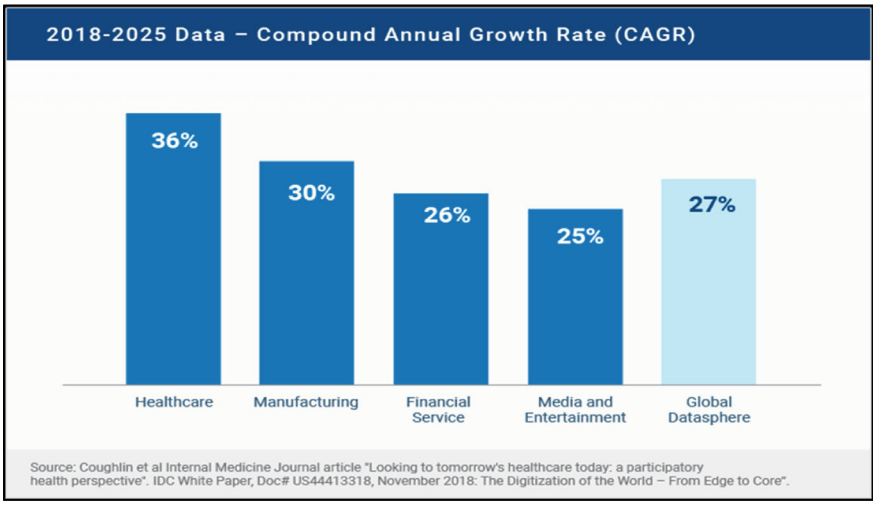


Fig. 1 Data-compound annual growth rate

that could be used as an input to the rule mining algorithm to find the association between symptoms, diseases, and drugs. These rules would be very valuable in understanding the relation between them which otherwise can go unnoticed given the volume of data handled. They could serve as indicators for drug development process and identifying treatment procedures to patients.

2 Literature Review

Based on [2], research has been done to use supervised learning algorithms to identify or diagnose a disease from medical text. Various algorithms like Decision Tree, Support Vector Machine, K-Nearest Neighbor, Naïve Bayes, Logistic Regression, AdaBoost, etc. and deep learning algorithms like Convolutional Neural Networks (CNN) were used. According to [3], a novel medical diagnosis system was developed considering this as a multi-class problem with optimal set of features that corresponds to heart disease and was attempted to solve using feature selection methods. The classification algorithms like Random Forest (RF), Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Extreme Gradient Boost (XGBoost) and ensemble learning classifiers (Stacking and Voting) were used to achieve the multi-class classification. The performance measures such as Receiver Optimistic Curve (ROC), accuracy precision, f1-score, and recall were calculated and considered in this process.

In accordance with [4], leveraging transformer-based large language models such as BERT, RoBERTa, XLNet, DistilBERT with proper combination of strategies is suggested to highly impact to provide positive outcomes, when weighed higher, with the right set of pre-processing techniques. Pre-processing is intensely carried out when it comes to text data which is always better as suggested in [5]. Techniques that include tokenization through NLTK, handling stop words, spelling errors, numbers, and many such noises in the data that may not add value to the model are experimented and dealt with. This has improvised the accuracy as the model is trained with a clearer data with the information that provides an advantage to the model. Furthermore, the better utilization of pretrained GPT-2 transformer model is discussed in [6]. Title prediction task was carried over here which clearly provides a better understanding toward which class the given text belongs to, as an output, enabling the end user to refine the final view toward the data. Building a CNN model with the required parameters that will weigh the features in our data is highly essential, Bin et al. [5] provides more insights on techniques involved such as pooling, regularization, and other important hyperparameters along with the model variations of CNN architecture.

On moving toward neural networks, with an understanding that NNs have given remarkable results in the fields of computer vision and speech recognition, Kim [7] discusses the methodology and intuition behind using CNN for a text classification problem in addition to leveraging several NLP techniques. The approach followed

used a single layer CNN algorithm on top of word vectors obtained from an unsupervised neural language model. With further fine-tuning that aligns with the task, model was able to produce excellent results which concluded that the ideology of leveraging pretrained vectors for the task in hand provides improved results, given the respective fine-tuning and pre-processing.

According to [8], the suggested approach was to extract association rules from medical records and rule mining algorithm is used along with multi-criteria decision analysis. The purpose served here was to identify the interrelationship among diseases, symptoms, and drugs. Stanford parser which can generate constituency and dependency parses of sentences was used for deriving syntactic relations among words in sentences and then UMLS was used to segregate the medical terms. Apriori algorithm, FP Growth, Eclat, Apriori TID, Relim, and Close algorithms were used for finding association rules and to choose the best algorithm among these, Electre 1 method for multi-criteria decision analysis has been done. Among these algorithms, RElim, Apriori TID, and FP-Growth algorithms have been more effective compared to the other algorithms based on the multi-criteria decision analysis. Interesting rules have been calculated with the lift value of the rules. A new approach which combines rule-based features and knowledge-guided deep learning models for effective disease classification is proposed by Yao et al. [9]. The authors have deployed a CNN model with word embeddings that also deploys entity from UMLS library.

The objective behind this paper is to build a classifier to predict the topographic classes of the medical text and then use the text within each of the classes to identify strong associations. Generating rules specific to a topographic class would help in generating more specific and strong medical associations. The future work of this paper is to extend the rules generated back to the data processing stage for better prediction of classes. The process flow of the methodology adopted for this paper is as follows (Fig. 2).

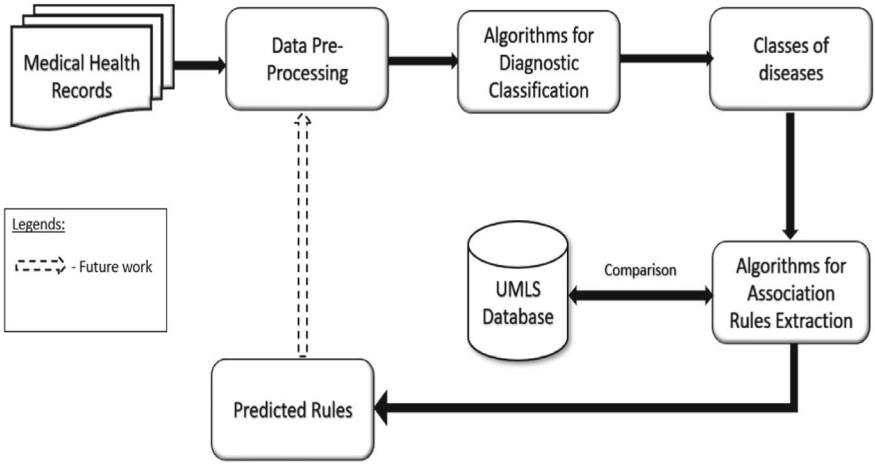


Fig. 2 Flow diagram

3 Methodology

3.1 Topographic Class Classifier

Topographic classification of diseases is considered for this paper for assigning the classes to the medical texts. Topographic classification is based on the body region from where a disease originates from. Classes such as Digestive system, Cardiovascular, Nervous, General Pathological Conditions, Neoplasms were used for classifying the medical text into their respective topographic class [10]. The dataset used to build this classifier model is an open-source medical health records data taken from Kaggle [11]. This comprises 15,000 free texts and their respective topographic class. Figure 3 shows the number of records present in the input data under each target.

The paper has modeled and tested several machine learning and deep learning models to build the medical text topographic classifier, namely, SVM, KNN, XGBOOST, bi-directional LSTM-CNN, and other transformer-based LLMs that include BlueBERT, XLNet, RoBERTa, and GPT2.

Most of the models mentioned above provided optimal results with few distinct variations which are discussed in the section below post the pre-processing methodology adopted for the study.

Pre-processing

From a text classification perspective, text cleaning and pre-processing play a crucial role in defining the output of the classifier model. There are a high chances of

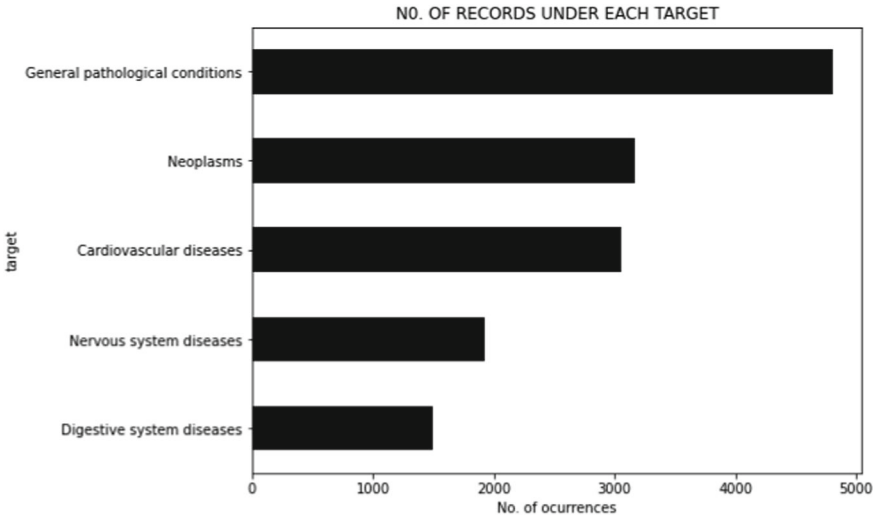


Fig. 3 Total records against each target

deviating from a true output if the input texts are not rightly processed. The pre-processing deployed in this study includes stop words removal, punctuation and lowercase removal, and numbers handling in the free text. Alongside the above steps, lemmatization technique is also implemented which finds each root word of the given text input thereby reducing the bias created based on grammatical differences.

Another important pre-processing step that is carried out for LLMs and CNN is tokenization, which is used create tokens for the text input provided, that is fed into the model. This type of tokenizer used varies for each model as each of it requires different tokenization strategy that should align with the model's characteristics. For a few models, TF-IDF vectorizer is used rather than tokenization models for vectorizing the pre-processed texts that are then processed by the classifiers.

Models

Bi-LSTM CNN: The model is initialized by leveraging GLoVe word vectors as input to the convolving layers of CNN alongside bi-directional LSTM layer, in which two LSTMs work parallelly, one taking forward directional input while the other being backward. Dense layers are introduced which involves the usage of dropout technique for avoiding overfitting. Max-pooling operation is used to extract the most significant feature. SoftMax activation function is used for the model, as this is a multi-class classification problem. In addition to this, loss layer has been added. Though the accuracy obtained using this classifier is acceptable, it can still be improved as some of the machine learning models were outperforming its results.

ML Models: Following the TF-IDF vectorizer on the pre-preprocessed text, three models are tested with the sparse matrix vectors from this layer. The models include SVM, KNN, and XGBOOST. Though the results obtained from all the 3 models were reasonable, SVM outperformed the other two by having hightest score in most of the metrics.

Transformer-Based LLMs: The pre-processed texts had to be tokenized to feed into the pretrained LLMs. Then, the tokenized data is converted to tensors and fed into the LLMS. Five LLMs are tested for this classification purpose that include Blue-BERT, GPT2, RoBERTa, XLNET, and DistilBERT. Among the five, GPT2 was able to show better outcomes when it is coupled with Keras NN layers. With further tweaking, more improvement in result can be expected. The other model's results are significantly lower than the GPT2, hence are dropped from the further analysis.

The accuracy metrics of the top 5 models are tabulated (Table 1), followed by the accuracy comparison graph (Fig. 4).

By looking through the accuracy scores, it is evident that Bi-LSTM-CNN, GPT2, and SVM results are higher than the other models. For further analysis, this paper has considered SVM classifier.

Table 1 Topographic class classifier-model accuracy

Model	Accuracy	Precision	Recall	F1-Score
Bi-LSTM-CNN	66.5	67	65	66
SVM	78	75	76	75.5
KNN	72	72	74	73
XGBOOST	74	72	76	74
GPT2	71	71	70	70.5

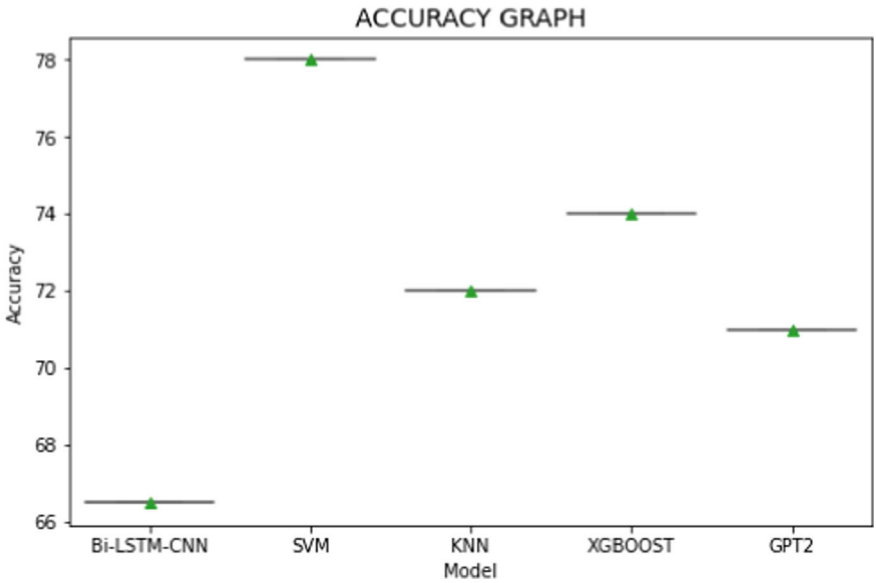


Fig. 4 Accuracy comparison of the models

3.2 UMLS

UMLS (Unified Medical Language System) [12] which is a thesaurus that consists of all medical terminologies and vocabulary associated with a disease, drug, symptom, medical device, etc. can be used to understand the medical terms present in the free text as it generates the mapping of these terms to the exact class that it links to. The free texts are passed through the UMLS server to get the phrases tagged appropriately to the corresponding UMLS lexicon with respect to its CUI (Concept Unique Identifier). This UMLS lexicon is in turn mapped to the exact concept/context that it refers to. Among all the contexts that were identified, the disease, symptom, and drug-related contexts have been filtered out. This subset of data generated from UMLS server along with the identified topographical disease class is used to create a dataset that would act as an input to the Apriori algorithm to further identify the

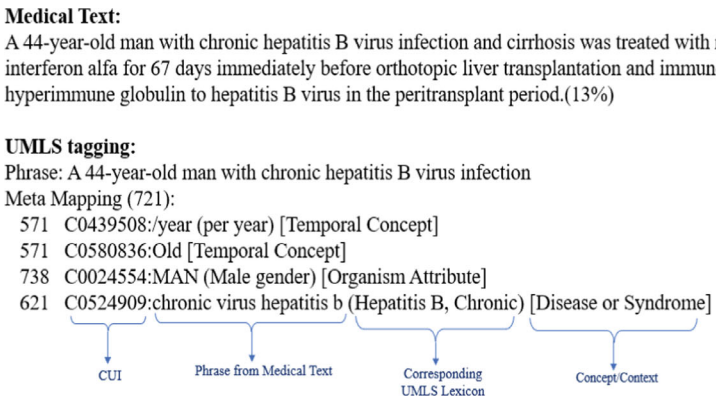


Fig. 5 UMLS tagging

association rules. Figure 5 is an example of identification of context from the phrases present in the medical text by the UMLS.

3.3 Association Rule Generation

An association rule is of the form A → B, where A is referred to as the antecedent and B is the consequent [8]. Association rules are used to find the correlation between terms in textual data [8]. There are several algorithms available to generate the association rules, however, for simplicity, this paper has adopted Apriori algorithm to generate the rules. The medical texts are mined to identify the most common co-occurring diseases, symptoms, and drugs in the medical text. Firstly, the medical texts are classified into their respective topographic classes using the classifier to validate if the rules are specific to their respective classes. Using this process, over 2500 rules were extracted across all the classes. Each of the rules was then analyzed and the top associations (Fig. 6) were identified in each of the classes using metrics such as lift, support, and confidence. However, more emphasis was given to lift as it signifies the type of correlation as in positive or negative that would lead to more clarity [13]. The top 42 associations across all the classes are presented in the appendix (Table 5).

4 Discussion

The topographic class classifier layer enabled us to narrow down toward a particular class rather than looking onto whole medical concept as one. While the results are optimal, it clearly shows that few improvements can still be done. One key factor that should be considered is that patient records do not fall under one class. There

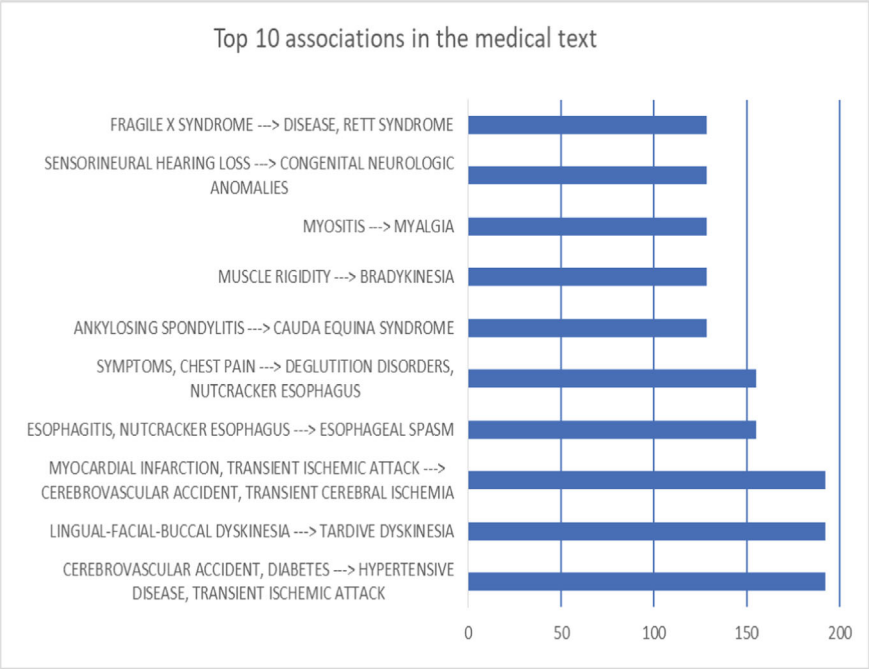


Fig. 6 Top 10 rules extracted from the medical text

is a higher chance of a particular patient having multiple diagnosis that falls under multiple classes. So, for further enhancement, multi-label classification technique must also be introduced for increasing the accuracy.

Another exercise that is currently in pipeline is playing around the LLMs and leveraging it alongside the models that are providing better results as an ensemble model. This might provide us with better insights on understanding what is the antecedent class and the precedent class which can also be leveraged in association rule mining.

The association rule mining exercise helped in bringing out the patterns of medical conditions most commonly discussed in the medical writings. While some of the associations are more commonly observed, it also gave insights on some uncommon patterns. For instance, in neoplasms class bulbo-spinal atrophy was highly associated with mucocutaneous lymph node syndrome, though they are not interlinked obviously as per medical literature. Table 2 shows such correlations that were mined using the medical texts.

Such interactions can be reported to medical practitioners and drug developers to research more on these lines and understand the reasons for their co-occurrences. They could play a vital role in disease diagnostics, drug development practices. There were also common, interesting associations between symptoms and diseases identified in the analysis presented in Table 3.

Table 2 Unnoticeable associations

Class	Associations	Support	Confidence	Lift
Neoplasms	Mucocutaneous lymph node syndrome → Bulbo-spinal atrophy	0.011	1.000	94.000
Digestive system diseases	Acquired immunodeficiency syndrome → Cryptosporidiosis	0.006	0.250	25.833
Digestive system diseases	Mastodynia → Coronary arteriosclerosis	0.006	0.571	5.408
Cardiovascular diseases	Dental plague → Myocardial infarction	0.006	0.444	4.535
Digestive system diseases	Encephalopathies → Ascites	0.006	1.000	31.000
Cardiovascular diseases	Chiari malformation type II → Rheumatic mitral stenosis	0.006	0.333	27.208

Table 3 Associations between symptoms and diseases

Class	Associations	Support	Confidence	Lift
Digestive system diseases	Chest pain → Nutcracker oesophagus	0.01	1.00	155.00
Nervous system diseases	Muscle rigidity → Bradykinesia	0.01	1.00	128.33
Nervous system diseases	Angina pectoris → Coronary arteriosclerosis	0.01	1.00	96.25
Cardiovascular diseases	Angina pectoris → Myocardial Ischemia	0.01	0.17	4.06
Nervous system diseases	Myositis → Myalgia	0.01	1.00	128.33

These associations are very helpful to patients and medical doctors to understand the kind of symptoms associated with the underlying medical conditions. Finally, the relationships between two medical conditions where one can lead to the other are the most important take away from this analysis. Table 4 are the associations between two diseases or conditions that are commonly observed and found frequently in medical literature.

They highlight the relation between two medical conditions or the likelihood of getting one if the other is already present, thereby providing much more need for more research in this space to understand the casual relationships. This plays a pivotal role in urging drug developers and medical practitioners to be aware of such consequences so that they can plan better while treating the patients. Also, all the recurring complications of a specific medical condition can be well understood through these patterns, for it has been well discussed in the literature about the coexistence of diabetes and hypertension and the likelihood of getting one given the presence of the other [14]. Medical practitioners have to take care in treating the patients not just for the current condition but also to the possible disorders that can happen in the future, that could lead to much more healthier society.

Table 4 Commonly observed associations

Class	Associations	Support	Confidence	Lift
Digestive system diseases	Alcohol effect → Liver cirrhosis	0.006451613	1	11.92307692
Cardiovascular diseases	Hypertension → Diabetes mellitus	0.0061225574	1	22.51724138
Cardiovascular diseases	Kidney failure → Hypertensive disease	0.0061225574	0.8	4.882242991
Cardiovascular diseases	Coronary arteriosclerosis→ Diabetes mellitus	0.0061225574	1	22.51724138
Digestive system diseases	Zollinger-Ellison syndrome → Duodental ulcer	0.006451613	1	23.84615385

The paper also had anticipated some significant associations between drugs and symptoms/diseases. But, since not many drugs related contents were available in the medical texts taken, no such significant rules emerged in this process. This can be extended as a future scope to consider more drug-related medical text for analysis to check for any associations between the drugs and the related composition and dosage with the underlying symptoms and diseases.

5 Summary

The aim of this work is to build a classifier to predict the topographic classes of medical heath records that would help in faster diagnosis during treatments. Also, this classifier model is used to identify the hidden patterns in the text to bring out significant associations between symptoms, diseases, and drugs specific to each topographic class. UMLS is used to tag the various medical terms, and, along with association rule mining, it helped in identifying co-occurrences of diseases and symptoms in the medical records. While some of the associations are medically proven, there emerged a few associations that were not proved in the previous medical literature. This calls the attention of the medical practitioners, drug developers, and healthcare professionals to do more research to understand these associations and validate them. This paper can further be extended to feed in the insights from association rules back to the text processing for better classification and analysis.

Appendix

See Table 5.

Table 5 Top 42 association rules identified across the five topographic classes

Class	Antecedent	Consequent	Support	Confidenc	Lift
Neoplasms	Mucocutaneous lymph node syndrome	Bulbo-spinal atrophy, X-Linked	0.010638	1	94
Digestive system diseases	Esophagitis, Nutcracker Esophagus	Esophageal spasm	0.006452	1	155
Digestive system diseases	Symptoms, chest pain	Deglutition disorders, Nutcracker esophagus	0.006452	1	155
Digestive system diseases	Nutcracker esophagus, Non-cardiac chest pain	Mastodynia	0.006452	1	103.3333
Digestive system diseases	Symptoms, Acquired immunodeficiency syndrome	Cryptosporidiosis	0.006452	1	103.3333
Digestive system diseases	Organic disorders	Irritable bowel syndrome	0.006452	1	77.5
Digestive system diseases	Trichohepatoenteric syndrome	Irritable bowel syndrome	0.006452	1	77.5
Digestive system diseases	Gangrene	Intestinal obstruction	0.006452	1	44.28571
Digestive system diseases	Terminal esophageal web	Deglutition disorders	0.006452	1	38.75
Digestive system diseases	Crohn disease, Inflammatory bowel diseases	Ulcerative colitis	0.012903	1	31
Digestive system diseases	Encephalopathies, Liver cirrhosis, Cirrhosis	Ascites	0.006452	1	31
Digestive system diseases	Zollinger-Ellison syndrome	Duodenal ulcer	0.006452	1	23.84615
Digestive system diseases	Helicobacter pylori infection	Peptic ulcer	0.009677	0.6	15.5
Digestive system diseases	Alcohol effect, Cirrhosis	Liver cirrhosis	0.006452	1	11.92308
Nervous system diseases	Cerebrovascular accident, Diabetes	Hypertensive disease, transient ischemic attack	0.005195	1	192.5

(continued)

Table 5 (continued)

Class	Antecedent	Consequent	Support	Confidenc	Lift
Nervous system diseases	Lingual-facial-buccal dyskinesia	Tardive dyskinesia	0.005195	1	192.5
Nervous system diseases	Myocardial infarction, Transient ischemic attack	Cerebrovascular accident, transient cerebral isch	0.005195	1	192.5
Nervous system diseases	Ankylosing spondylitis	Cauda equina syndrome	0.005195	1	128.3333
Nervous system diseases	Muscle rigidity	Bradykinesia	0.005195	1	128.3333
Nervous system diseases	Myositis	Myalgia	0.005195	1	128.3333
Nervous system diseases	Sensorineural hearing loss	Congenital neurologic anomalies	0.005195	1	128.3333
Nervous system diseases	Fragile x syndrome	Disease, rett syndrome	0.005195	0.666667	128.3333
Nervous system diseases	Angina pectoris	Coronary arteriosclerosis, cerebrovascular accid	0.005195	1	96.25
Nervous system diseases	Intracranial hypertension	Meningitis, bacterial, meningitis	0.007792	1	77
Nervous system diseases	Pseudomeningocele	Cyst	0.005195	1	77
Nervous system diseases	Neuropathy	Eosinophilia-myalgia syndrome	0.005195	0.333333	32.08333
Nervous system diseases	Thyroid hormone plasma membrane transport defe	Cctyst	0.005195	0.4	30.8
Nervous system diseases	Quadriplegia	Cerebral palsy	0.005195	0.5	27.5
Nervous system diseases	Quadriplegia	Littles disease	0.005195	0.5	19.25
Cardiovascular diseases	Cerebral infarction	Cerebral ischemia	0.006126	0.571429	93.28571
Cardiovascular diseases	Chiari malformation type ii	Rheumatic mitral stenosis	0.006126	0.333333	27.20833
Cardiovascular diseases	Coronary arteriosclerosis, hypertensive disease, dia	Diabetes mellitus	0.006126	1	22.51724

(continued)

Table 5 (continued)

Class	Antecedent	Consequent	Support	Confidenc	Lift
Cardiovascular diseases	Hypertensive disease, hyperglycemia	Diabetes mellitus	0.006126	1	22.51724
Cardiovascular diseases	Mastodynia	Coronary arteriosclerosis	0.006126	0.571429	5.407867
Cardiovascular diseases	Disease, kidney failure	Hypertensive disease	0.006126	0.8	4.882243
Cardiovascular diseases	Hypernatremia	Hypertensive disease	0.006126	0.8	4.882243
Cardiovascular diseases	Dental plaque	Myocardial infarction	0.006126	0.444444	4.534722
Cardiovascular diseases	Angina pectoris	Myocardial ischemia	0.006126	0.173913	4.055901
General pathological conditions	Sensory discomfort	Malaise	0.008686	1	61.4
General pathological conditions	Cerebrovascular accident	Cerebral infarction	0.006515	0.26087	24.02609
General pathological conditions	Coronary arteriosclerosis	Myocardial infarction	0.005429	0.238095	8.771429
General pathological conditions	Abscess	Communicable diseases	0.005429	0.357143	4.837185

References

1. Coughlin S, Roberts D, O'Neill K, Brooks P (2018) Looking to tomorrow's healthcare today: a participatory health perspective. Intern Med J 48(1):92–96. https://doi.org/10.1111/imj.13661. PMID: 29314515
2. Ahsan MM, Luna SA, Siddique Z (2022) Machine-learning-based disease diagnosis: a comprehensive review. Healthcare 10(3):541. https://doi.org/10.3390/healthcare10030541
3. Ahmed T, Qaisar S (2022) A novel web-based multi-class heart disease prediction using machine learning algorithms. In: International conference on electrical engineering and sustainable technologies (ICEEST)
4. Shaheen Z, Wohlgenannt G, Filtz E (2020) Large scale legal text classification using transformer models. arXiv:2010.12871
5. He B, Guan Y, Dai R (2019) Classifying medical relations in clinical text via convolutional neural networks. Artif Intell Med 43–49. https://doi.org/10.1016/j.artmed.2018.05.001
6. Puri R, Catanzaro B (2019) Zero-shot text classification with generative language models. arXiv:1912.10165v1
7. Kim Y, New York University (2014) Convolutional neural networks for sentence classification. arXiv:1408.5882
8. Lakshmi KS, Vadivub G (2017) Extracting association rules from medical health records using multi-criteria decision analysis. In: 7th International conference on advances in computing & communications (ICACC-2017)
9. Yao L, Mao C, Luo Y (2019) Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. BMC Med Inform Decis Mak 19(Suppl 3):71
10. Robbins SL, Robbins JH, Scarpelli DG (2023) Classifications of diseases. https://www.britannica.com/science/human-disease/Classifications-of-diseases
11. Schopf T, Braun D, Matthes F (2023) Evaluating unsupervised text classification: zero-shot and similarity-based approaches. In: Proceedings of the 2022 6th International conference on natural language processing and information retrieval (NLPIR '22). Association for Computing Machinery, New York, NY, USA, pp 6–15. https://doi.org/10.1145/3582768.3582795
12. Bodenreider O (2004) The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res 32:D267–D270. PubMed PMID: 14681409; PubMed Central PMCID: PMC308795. https://doi.org/10.1093/nar/gkh061
13. Hussein N, Alashqur A, Sowan B (2015) Using the interestingness measure lift to generate association rules. J Adv Comput Sci Technol 4(1):156–162
14. Lago RM et al (2007) Diabetes and hypertension. Nat Clin Pract Endocrinol Metab 3(10):667. Gale OneFile: Health and Medicine, link.gale.com/apps/doc/A188734101/HRCA?u=anon~2c408167&sid=googleScholar&xid=9ecd7981. Accessed 7 Dec 2023

A Lightweight Mutual Authentication and Privacy-Preserving Protocol for Secure Message Transmission in Vehicular Ad-Hoc Networks

Ram Prakash, Vinod Kumar, and Khushboo Gupta

Abstract Vehicular Ad-Hoc Networks (VANETs) play a crucial role in improving traffic efficiency and road safety through the exchange of critical information among vehicles. Recently, many researchers have proposed various privacy-preserving authentication protocols for secure communication in VANETs. However, most of the protocols suffer from un-linkability, higher computation, and communication overheads. To solve the aforementioned issues in VANETs, we present a lightweight mutual authentication and privacy-preserving protocol that significantly reduces computation and communication overheads of Trusted Authority (TA) and vehicles. Security verification demonstrates that our proposed protocol is secure against various prevalent cryptographic attacks. Moreover, the performance analysis illustrates the efficiency of our proposed work in comparison with existing works. Effectiveness of our proposed work makes it suitable for secure message transmission among vehicles in VANETs.

Keywords Privacy-preserving · Authentication · Security · Un-linkability · VANETs

1 Introduction

Extensive research conducted in the area of secure transportation has made it abundantly evident that inadequate security provisions for road safety can result in numerous disasters. Consequently, in order to prevent such unfortunate events, Intelligent Transportation Systems (ITS) use a secure network technology called VANET, which provides real-time warning to a wide range of vehicles in urban areas regarding traffic congestion, pedestrian crossings, road conditions, weather, emergency vehicles, traffic violations, and road intersections. VANET environments comprise three components: On-Board Units (OBUs), Road Side Units (RSUs), Trusted Authorities

R. Prakash · V. Kumar (✉) · K. Gupta
Department of Electronics and Communication, University of Allahabad, Prayagraj, India
e-mail: vinodmtech2010@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_46

(TAs) and an online wireless communication system based on 802.11p. The TA is a central authority that has excessive computation and storage than vehicles, and it maintains the whole VANETs. Ensuring secure communication in VANETs is imperative to thwart spiteful activities and protect the confidential messages exchanged among vehicles. However, the open and dynamic nature of VANETs introduces significant security and privacy concerns. VANETs model improves traffic efficiency in present and futures.

In VANETs, all vehicles or RSU messages are transmitted or received through and vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications. Every vehicle is equipped with an OBU and uses a Dedicated Short-Range Communication (DSRC) system, DSRC technology makes use of wireless standards IEEE 802.11p [1, 2]. The first step in guaranteeing secure message transmission in a VANET environment is authentication. Authentication, a crucial process, involves validating a user's identity before allowing access to the network. In the transportation system, privacy-preserving vehicle identity and unconditional tracking are features maintained for secure message transmission. In this work, we introduce a lightweight mutual authentication and privacy-preserving protocol which ensures that only the verified vehicles can communicate with each other via RSU.

The rest of the paper is organized as follows: review of related protocols is presented in the Sect. 2. Section 3 describes the phases of our proposed approach. Security analysis is described in Sect. 4. Performance analysis is described in Sect. 5. In Sect. 6, a brief conclusion is presented.

2 Related Work

To address the security issues in VANETs and other related fields, several authentication protocols [1–25] have been presented by the numerous researchers. In 2007, Raya et al. [15] introduced an authentication process based on PKI. Their approach required the pre-installation of numerous incognito certifications and the associated private–public keys in the vehicle, leading to a significant burden in certification management for the Trusted Authority. He et al. [3] and Vijay Kumar et al. [11] used the master secret key, which is produced by the Trusted Authority, stored in the Tamper Proof Device (TPD), operating under the assumption that it remained compromised by any malicious attacker. Nevertheless, an attacker could exploit a side channel attack to gain access to information stored in the TPD. Wang et al. [16, 17] addressed a novel authentication protocol using complex certificate exchange. However, the protocol required the transmission of certain RSU master keys to the vehicles. For this reason, a system can collapse and break the integrity of the message, suffered from un-linkability and various attacks.

In 2018, Islam et al. [8] have introduced an innovative vehicle authentication method using a new approach based on passwords and a novel group key among vehicle. Furthermore, the author Cui et al. [9] proposed a streamlined method built upon the foundation laid by [8]. Later, it was observed that schemes [8, 9] suffered

from un-linkability and various attacks. In 2020, Li et al. [10] pointed out and addressed the problems presented in [8, 9]. In 2018, Wu et al. [22] addressed an innovative authentication protocol using session key, for wireless sensors medical system. Later, it is found that the scheme had some vulnerability such as password guessing attacks and some security keys disclosed at the sensor networks. Zhang et al. [23] and Hu Xiong et al. [24] have introduced authentication protocols based on Chinese Remainder Theorem. In the protocols, service provider's contents should only be available to vehicles in the domain. Domain management is costly due to constant vehicle join and leaving in the VANETs, thus membership in this domain requires appropriate management.

3 Proposed Protocol

The VANET system model contains the following constituents: TA, RSUs, and OBUs equipped with the vehicles. Figure 1 represents the architecture of VANET model.

Trusted Authority: TA is a strong, huge storage-capable, and not-compromising device in VANET. Before the communication starts, each vehicle and RSU has to be registered by TA and receive a secret key from TA.

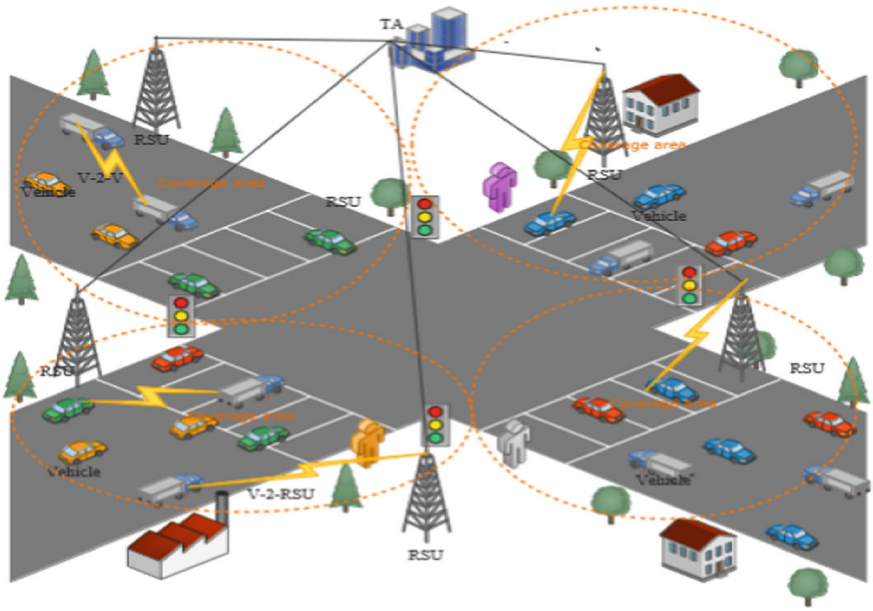


Fig. 1 Architecture of VANET model

Table 1 Definitions of notations

Notations	Descriptions
TID_{V_j}	True identity of vehicle V_j
$\mathrm{TID}_{\mathrm{RS}_i}$	True identity of RSU RS_i
σ_{TA}	TA secret key
τ_i	Shared secret key between TA and RSU
$H(.)$	One way hash function
Z_q^*	Multiplicative group using q
nv_j, nr_i	Random numbers of vehicle and RSU, respectively
Δt	Maximum network time delay
tv_j, tr_i, t_{TA}	Time stamp uses by vehicle, RSU, and TA, respectively
S_K, S_{KV}, S_{KR}	Session key calculated by TA, vehicle, and RSU, respectively

Road Side Unit: RSUs are physical components that are installed along road borders. The RSU is made up of a processor, sensors, an antenna, and transceiver. RSUs are linked to one other and the root TA.

On-Board Unit: It is a physical component attached to the vehicle for information interchanges with others. The OBU receives the data from the surroundings of the vehicle, also collecting the data from RSUs, and rapidly signing the message with a session key.

Our proposed work includes five phases: system initialization, vehicle registration, RSU registration, message authentication, and message verification as described below.

3.1 System Initialization

A large prime number q is chosen by TA, σ_{TA}, and $H(.)$ [4], where q is used to define Z_q^* and to select the required secret key and random number. $H(.)$ is a cryptographic hash function, and it is described as $H(.) : \{0, 1\}^* \rightarrow \{0, 1\}^l$, where l is the length of the output string. σ_{TA} is the secret key of system. Further, the system parameter $\Phi = \left(Z_q^*, H(.)\right)$ is shared to all vehicles and RSUs in advance (Table 1).

3.2 Vehicle Registration

Initially, TA chooses the true identity $\mathrm{TID}_{V_j} \in Z_q^*$ and pseudonym identity $\mathrm{PD}_{V_j} \in Z_q^*$ for vehicle V_j. Furthermore, it computes $a_1 = H\left(\mathrm{TID}_{V_j} \| \sigma_{TA}\right)$. Then, TA sends $\{\mathrm{TID}_{V_j}, \mathrm{PD}_{V_j}, a_1\}$ to vehicle V_j through secure channel and stores these parameters

in its database. After collecting the information from TA, vehicle V_j stores this in the local memory of OBU V_j.

3.3 RSU Registration

TA chooses a true identity $\mathrm{TID}_{\mathrm{RS}_i} \in Z_q^*$ and a shared secret key $\tau_i \in Z_q^*$ for RSU RS_i. Finally, TA sends $\{\mathrm{TID}_{\mathrm{RS}_i}, \tau_i\}$ to RS_i via secure channel and also stores in its database.

3.4 Message Generation

1. Firstly, when vehicle V_j reaches in the range of RS_i then OBU_j of V_j chooses a random number nv_j at time stamp tv_j. Further, it computes $b_1 = H\left(a_1 \| tv_j\right) \oplus nv_j$ and $b_2 = H\left(\mathrm{PD}_{V_j} \| \mathrm{TID}_{V_j} \| tv_j \| nv_j\right)$, and sends message $\mathrm{MS}_1 = \left\{b_1, b_2, tv_j, \mathrm{PD}_{V_j}\right\}$ to RS_i.
2. Further, when RS_i received message $\mathrm{MS}_1 = \left\{b_1, b_2, tv_j, \mathrm{PD}_{V_j}\right\}$ from vehicle V_j, at tv_j', RS_i checks time stamp validity $tv_j' - tv_j \leq \Delta t$. If it does not hold, the message is discarded by RS_i. Otherwise, RS_i chooses a random number nr_i at the time stamp tr_i, calculates $c_1 = H(\tau_i \| tr_i) \oplus nr_i$ and $c_2 = H\left(b_1 \| tv_j \| \tau_i \| nr_i \| \mathrm{TID}_{\mathrm{RS}_i} \| tr_i\right)$, and sends $\mathrm{MS}_2 = \{b_1, b_2, tv_j, PD_{V_j}, c_1, c_2, TID_{RS_i}, tr_i\}$ to TA.

3.5 Message Verification

1. When TA receives message MS_2 from RS_i at time tr_i', TA checks the time stamps validity, i.e., whether $tr_i' - tr_i \leq \Delta t$ and $tr_i' - tv_j \leq \Delta t$ hold. If not, the message is discarded by TA. Otherwise, it computes $nr_i = c_1 \oplus H(\tau_i \| tr_i)$ and $c_2\prime = H\left(b_1 \| tv_j \| \tau_i \| nr_i \| \mathrm{TID}_{RS_i} \| tr_i\right)$. Further, it checks whether $c_2\prime = ?c_2$ holds. If holds, then, TA searches $\{\mathrm{TID}_{V_j}, a_1\}$ according to PD_{V_j} of vehicle V_j, and computes $nv_j = b_1 \oplus H\left(a_1 \| tv_j\right)$, and $b_2\prime = H\left(\mathrm{PD}_{V_j} \| \mathrm{TID}_{V_j} \| tv_j \| nv_j\right)$ and checks whether $b_2\prime = ?b_2$ holds. If holds, TA chooses new pseudonym identity PD_{V_j}' at time stamp t_{TA} for authenticated vehicle V_j. Furthermore, TA computes a session key $S_k = H\left(\mathrm{TID}_{V_j} \| tv_j \| nv_j \| \mathrm{TID}_{\mathrm{RS}_i} \| tr_i \| nr_i\right)$, $d_1 = H(nr_i \| \tau_i \| t_{TA}) \oplus S_k$, $d_2 = H\left(nr_i \| S_k \| t_{\mathrm{TA}} \| \mathrm{TID}_{\mathrm{RS}_i}\right)$, $d_3 = H\left(nv_j \| t_{\mathrm{TA}} \| \mathrm{TID}_{V_j}\right) \oplus S_k$, $d_4 = \mathrm{PD}_{V_j}' \oplus H\left(nv_j \| \mathrm{TID}_{V_j} \| t_{\mathrm{TA}}\right)$, $d_5 = H\left(PD_{V_j}' \| \mathrm{TID}_{V_j} \| nv_j \| S_k \| t_{\mathrm{TA}}\right)$, and sends the message $\mathrm{MS}_3 = \{d_1, d_2, d_3, d_4, d_5, t_{\mathrm{TA}}\}$ to RS_i.
2. Further, when RSU receives message $\mathrm{MS}_3 = \{d_1, d_2, d_3, d_4, d_5, t_{\mathrm{TA}}\}$ from TA at time t_{TA}', RS_i checks the validity of time stamp, i.e., whether $t_{TA}' - t_{TA} \leq \Delta t$ holds. If not, MS_3 is rejected by RS_i. Otherwise, it computes $S_{\mathrm{KR}} = d_1 \oplus H(nr_i \| \tau_i \| t_{\mathrm{TA}})$

and computes $d_2 = H\left(nr_i \| S_{\mathrm{KR}} \| t_{\mathrm{TA}} \| \mathrm{TID}_{\mathrm{RS}_i}\right)$. Further, if $d_2\prime = ?d_2$ holds, RS_i sends message $\mathrm{MS}_4 = \{d_3, d_4, d_5, t_{\mathrm{TA}}\}$ to V_j.

3. Next, when vehicle V_j receives message $\mathrm{MS}_4 = \{d_3, d_4, d_5, t_{\mathrm{TA}}\}$ from RS_i at time t''_{TA}, V_j checks the time stamp validity whether $t''_{\mathrm{TA}} - t_{\mathrm{TA}} \leq \Delta t$ holds. If not, MS_4 is rejected by V_j. Otherwise, it computes $S_{\mathrm{KV}} = d_3 \oplus H\left(nv_j \| t_{\mathrm{TA}} \| \mathrm{TID}_{V_j}\right)$, $\mathrm{PD}'_{V_j} = d_4 \oplus H(nv_j \| \mathrm{TID}_{V_j} \| t_{\mathrm{TA}})$, and $d_5 = H\left(PD'_{V_j} \| \mathrm{TID}_{V_j} \| nv_j \| S_{\mathrm{KV}} \| t_{\mathrm{TA}}\right)$. If $d_5\prime = ?d_5$ holds, V_j updates PD_{V_j} with PD'_{V_j}.

After the mutual authentication, S_{KV} and S_{KR} serve as the session keys for V_j and RS_i, respectively. Each entity retains its key for a period, ensuring secure communication until the other becomes undetectable. Additionally, when two vehicles are within the detection range of the same RSU, they can initiate communication with each other via the RSU.

4 Security Analysis

To evaluate the security strength of our proposed work, security analysis is performed which shows that proposed work ensures high resistance to various known cryptographic attacks.

4.1 Vehicle Anonymity

Vehicle identification is a crucial feature of the security aspect and the vehicle's actual identity needs to be hidden from other vehicles, so that other vehicles cannot tamper with any message in the VANET. In our work, the true identity of the vehicle V_j is present in $b_1 = H\left(a_1 \| tv_j\right) \oplus nv_j$, $b_2 = H\left(\mathrm{PD}_{V_j} \| \mathrm{TID}_{V_j} \| tv_j \| nv_j\right)$, where $a_1 = H\left(\mathrm{TID}_{V_j} \| \sigma_{\mathrm{TA}}\right)$. In our work, σ_{TA} is only known by the TA and a_1 is stored in the OBU of V_j. An adversary is not able to get a_1 and σ_{TA}. Therefore, our work preserves the vehicle anonymity.

4.2 Traceability

In our work, the authentication message that vehicle V_j generates allows TA to determine the real identity of the vehicle. On collecting message $MS_2 = \left\{b_1, b_2, tv_j, \mathrm{PD}_{V_j}, c_1, c_2, \mathrm{TID}_{\mathrm{RS}_i}, tr_i\right\}$ from RS_i, TA can access the real identity of vehicle with the help of PD_{V_j}. Therefore, our scheme guarantees the traceability.

4.3 Mutual Authentication

In our proposed protocol, TA authenticates V_j and RS_i by verifying the conditions $b_{2'} = ?b_2$, and $c_{2'} = ?c_2$, respectively. Also, RS_i authenticates TA by evaluating the condition $d_{2'} = ?d_2$, and V_j authenticates TA by checking the condition $d_{5'} = ?d_5$. Thus, our proposed work provides the mutual authentication.

4.4 Un-Linkability

Un-linkability is an important aspect of security. This prevents any adversary from linking different sessions of a specific vehicle to find out the security key based on messages received through the public channel. In our proposed protocol, vehicle V_j, authentication process done by using the message $\mathrm{MS}_1 = \{b_1, b_2, tv_j, \mathrm{PD}_{V_j}\}$, where b_1, and b_2, are different in different session due to modification the value of nv_j and tv_j. Besides, pseudonym identity PD'_{V_j} updates with PD_{V_j} in each session during the message authentication. Most of parameters of MS_1 are changes. Therefore, an intruder cannot link two different messages produced by same vehicle. Thus, our proposed work guarantees un-linkability.

4.5 Resistance to Different Attacks

In this section, we describe our proposed method, which provides strong resistance to prevalent attacks in VANET.

4.5.1 Replay Attack

In this work, freshness of each message MS_1, MS_2, and MS_3 is verified by checking time stamp condition $t_i - t_j \leq \Delta t$. Thus, our proposed protocol provides resistance to replay attack.

4.5.2 Modification Attack

In this work, we check the integrity of the messages MS_1, MS_2, MS_3, and MS_4 by evaluating the $b_{2'} = ?b_2$, $c_{2'} = ?c_2$, $d_{2'} = ?d_2$, and $d_{5'} = ?d_5$. If any value is changed by adversary, change in hash values is also observed. Therefore, our proposed work ensures high resistance to modification attacks.

4.5.3 Impersonation Attack

In our approach, the authentication message is based on parameters of TA $\{a_1 = H(\mathrm{TID}_{V_j} \| \sigma_{\mathrm{TA}}), \sigma_{\mathrm{TA}}\}$ which is only accessible by the legitimate entities. Therefore, adversary cannot have secret key and a_1. Therefore, intruder cannot generate the authentication message. Hence, our proposed work guarantees the protection against impersonation attack.

4.5.4 Side Channel Attack

Many researchers have described the master secret key and these keys are stored in TPD, ensuring it remains impervious to any compromise by attackers. But attackers may be acquiring the information from TPD through side channel attacks. In our proposed protocol, pseudonym identity PD_{V_j} will be updated during each session of message authentication. Thus, our proposed protocol is free from side channel attack.

4.6 Security Features Comparison

To examine the security strength of our work, we compare the security features of recent competing authentication protocols and our protocol. The summary of security features comparison is shown in Table 2.

5 Performance Analysis

To measure the effectiveness of our work, we perform computation and communication and communication costs analysis as follows.

Table 2 Comparison of security features

Schemes	rs_1	rs_2	rs_3	rs_4	rs_5
Islam et al. [8]	✔	✔	×	×	×
Cui et al. [9]	✔	✔	×	×	×
Li et al. [10]	✔	✔	✔	✔	×
Ours	✔	✔	✔	✔	✔

Note Let rs_1, rs_2, rs_3, rs_4, and rs_5 represent preserving the vehicle identity, traceability, mutual authentication, un-linkability, and resistance to different attacks, respectively. Symbols ✔ and × respectively indicate whether the security requirement in the mentioned protocols is met or not

Table 3 Computation costs of message generation & verification phase of existing works

Schemes	Cost at V_J (ms)	Cost at RS_i (ms)	Cost at TA (ms)	Total cost (ms)
Islam et al. [8]	$5\,T_H + n_j(T_H + T_M) + 2\,n_l T_H \approx 986.7$	$(1 + n_j + n_l)T_H \approx 19.5$	$(1 + n_j)(6T_H + 2T_M + T_I) + n_l(2T_H + T_M) \approx 3968.7$	≈ 4974.9
Cui et al. [9]	$(1 + n_j + n_l)T_H \approx 19.5$	$(1 + n_j)T_H \approx 10$	$3T_H + (n_j + n_l)(T_{CRT} + 2T_M \approx 5739.5$	≈ 5769.0
Li et al. [10]	$6T_H \approx 3$	$4(1 + n_j)T_H \approx 40$	$14(1 + n_j)T_H \approx 140$	≈ 183.0
Proposed	$4T_H \approx 2$	$4(1 + n_j)T_H \approx 40$	$10(1 + n_j)T_H \approx 100$	≈ 142.0

5.1 Computation Cost Analysis

To measure the computational efficiency, we perform computation cost analysis of our work and existing schemes [8–10]. The analysis is performed using python, 4 GB RAM, intel core i5 processor, 1 TB Hard Disk, windows 10 OS. In computation cost analysis, T_M, T_I, T_{CRT}, and T_H are the execution time for multiplication, inverse, Chinese remainder, and hash operations, respectively. Their estimated values are 50.3 ms, 46.1 ms, 50.4 ms, and 0.5 ms, respectively. Besides, we consider the bit length of hash function (SHA-256), pseudonym identity, true identity, random number, and time stamp to be 32 bytes, 20 bytes, 20 bytes, and 8 bytes, respectively. Further, n_j, and n_l are the number of nodes that join and leave, respectively, in a range of RSUs. Maximum value of n_j and n_l is 19. In Table 3, we present a summary of the computation cost analysis. The analysis clearly shows that our protocol requires less execution time than the existing protocols [8–10]. Figure 2 illustrates the computation costs of our work and existing works [8–10].

5.2 Communication Cost Analysis

In this analysis, we analyze communication costs of existing schemes [8–10] and our proposed work. In Table 4, we present a summary of communication cost analysis. The analysis clearly demonstrates that our protocol requires less communication cost than the existing protocols [8–10]. Figure 3 illustrates the communication costs of our work and existing works [8–10].

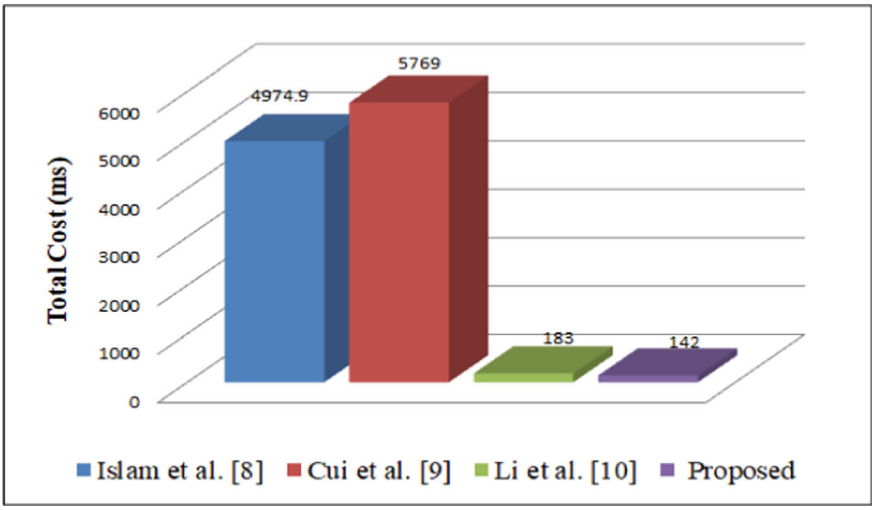


Fig. 2 Computation cost of message generation & verification phase of various schemes

Table 4 Communication costs of various schemes

Schemes	Communication cost (bytes)	Total communication cost (bytes)
Islam et al. [8]	368 $(1 + n_j + n_l)$	14,352
Cui et al. [9]	304 $(1 + n_j + n_l)$	11,856
Li et al. [10]	676 $(1 + n_j)$	13,520
Ours	552 $(1 + n_j)$	11,040

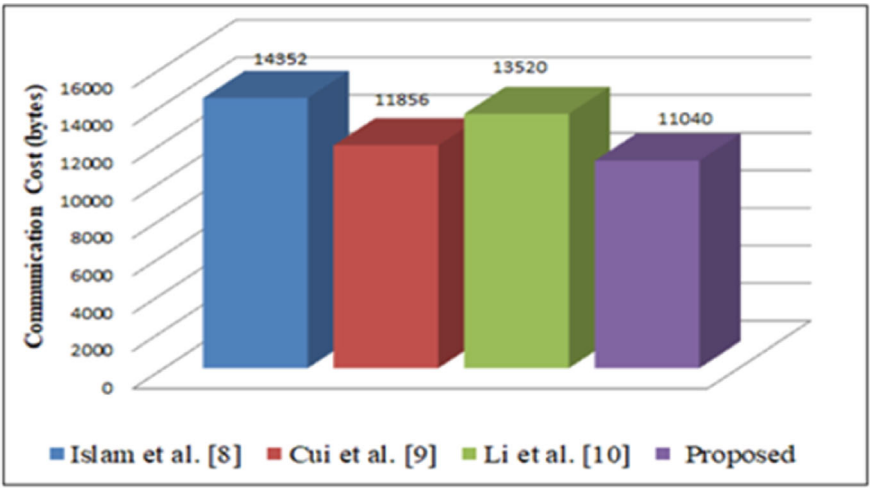


Fig. 3 Communication cost of various schemes

6 Conclusion

This work proposes an effective privacy-preserving mutual authentication protocol for VANETs that minimizes computational and communicational costs of TA, RSU, and vehicles. Our protocol provides resistance to various attacks as explained in the security analysis. The performance analysis shows the efficiency of our proposed work. Therefore, proposed work is well suited for the VANET environment since it provides strong security and effective performance.

References

1. Mejri MN, Ben-Othman J, Hamdi M (2014) Survey on VANET security challenges and possible cryptographic solutions. Veh Commun 1(2):53–66
2. Sahu P, Kumar V (2023) Design and performance analysis of a lightweight authentication protocol for vehicular Ad-Hoc networks. In: 2023 International conference on IoT, communication and automation technology (ICICAT). Gorakhpur, India, p 16
3. He D, Zeadally S, Xu B, Huang X (2015) An efficient identity-based conditional privacy-preserving authentication scheme for vehicular Ad Hoc networks. IEEE Trans Inf Forensics Secur 10(12):2681–2691
4. Stallings W (2010) Cryptography and network security: principles and practice, 5th edn. Prentice Hall Press, Upper Saddle River, NJ, USA
5. Kumar V, Kumar R, Pandey SK (2020) A secure and robust group key distribution and authentication protocol with efficient rekey mechanism for dynamic access control in secure group communications. Int J Commun Syst 33(14):1–22
6. Kumar V, Kumar R, Pandey SK, Alam M (2018) Fully homomorphic encryption scheme with probabilistic encryption based on Euler's theorem and application in cloud computing. In: Aggarwal VB, Bhatnagar V, Mishra DK (eds) Big data analytics. AISC, vol. 654. Springer, Singapore, pp 605–611.
7. Kumar V, Kumar R, Pandey SK (2017) Enhanced and secured RSA public key cryptosystem algorithm using Chinese remainder theorem. Third International conference, NGCT 2017, (CCIS), pp 1–12
8. Islam SKH, Obaidat MS, Vijayakumar P et al (2018) A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs. Future Gener Comput Syst 84:216–227
9. Cui J, Tao X, Zhang J, Xu Y, Zhong H (2018) HCPA-GKA: a hash function-based conditional privacy-preserving authentication and group-key agreement scheme for VANETs. Veh Commun 14:15–25
10. Li X, Liu T, Obaidat MS, Wu F, Vijayakumar P, Kumar N (2020) A lightweight privacy-preserving authentication protocol for VANETs. IEEE Syst J 14(3):3547–3557. https://doi.org/10.1109/JSYST.2020.2991168
11. Vijayakumar P, Azees M, Kannan A, Jegatha Deborah L (2016) Dual authentication and key management techniques for secure data transmission in vehicular Ad Hoc networks. IEEE Trans Intell Transp Syst 17(4):1015–1028
12. Kumar V, Kumar R, Pandey SK (2020) A computationally efficient centralized group key distribution protocol for secure multicast communications based upon RSA public key cryptosystem. J King Saud Univ-Comput Inf Sci 32(9):1081–1094
13. Kumar V, Kumar R, Pandey SK (2020) Polynomial based non-interactive session key computation protocol for secure communication in dynamic groups. Int J Inf Technol 12:283–288. https://doi.org/10.1007/s41870-018-0140-1

14. Gupta K, Kumar V (2023) KMS–AMI: an efficient and scalable key management scheme for secure two-way communications in advanced metering infrastructure of smart grid. J Supercomput. https://doi.org/10.1007/s11227-023-05756-5
15. Raya M, Hubaux J-P (2007) Securing vehicular ad hoc networks. J Comput Secur 15(1):39–68. http://dl.acm.org/citation.cfm?id=1370616.1370618
16. Wang S, Yao N (2017) Liap: a local identity-based anonymous message authentication protocol in VANETs. Comput Commun 112:154–164
17. Lo N-W, Tsai J-L (2016) An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. IEEE Trans Intell Transp Syst 17(5):1319–1328. https://doi.org/10.1109/TITS.2015.2502322
18. Kumar V, Kumar R, Pandey SK (2021) LKM-AMI: a lightweight key management scheme for secure two-way communications between smart meters and HAN devices of AMI system in smart grid. Peer-to-Peer Netw Appl 14:82–100. https://doi.org/10.1007/s12083-020-00921-6
19. Kumar V, Kumar R, Pandey SK (2021) A Computationally efficient and scalable key management scheme for access control of media delivery in digital Pay-TV systems. Multimedia Tools Appl 80:1–34
20. Xiong W, Wang R, Wang Y, Zhou F, Luo X (2021) CPPA-D: efficient conditional privacy-preserving authentication scheme with double-insurance in VANETs. IEEE Trans Veh Technol 70(4):3456–3468
21. Gupta K, Kumar V (2023) A provably secure ECC-based key distribution scheme for secure message transmission in vehicular Ad Hoc networks. In: Noor A, Saroha K, Pricop E, Sen A, Trivedi G (eds) Proceedings of third emerging trends and technologies on intelligent systems. ETTIS 2023. Lecture notes in networks and systems, vol 730. Springer, Singapore
22. Wu F, Li X, Sangaiah AK, Xu L, Kumari S, Wu L, Shen J (2018) A lightweight and robust factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. Future Gener Comput Syst 82
23. Zhang J, Cui J, Zhong H, Chen Z, Liu L (2021) PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular Ad-Hoc networks. IEEE Trans Dependable Secure Comput 18(2):722–735. https://doi.org/10.1109/TDSC.2019.2904274
24. Xiong H, Chen J, Mei Q, Zhao Y (2022) Conditional privacy-preserving authentication protocol with dynamic membership updating for VANETs. IEEE Trans Dependable Secure Comput 19(03):2089–2104
25. Gupta K, Kumar V (2023) Security and performance analysis of advanced metering infrastructure in smart grid and use of blockchain in security perspective. In: Cyber security using modern technologies. CRC Press, pp 73–90

Siamese BERT with Enhanced Feature Models for Paraphrase Detection

M. Mumthaz Beegum and S. Aji

Abstract Paraphrase detection, an essential aspect of natural language processing, involves assessing whether two provided sentences express equivalent meanings. Recent studies addressing the classification of sentence pairs for paraphrase detection commonly employ an approach that involves concatenating the embeddings of the sentences, which are then utilized as input for the classifier. This work enhances the effectiveness of paraphrase detection models by systematically and logically preparing the inputs to capture relevant features better. The effectiveness of the proposed method is confirmed through validation using two publicly available datasets, MRPC and QQP. The experimental results demonstrate superior quantitative measures, including precision, recall, and F1-measure, for the proposed approaches compared to baseline methods.

Keywords Paraphrase detection · BERT · Cosine weighted · SVM · Random forest

1 Introduction

The importance of natural language processing (NLP) applications has been highlighted in recent years due to the rapid growth of digital content. Paraphrase detection is an essential research area in NLP, playing a vital role in preserving the integrity of written content across different domains. Recognizing texts with similar semantic content, even in the presence of substantial differences in word selection and structure, is crucial for a range of applications, including the detection of plagiarism [1]. The term 'paraphrase' pertains to sentences that convey identical meanings as other sentences but employ different wording [2]. Paraphrase identification involves a two-class classification task where the goal is to determine whether two provided texts, S1

M. Mumthaz Beegum (✉) · S. Aji
Department of Computer Science, University of Kerala, Thiruvananthapuram, Kerala, India
e-mail: mumthazbeegumm@keralauniversity.ac.in

S. Aji
e-mail: aji@keralauniversity.ac.in
© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_47

and S2, convey the same meaning or not. Paraphrasing exhibits distinct categories based on the extent of alterations within the texts [3]. Low paraphrasing involves the substitution of elements such as synonyms, hypernyms, hyponyms, meronyms, and holonyms. In contrast, high paraphrasing encompasses the realization of low paraphrase phenomena, along with additional modifications in morphology, lexicon, semantics, syntax, and discourse. The increasing prevalence of large language models (LLMs) like GPT-4 [4] and ChatGPT has facilitated the automated creation of high-quality paraphrases, highlighting the crucial need for robust detection methods.

Conventional methods for paraphrase identification often depended on manually crafted features and shallow representations, limiting their ability to capture the intricate nuances of language. Deep learning has ushered in the emergence of embedding models, which serve as effective tools for acquiring intricate representations of textual data. The introduction of the transformer architecture [5] marked a transformative era in NLP, initiating the era of pre-trained large language models. These models, through pre-training, acquire universal language representations that can be fine-tuned for specific tasks, eliminating the necessity of training each model from the ground up [6]. Among these models, bidirectional encoder representations from transformers (BERTs) [7] have demonstrated remarkable success in capturing contextual information, enabling it to encode complex semantic relationships within text. Since the unveiling of the BERT model, several methodologies have been developed to enhance its performance. These range from modifying the pre-training stage [8] to adjusting attention mechanisms [9], implementing knowledge distillation [10], and exploring other intricate approaches.

BERT and RoBERTa have established a new benchmark in performance for tasks related to sentence classification and sentence-pair regression. Sentence-BERT (SBERT) is a modified version of the pre-trained BERT network. It integrates siamese and triplet network structures to produce sentence embeddings that capture richer semantic meanings. This modification significantly decreases the time required to identify the most similar pair, reducing the time required from 65 h with BERT to approximately 5 s with SBERT [11]when applied to semantic textual similarity tasks. Remarkably, this acceleration in processing speed is achieved without compromising the accuracy achieved by BERT. In this work, the authors introduce three distinct approaches to optimize various objectives, encompassing classification, regression, and triplet methodologies. The goal of the classification task is to precisely assign a provided pair of sentences to one of several predefined classes. The classifier operates on a concatenated input embedding to execute the classification task.

In order to address the task of classifying sentence pairs for paraphrase identification, the recent studies [11, 12] demonstrate a common approach where the embeddings of the sentences are concatenated and provided as input to the classifier. However, treating these sentence-pair embeddings as a singular entity in the classifier neglects the individual meanings embedded in each sentence. Recognizing the meaningful representations of both sentences, we propose novel approaches for input combinations that leverage the distinct information encoded in each sentence's embedding.

The proposed approaches for paraphrase identification tasks are evaluated on two prominent datasets: the Microsoft Research Paraphrase Corpus (MSRP) and the Quora Question Pairs (QQP) dataset. Our methods have demonstrated comparable outcomes when compared to the cutting-edge techniques in the field.

1.1 Research Objective

The purpose of this study is to improve the classification accuracy of sentence pairs for paraphrase identification by developing innovative approaches that consider the distinct information encoded in the embeddings of individual sentences. Unlike current methodologies that concatenate sentence embeddings as a single input to the classifier, our goal is to recognize and exploit the unique semantic representations inherent in each sentence. This work aims to enhance the performance of paraphrase detection models by preparing the inputs logically and systematically to capture the features for generalizing the process to a large extent.

The article's subsequent sections are arranged as follows: Sect. 2 conducts an in-depth exploration of prior BERT-based methodologies employed in paraphrase identification tasks. The architecture that is suggested in this study is introduced in Sect. 3. Section 4 provides comprehensive insights into the implementation, outcomes, and discussions. Finally, Sect. 5 encapsulates the conclusions derived from our research endeavors.

2 Related Works

The invention of the transformer architecture [5], a groundbreaking development in the field of NLP. Before the transformer's introduction, sequential models faced challenges with parallelization and capturing long-range dependencies But transformer architecture addressed these limitations by introducing a novel attention mechanism that allows for parallel computation of input sequences.

The BERT model [7], built upon the transformer architecture through a cross-encoder approach, stands out as the most renowned language model. It has achieved SOTA results across a diverse range of tasks. Two versions of the model are accessible: the base version containing 12 transformer encoder blocks, and the large version comprising 24 transformer encoder blocks. Following the inception of the BERT model, the NLP community introduced various methodologies aimed at enhancing the efficacy of large language models constructed upon the transformer architecture. These strategies commonly adopt a two-stage scheme, which includes pre-training and fine-tuning phases. The pre-training process incorporates a Masked Language Model and Next Sentence Prediction techniques.

The Sentence-CROBI [12] architecture comprehensively integrates cross-encoders and bi-encoders to represent sentence pairs. The bi-encoder element in

this method is grounded on the SBERT model. A modified version of the BERT model, implemented via a Siamese neural network, was employed in the study. This modification allows for extracting individual, fixed-size vectors from each text. The BERT model's last hidden state undergoes a pooling operation to generate sentence vectors for each text, labeled as 'x' and 'y'. In the cross-encoder component, the RoBERTa model is utilized to process the joint encoding of both sentences. The final sequence representation is achieved using the classification token [CLS]. The classifier receives input that comprises the concatenation of the [CLS] token, sentence vectors x and y, along with the Euclidean distance between the sentence vectors.

The ALBERT [13] model aims to reduce model size through two main strategies. The first employs factorized embedding parameterization by decomposing the large vocabulary embedding matrix into smaller matrices, facilitating the growth of hidden size without a substantial increase in parameter size. The second step involves sharing parameters across different layers to control the increase in the model's depth.

The StructBERT [14] model introduces two tasks to learn language structure at both word and sentence levels. StructBERT enhances the masked LM task by rearranging tokens post-word masking to predict the correct sequence. Additionally, for improved sentence comprehension, StructBERT randomly exchanges sentence order and predicts both the subsequent and preceding sentences as part of a new sentence prediction task.

The DeBERTa [15] model introduces modifications to the transformer architecture by proposing a new attention mechanism. This mechanism involves encoding words with two vectors: one dedicated to the word itself and another to its relative location. In contrast to the original transformer, which combines content and position vectors through summation, DeBERTa independently calculates attention weights for both representations. Moreover, the model integrates absolute position data for the Masked Language Model task. This enables the model to consider not only the content of the word but also its relative and absolute positions when predicting the value of a masked token.

The Siamese BERT [16] model introduces the Siamese Bert Architecture Model, which incorporates an attention mechanism. This architecture combines Siamese network structures with bidirectional LSTM units to capture features of shared weights. Additionally, it utilizes BERT to extract text features. The attention mechanism is then applied to enhance the semantic representation of textual vectors.

3 Proposed Methodology

This section describes various feature models employed for the paraphrase identification task and outlines the steps undertaken to train and assess the model. Figure 1 depicts various input combination strategies.

In this study, we employed the Siamese neural network structure of the BERT model to compute the embeddings for both sentences. This network can derive distinct, standardized vectors from each textual input, ensuring a consistent vector size.

We generate sentence vectors for each text by applying a pooling operation to the final hidden state of the BERT model. The resulting sentence vectors are denoted as 'm' and 'n', respectively. The various feature models employed in this work are normal concatenation (NC), cosine weighted (CW), PCA projection (PCAP), and cosine-weighted mean vector (CWMV). In normal concatenation approach, the embedding vectors m and n are concatenated and then fed to the classifiers. In cosine-weighted approach, we begin by computing the cosine similarity between the sentence embeddings m and n using Eq. 1. Subsequently, we multiply this similarity by the sum of the embedding vectors m and n.

$$\text{Cosine Similarity}(\mathbf{m}, \mathbf{n}) = \frac{\mathbf{m} \cdot \mathbf{n}}{\|\mathbf{m}\| \cdot \|\mathbf{n}\|} \tag{1}$$

In PCA projection, we perform principal component analysis (PCA) on the embedding vectors. The resulting PCA embeddings are multiplied by the cosine similarity computed from the embedding vectors m and n. Finally, cosine weighted with mean vector approach is similar to cosine-weighted method. Here we enhance the cosine-weighted approach by multiplying the cosine similarity with the average of the embedding vectors m and n.

In this study, random forest (RF) and support vector machine (SVM) are the two classifiers employed. RF [17] finds extensive application in solving classification problems owing to its remarkable predictive accuracy. Additionally, the model offers valuable insights into the significance of variables, enhancing its utility for classification tasks. The technique is based on a large ensemble of decision trees, where each tree produces a classification result. The total votes cast by all of the forest's trees decide the ultimate categorization.

SVM [18] is a flexible machine learning tool suitable for both regression and classification. However, it excels particularly in solving classification problems. It classifies linear data by utilizing a maximum margin hyperplane (MMH), with the objective of maximizing the distance between data points, referred to as support vectors. The positive and negative hyperplanes, forming two parallel lines, effectively segregate the data. When dealing with nonlinear data, a kernel function is utilized to create a multi-dimensional hyperplane for the purpose of classification.

To ensure a balanced training process, we conduct SVM and RF training on all datasets once for each feature models. We use fivefold cross-validation in a tiny randomized grid search to find the right parameters for the SVM and RF that are specific to each approach. We conduct a grid search four times, once for each method, and select the trained SVMs and RFs with the highest test scores across datasets for further analysis. The grid search space parameters and values for random forest are displayed in Table 1. We have explored the SVM grid search space with various parameters, such as kernel, C, and Gamma. Kernel options include rbf, sigmoid, poly, and linear. The parameter C is explored with values of 1.0, 10, and 100, while the gamma parameter ranges from 0.0001 to 1.0, with additional options for 'scale' and 'auto'.

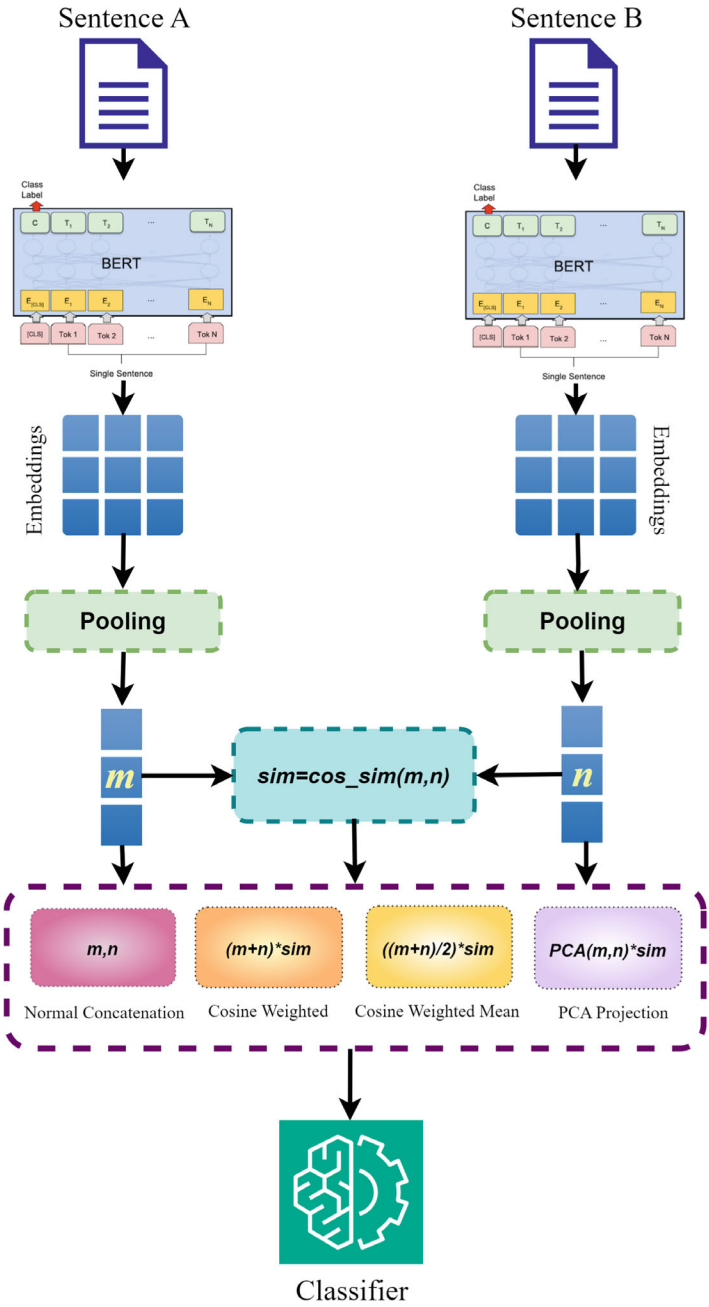


Fig. 1 Proposed feature models

Table 1 Grid search space for random forest

Parameter	Ranges
n_estimators	start = 200, stop = 2000, num = 10
max_features	auto, sqrt, log2
max_depth	10 integers, evenly spaced between 10 and 1000
min_samples_split	2, 5, 10, 14
min_samples_leaf	1, 2, 4, 6, 8

Table 2 Statistics of MRPC and QQP corpora

Corpus	Paraphrased instances	Non-paraphrased instances	Total instances
MRPC (train)	2753	1323	4076
MRPC (test)	1147	578	1725
QQP (train)	134,623	229,223	363,846
QQP (val)	14,959	25,471	40,430

4 Results and Discussion

To evaluate the effectiveness of the proposed feature models, tests were conducted on the MRPC [20] and QQP [21] datasets. Statistical details for these datasets are presented in Table 2.

Over a span of two years, a total of 5801 phrase pairs were gathered from diverse news sources. The pairs were manually classified into two distinct groups: those that are paraphrased and those that aren't. The corpus was then segmented into 4076 sections for the training process and 1323 for the testing process.

The QQP corpus consists of 795,241 pairs of questions, which are labeled as either Duplicate or Not Duplicate. This corpus is split into three parts: a training set with 363,846 pairs of questions, a validation set with 40,430 pairs, and a testing set with 390,965 pairs.

4.1 Evaluation Measures

The proposed feature models' performance was evaluated using precision, recall, and F1-score metrics. Precision, representing the proportion of accurately predicted paraphrases among the total predicted paraphrases, functions as an indicator of the precision of positive predictions.

$$\text{Precision} = \frac{\text{Number of correctly predicted paraphrases}}{\text{Total number of predicted paraphrases}} \tag{2}$$

Recall assesses the model's capability to recognize all relevant instances, representing the ratio of accurately predicted paraphrases to the overall number of actual paraphrases.

$$\text{Recall} = \frac{\text{Number of correctly predicted paraphrases}}{\text{Total number of actual paraphrases}} \tag{3}$$

The F1-score, which computes the harmonic mean of precision and recall, offers a balanced evaluation by taking into account both false positives and false negatives.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

4.2 Result Analysis

Table 3 shows the results of various approaches on the QQP dataset, and each approach is evaluated in terms of precision (P), recall (R), and F1-score (F1). The normal concatenation approach demonstrated a balanced performance with moderate precision, recall, and F1-score. Contrastingly, the cosine-weighted approach showcased a significant enhancement in performance, emphasizing the important role of cosine similarity in feature models. The PCA Projection approach demonstrates a decrease in performance compared to the other approaches, with lower precision, recall, and F1-score. The cosine-weighted mean vector method demonstrates superior performance when compared to the normal and PCA projection approaches. The cosine weighted and cosine-weighted mean vector approach with SVM grid search achieved the highest F1-score for QQP dataset. The SVM grid search employing the cosine-weighted approach yielded optimal parameter values of 10 for C, 10 for gamma, and rbf for the kernel. On the other hand, the cosine-weighted mean vector approach resulted in optimal parameter values of 1.0 for C, scale for gamma, and rbf for the kernel. Figure 2 shows the receiver operating characteristic (ROC) curve and precision recall (PR) curve of cosine weighted and cosine-weighted mean vector approaches of SVM grid space.

Table 4 presents the outcomes of various approaches on the MRPC dataset. The table shows that the cosine weighted and cosine-weighted mean vector approaches attained the highest F1-scores for RF and RF grid search space. PCA projection results in a significant drop in performance with markedly lower F1-scores. In the normal approach, both SVM and RF demonstrate robust performance with high F1-scores around 0.94. The RF grid search, utilizing a cosine-weighted approach, produced optimal parameter values: criterion set to entropy, max_depth at 120, max_features set to sqrt, min_samples_leaf at 3, min_samples_split at 4, and n_estimators at 1200. Meanwhile, employing the cosine-weighted mean vector approach, the optimal parameter values were as follows: criterion set to entropy, max_depth at

Table 3 Result on QQP corpora

Approaches	SVM			SVM Grid			RF			RF Grid		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Normal concatenation	0.64	0.63	0.63	0.70	0.69	0.69	0.66	0.66	0.66	0.69	0.69	0.69
Cosine weighted	0.78	0.78	0.78	0.84	0.84	0.83	0.83	0.81	0.80	0.83	0.82	0.82
PCA Projection	0.66	0.63	0.61	0.79	0.79	0.78	0.79	0.78	0.78	0.80	0.79	0.79
Cosine-weighted mean vector	0.79	0.79	0.79	0.84	0.83	0.83	0.83	0.81	0.80	0.83	0.82	0.82

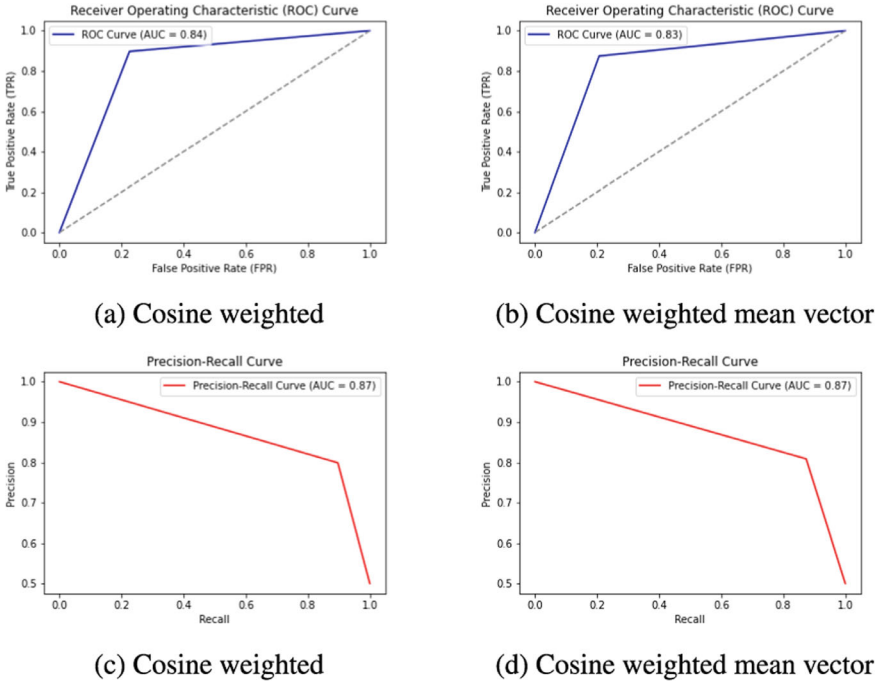


Fig. 2 ROC and precision recall curve of cosine weighted and cosine-weighted mean vector approaches using SVM grid

1000, max_features set to sqrt, min_samples_leaf at 4, min_samples_split at 2, and n_estimators.8 at 100. Figure 3 shows the ROC curve and PR curve of cosine weighted and cosine-weighted mean vector approaches of RF grid space.

Table 4 Result on MRPC corpora

Approaches	SVM			SVM grid			RF			RF grid		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Normal concatenation	0.73	0.73	0.71	0.94	0.94	0.94	0.94	0.94	0.93	0.94	0.94	0.94
Cosine weighted	0.73	0.74	0.74	0.89	0.89	0.89	0.95	0.95	0.95	0.95	0.95	0.95
PCA projection	0.38	0.61	0.47	0.69	0.70	0.68	0.69	0.70	0.67	0.69	0.70	0.67
Cosine-weighted mean vector	0.72	0.72	0.72	0.84	0.84	0.84	0.95	0.95	0.95	0.95	0.95	0.95

The results on both datasets reveals that the cosine weighted and cosine-weighted mean vector approaches consistently outperform other methods in terms of the F1-score. When two sentences share similar meanings, their embeddings are expected to be closer in the vector space. Adding two sentence embeddings and multiplying with cosine similarity allow the model to aggregate information from both sentences. Multiplying the cosine similarity with the combined embeddings further emphasizes the similarity strength. It gives more weight to similar components in the embeddings, effectively focusing on the shared semantic information. Utilizing the average of sentence embeddings aids in capturing the comprehensive semantic content encapsulated within the sentences. Averaging can be more robust to noise or irrelevant details in individual sentences. If one sentence has outliers or noisy information, averaging can help mitigate the impact of such noise. The PCA projection method yields the lowest outcome in both datasets when compared to other approaches. The goal of PCA is to capture the maximum variance in the data with the first few principal components. The variance quantifies the degree to which the values within a dataset are dispersed. When applying PCA to a dissimilar sentence embedding, it typically results in less effective dimensionality reduction and it may not effectively represent meaningful information. The variance in the data is likely to be distributed across the principal components.

4.3 Comparison with Baseline Approaches

Table 5 represents the outcomes of both the SOTA method from the GLUE Benchmark leaderboard and the proposed approach on the QQP and MSRP datasets. The results of our approaches are highlighted in bold to emphasize the key findings. In Ortiz-Barajas et al. [12] investigation, both cross-encoder and bi-encoder methodologies were employed. The cross-encoder utilized the Siamese BERT architecture,

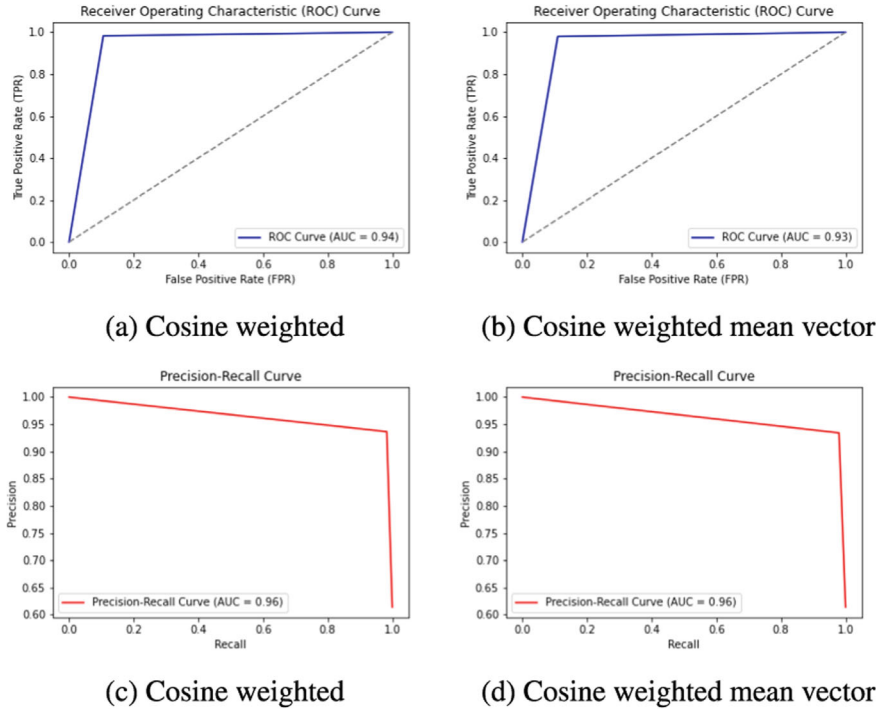


Fig. 3 ROC and precision recall curve of cosine weighted and cosine-weighted mean vector approaches using RF grid

Table 5 Model performance comparison

Model	QQP		MRPC	
	F1-Score	Changes	F1-Score	Changes
Sentence-CROBI [12]	73.80	−9.20	93.35	−1.65
ALBERT [13]	74.20	−8.80	93.40	−1.60
StructBERT [14]	74.40	−8.60	93.60	−1.40
BORT [19]	66.00	−17.0	94.10	−0.90
Siamese BERT [16]	78.65	−4.35	84.49	10.51
Cosine weighted	**83.00**	–	**95.00**	–

while the bi-encoder employed RoBERTa. They obtained F1-scores of 73.80 and 93.35 for the QQP and MRPC datasets, respectively.

Lan et al. [13] addressed the challenges of increasing model size and training time. To mitigate these issues, ALBERT introduces two parameter reduction techniques. They achieved F1-scores of 74.20 for the QQP dataset and 94.40 for the MRPC dataset. Wang et al. [14] enhanced BERT pre-training by introducing two innovative linearization strategies, incorporating structural information such as word-level

ordering and sentence-level ordering into the traditional masking strategy. Their experimental efforts secured F1-scores of 74.40 on the QQP dataset and 93.60 on the MRPC dataset. de Wynter and Perry et al. [19], employing a fully polynomial time approximation scheme as its model reduction approach, attained F1-scores of 66.00 for the QQP dataset and 94.10 for the MRPC dataset. Li, Ruihao, et al. [16] utilized a Siamese BERT network model with bidirectional LSTM, incorporating an attention mechanism to obtain advanced semantic features and acquired F1-scores of 78.65 for the QQP dataset and 84.49 for the MRPC dataset.

5 Conclusion

This work focuses on paraphrase detection, aiming to determine the equivalence of meaning in provided sentence pairs. The common practice in recent studies involves concatenating sentence embeddings and utilizing them as input for classifiers. The proposed approaches significantly improve paraphrase detection models' efficiency by systematically and logically preparing inputs to capture relevant features better. The validation on two publicly available datasets, MRPC and QQP, demonstrates the superiority of the proposed approaches over baseline methods, as evidenced by improved quantitative measures.

Acknowledgements We would like to thank the University of Kerala for providing the facilities for doing this research under the University JRF scheme.

References

1. Foltýnek T, Ruas T, Scharpf P, Meuschke N, Schubotz M, Grosky W, Gipp B (2020) Detecting machine-obfuscated plagiarism. In International conference on information (pp 816-827). Cham: Springer International Publishing
2. Bhagat R, Hovy E (2013) What is a paraphrase? Comput Linguist 39(3):463–472
3. Mota Montoya MA, Da Cunha I, Lopez-Escobedo F (2016) Un corpus de paráfrasis en español: metodología, elaboración y análisis. RLA. Revista de lingüística teórica y aplicada 54(2):85-112
4. OpenAI R (2023) Gpt-4 technical report. arxiv 2303.08774. View in Article, 2, 13
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, ... Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems, 30
6. Sun K, Luo X, Luo MY (2022) A survey of pretrained language models. In: international conference on knowledge science, engineering and management (pp 442-456). Cham: Springer International Publishing
7. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
8. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, ... Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
9. He P, Liu X, Gao J, Chen W (2020) Deberta: decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654

10. de Wynter A, Perry DJ (2020) Optimal subarchitecture extraction for BERT. arXiv preprint arXiv:2010.10499
11. Reimers N, Gurevych I (2019) Sentence-bert: sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084
12. Ortiz-Barajas JG, Bel-Enguix G, Gómez-Adorno H (2022) Sentence-CROBI: a simple cross-bi-encoder-based neural network architecture for paraphrase identification. Mathematics 10(19):3578
13. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: a lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942
14. Wang W, Bi B, Yan M, Wu C, Bao Z, Xia J, Peng L, Si L (2019) Structbert: Incorporating language structures into pre-training for deep language understanding. arXiv preprint arXiv:1908.04577
15. He P, Liu X, Gao J, Chen W (2020). Deberta: decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654
16. Li R, Cheng L, Wang D, Tan J (2023) Siamese bert architecture model with attention mechanism for textual semantic similarity. Multimed Tools Appl 82(30):46673–46694
17. Breiman L (2001) Random forests. Mach Learn 45:5–32
18. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
19. de Wynter A, Perry DJ (2020) Optimal subarchitecture extraction for BERT. arXiv preprint arXiv:2010.10499
20. Dolan B, Brockett C (2005) Automatically constructing a corpus of sentential paraphrases. In: Third international workshop on paraphrasing (IWP2005)
21. Chen Z, Zhang H, Zhang X, Zhao L (2018) Quora question pairs. https://data.quora.com/First-QuoraDataset-Release-Question-Pairs

SGPT: An Improved Sentiment Classifier Using Generative Pretrained Transformer

Yamuna Prasad, Nitin, and Marc Cahay

Abstract Sentiment analysis has numerous applications across various domains such as Social Media Monitoring, Customer Feedback Analysis, Market Research, Financial Markets, Product Reviews, Political Analysis, Employee Feedback, Healthcare, Tourism, and Education. Industries use sentiment analysis to develop informing policy and product design for wider acceptance and improved revenue. Classification methods, from keyword distribution to advanced models like Recurrent Neural Networks and Transformers, have been employed to categorize sentiments into Positive, Negative, and Neutral. Transformer-based models like OpenAI's chatGPT (a chatbot using Generative Pretrained Transformer (GPT)-based Large Language Model) and Google's BARD (a chatbot based on Bidirectional Encoder Representations from Transformers (BERT) framework) struggle with local context and bias issues. In this paper, we combine Latent Semantic Analysis (LSA) with Continuous Bag of Words (CBOW), GPT and BERT features to predict the sentiments, addressing local context to enhance the model. We name our proposed method as SGPT. The proposed SGPT method demonstrates significant improvement over state-of-the-art sentiment classification approaches on benchmark finance sentiment datasets.

Keywords Sentiment · Generative pretrained transformer · Bard · Latent semantic analysis

Y. Prasad (✉)
Indian Institute of Technology Jammu, Jammu, India
e-mail: yamuna.prasad@iitjammu.ac.in

Nitin · M. Cahay
University of Cincinnati, Cincinnati, USA
e-mail: nitinfu@ucmail.uc.edu

M. Cahay
e-mail: cahaymm@ucmail.uc.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_48

1 Introduction

Sentiment analysis helps to interpret the opinions or polarity embedded in textual data. It is very important to understand the feeling of the people about products, services, or specific topics are crucial for businesses, researchers, and policymakers [1–3]. Sentiment analysis plays important role in understanding opinions and polarity in various applications involving Social Media, Business, Finance, Education, Healthcare, and Politics [4, 5]. In social media domain, it is employed to track brand mentions, understand customer perception and manage reputation. The businesses use the customer feedback to improve products and services [6, 7]. In market research, sentiment analysis helps in understanding the consumer preferences and market trends. In the finance, sentiment analysis aids in designing investment policies to gain high returns [8–10].

There are many approaches employed for sentiment classification or opinion mining ranging from traditional keywords based [1, 11] on advanced machine learning [12] and deep learning models involving transformers [13]-based large language models such as Generative Pretrained Transformer (GPT) [14] and Bidirectional Encoder Representations from Transformers (BERT) [15]. Despite its utility, most of the approaches fail to fully understand the sentiments embedded in the text due to the inherent subjectivity of the language. The differences in expressions, sarcasm, and cultural nuances limit the learning of the model to accurately capture the sentiment [5]. In order to address these challenges, word embedding-based features exploiting the local and global contexts are extracted to improve the sentiment inference in this work.

In order to learn the embeddings from the local context (upto sentence level), Latent Semantic Analysis (LSA) [16] and Continuous Bag of Words (CBOW) [17] models are widely used for text understanding task by generating the word embeddings. The LSA exploits the covariance present in the Term Frequency-Inverse Document Frequency matrix (TF-IDF) matrix and generates the word embeddings by choosing top $|E|$ eigenvectors. The LSA model reveals the relationships even when words are not identical, making it adept at understanding synonyms and nuances. On the other hand, CBOW employs a neural network classifier with embedding layers treating context words as input and inferencing the target word. The embedding vectors are generated from embedding layer after the training. Both of these models are capable of understanding the local semantics exploited in the neighborhood of a target word.

Recently, two very popular language understanding frameworks, namely GPT [14] and BERT [15] based on transformers modules, have been exploited for many Natural Language Processing (NLP) tasks such as question answering, machine translation, text generation, and language inferences. The GPT framework is a generative model focusing on autoregressive learning with left context, while the BERT framework is a masked language model capturing bidirectional context for a broader understanding of word relationships.

In this paper, we intend to assess the significance of local and global context embeddings to evaluate their robustness in the sentiment classification task. The proposed methods are tested against four popular word embeddings algorithms using three different machine learning classifiers on a benchmark finance sentiment analysis task. Our results show the integration of local embeddings with global large word embeddings representations which significantly improves sentiment classification accuracy when compared with local embeddings and global embeddings in isolation.

The main contributions of this work are summarized below:

- We propose to exploit the integration of local context-based embeddings with the large global context-based embeddings to enhance sentiment classification accuracy.
- We conduct extensive experiments on a benchmark finance sentiment classification datasets with the Random Forest, Linear Support Vector Machine, and the Multilayer Perceptron (MLP) classifiers to evaluate the effectiveness of the proposed integrated embeddings.
- We visualize the embeddings generated by LSA, CBOW, GPT, and BERT to provide intuition and support for our claim.

The rest of the paper is organized as follows: In Sect. 2, the literature survey on embedding frameworks is presented. In Sect. 3, we present our proposed methodology. The experiments and results are discussed in Sect. 4. We conclude the work including future directions in Sect. 5.

2 Literature Survey

This work investigates the significance of local semantics-based embeddings when compared with global context-based embeddings which are trained on large Internet text corpus. For local semantics, LSA- and CBOW-based word embedding generators are implemented while to assess the global semantics, BERT- and GPT-based word embedding generators are used. The brief details about these four embedding methods are described here.

2.1 Generative Pretrained Transformer (GPT) [14]

These models are trained on a diverse range of Internet text at massive scale. This enables them to capture intricate patterns and subtlety present in human language. The GPT-based frameworks can exploit the contextual information at passage level, sentence level, or even a conversational level to generate coherent and contextually appropriate responses using the decoder module of the transformer. It learns the

model considering the left context for generating the next word in a sequence. The recent chatGPT, a conversational platform from openAI (https://chat.openai.com/), is integrated into many business applications including conversational chatbots. The details for the GPT framework can be found in [14].

2.2 Bidirectional Encoder Representations from Transformers (BERT) [15]

The BERT model processes the entire input sequence simultaneously, considering the bidirectional context of words when learning. It aims to predict missing words in a sentence, both before and after the target word. This allows it to understand the relationships between words more effectively and enable a more comprehensive and deep understanding of word relationships within a sequence. Recently, Google's BARD (https://bard.google.com/chat) conversational AI framework based on BERT model is deployed in many applications like named entity recognition, question answering, sentiment analysis, and machine translation.

2.3 Latent Semantic Analysis (LSA) [16]

The LSA method works on the principle that words with similar meanings tend to appear in similar contexts. It analyzes a large collection of text to uncover hidden relationships between words by creating a semantic map where words that often hang out together are close neighbors. This is achieved by computing the top $|E|$ eigenvectors of the Term Frequency-Inverse Document Frequency matrix (TF-IDF). This method retains the dimensions ($|E|$ which captures high covariance in the corpus.

2.4 Continuous Bag of Words (CBOW) [17]

This method works on the art of word understanding by immersing it in the context of its linguistic neighborhood of a fixed context window. It understands that words gain meaning from their linguistic neighbors. The model is trained using left and right context using a classifier algorithm to infer the target word. The embeddings are generated from the embedding layer of the model after the training. It captures the words essence by observing how they play together in the symphony of sentences.

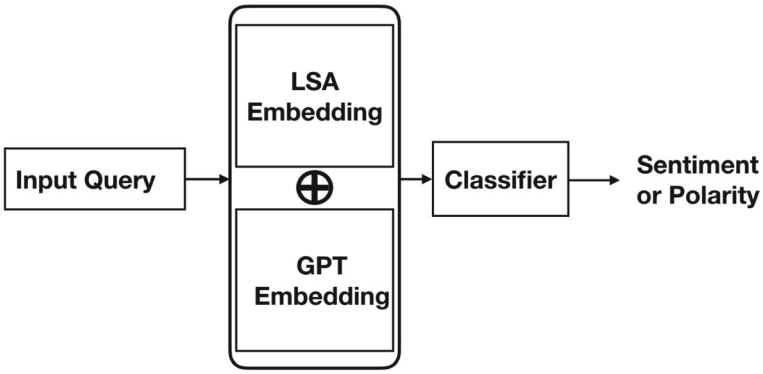


Fig. 1 LSA+GPT framework

3 Proposed SGPT Methodology

This work presents a detailed intuition behind the integration of local semantics with the global semantics for sentiment classification task. The four base methods, namely LSA, CBOW, GPT, and BERT, are chosen, and their suitability is established using extensive experiments. The short details about these methods are presented here for the completeness. In order to establish the intuition, the following four methods are proposed:

3.1 Proposed LSA + GPT

In this method, the embeddings generated using LSA are integrated with the embeddings generated by GPT model. This framework is illustrated in Fig. 1.

3.2 Proposed LSA + BERT

In this method, the embeddings generated using LSA are combined with the embeddings generated by BERT model. This framework is illustrated in Fig. 2.

3.3 Proposed BERT + GPT

In this method, the embeddings generated by BERT model are integrated with the embeddings generated by GPT model. This framework is illustrated in Fig. 3.

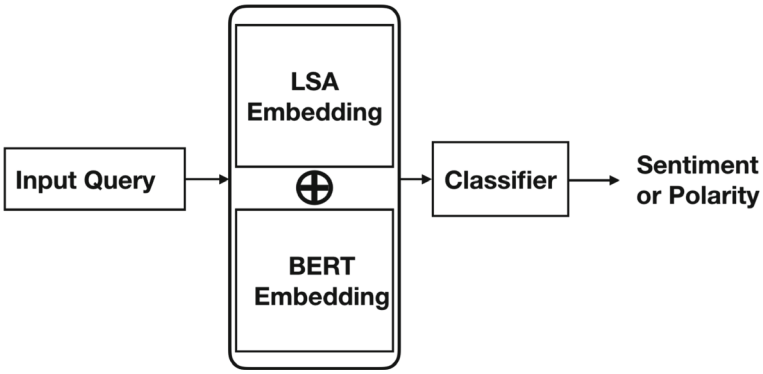


Fig. 2 LSA+BERT framework

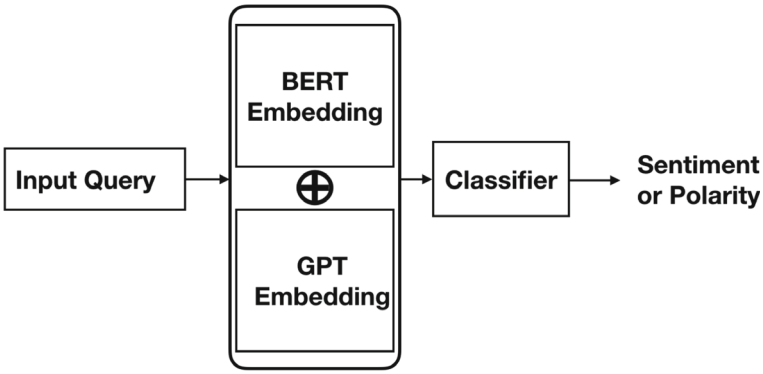


Fig. 3 BERT+GPT framework

3.4 Proposed LSA + BERT + GPT

This method exploits the contextual information extracted by LSA, GPT, and BERT models. The embeddings generated by LSA and GPT models are combined with the embeddings generated by BERT model. This framework is illustrated in Fig. 4.

Similarly, the integration of the embeddings generated by both of the transformer-based models GPT and BERT is combined with CBOW for the support of the proposed intuition. In addition, a framework with all the embeddings (combined) is presented in Fig. 5.

We have also conducted the empirical study to assess the effect of bias removal in the sentiment classification data. The outline of the proposed frameworks is presented in Algorithms 1–3.

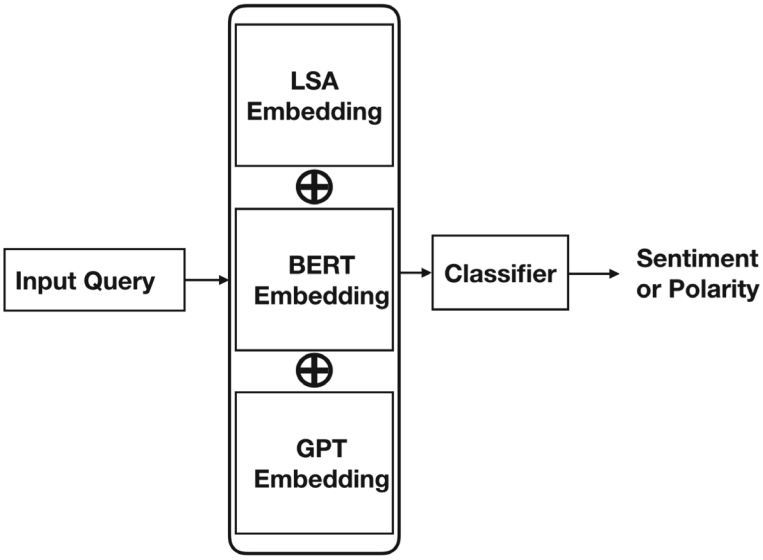


Fig. 4 LSA+BERT+GPT framework

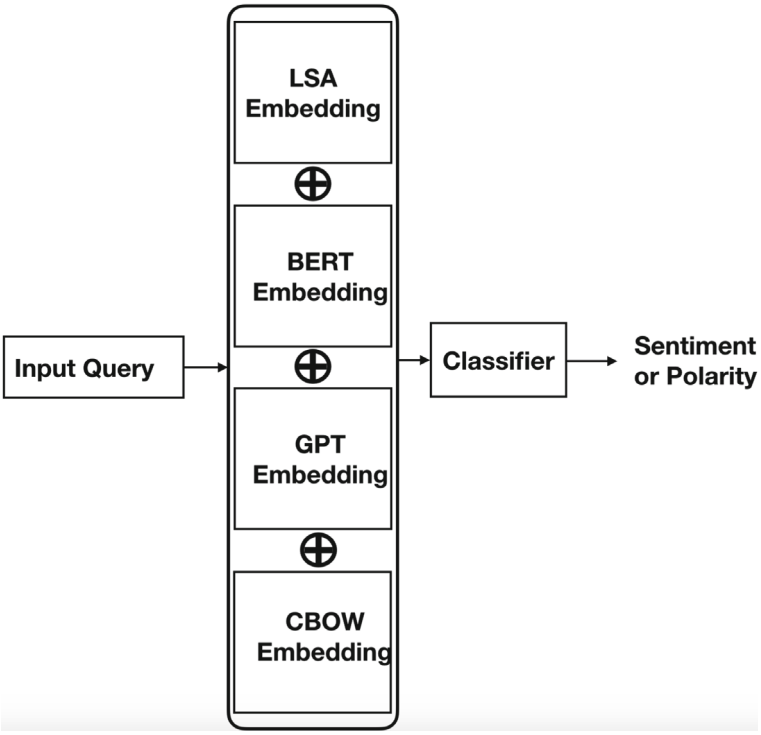


Fig. 5 LSA+BERT+GPT+CBOW framework

Algorithm 1: Integrated Embedding

Data: Raw Sentiment training data X, A list of Embedding algorithms LA for integration
Result: Integrated Embedding Matrix E

```
Preprocess X using tokenization, stop word removal, and lemmatization;
E = {};
for each Embedding algorithm LA_i do
    for each sample x_j ∈ X do
        Generate embeddings e_j using LA_i by aggregating the word
        embeddings for the words in sample x_j;
    Compute the embeddings E_i for X by stacking the aggregate embeddings
    e_j for all x_j ∈ X;
    E = E ∪ E_i;
return Integrated embedding Matrix E;
```

Algorithm 2: SGPT Training

Data: Raw Sentiment training data X, Label vector y, Integrated Embeddings Matrix E
Result: SGPT Model SM, Preprocess parameters P

```
Preprocess X using tokenization, stop word removal, and lemmatization and
store the preprocessing parameters in P;
Prepare Training dataset X using Embeddings Matrix E;
Apply a classifier C on X with 20% validation dataset from X for tuning the
hyper-parameters of C (Uses cross-validation);
SM = Trained model of X with tuned hyper-parameters;
return Integrated embeddings Matrix E and P;
```

Algorithm 3: SGPT Inference

Data: Raw Query sample x_q, SGPT Model SM, Preprocess parameters P
Result: SGPT Model inference y (sentiment label)

```
Preprocess x_q using preprocess parameters P;
Compute embeddings for x_q (Refer Algorithm 1);
y = compute label using SM;
return SGPT Model inference y;
```

3.5 Time Complexity of the Proposed Algorithms

- **Algorithm 1: Integrated_Embedding**
  Step 1: Tokenization, stop word removal, and lemmatization typically have linear time complexity O(tN), where N is the number of samples in X and t is the average number of words in a sample.
  Step 4: It is assumed that generating embeddings for each word takes L_i time for ith embedding algorithm for each word in the corpus. The time complexity for generating the embeddings for a sample x_j is $O(t\sum_i^k L_i)$ where k is the number of embedding algorithms to be integrated.
  Step 6: Stacking the embeddings can be considered $O(Nt\sum_i^k L_i)$ in the worst case.
  The overall time complexity of Algorithm 1 is $O(Nt\sum_i^k L_i)$.
- **Algorithm 2: SGPT Training**
  Step 2: Time for preprocessing is $O(tN)$, where N is the number of samples in X and t is the average number of words in a sample.
  Step 3: Preparing the training dataset based on the embeddings using Algorithm 1 requires $O(t\sum_i^k L_i)$ time where, L_i represents ith embedding algorithm, k denotes the number of embedding algorithms to be integrated.
  Step 4: Training the classifier using cross-validation can be $O(KJM)$, where K is the number of folds, J is the number of hyperparameter combinations to try, and M is the time complexity of the classifier.
  The overall time complexity of Algorithm 2 is $O(Nt\sum_i^k L_i + (KJM))$.
- **Algorithm 3: SGPT Inference**
  Step 1: The time for preprocessing is O(t).
  Step 2: Computing embeddings can be considered $O(t\sum_i^k L_i)$ time where, L_i represents ith embedding algorithm, k denotes the number of embedding algorithms to be integrated.
  Step 3: Computing the label using the trained model is typically $O(T|E|)$, where T is the inference cost per dimension of the query and $|E|$ denotes the size of integrated embedding dimensions.
  The overall time complexity of Algorithm 3 is $O(t\sum_i^k L_i + T|E|)$.

4 Experiments

We have conducted experiments on a benchmark finance sentiment classification dataset. The local embeddings are obtained using LSA and CBOW while the global context-based embeddings are extracted from GPT- and BERT-based frameworks. Further, the combinations of various embeddings for the sentiment classification task are compared to validate our claim.

4.1 Dataset

The finance sentiment analysis dataset is used for assessing the significance of various word embeddings methods. The dataset is comprised of the samples from two popular datasets, namely FiQA and Financial PhraseBank datasets. This data is comprised of 5842 samples with three labels, namely positive, negative, and neutral. The details of the dataset can be found in [8]. This dataset can be downloaded from https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis. The dataset is divided into 80% training and 20% testing parts randomly. The hyper-parameters for the classifiers were chosen using fivefold cross-validation with grid search using 20% validation data randomly sampled from the training set. The average test accuracies are reported in Table 1.

4.2 Experimental Setup

The word embeddings with varying dimensions have been extracted, and various combinations of these embeddings have been used to evaluate the significance of embeddings on sentiment classification task. The various embedding methods with their hyper-parameters are mentioned below:
Latent Semantic Analysis (LSA) and Continuous Bag of Words (CBOW) embeddings:
Embedding Dimension = 100
GPT and BERT embeddings:
Embedding Dimension = 768 (default and fixed as per the framework)

We have employed three classifiers, namely random forest (RF), support vector machine (SVM), and multilayer perceptron (MLP). The number of estimators (trees) in RF is set to 100, and the value of cost parameter is chosen as 1.0 with linear kernel for the SVM classifier. The MLP classifier is set with three hidden layers having 150, 100, and 50 neurons, respectively. The ReLU activation, Adam optimization with 300 epochs are used in all the experiments. Each experiment was conducted multiple times (K times) to account for variability, and the results were averaged. The experiments have been carried out on a virtual machine (VM) having 80 core CPU, 512 GB RAM, and NVIDIA V100 GPU operating within a Big Data framework.

4.3 Results and Discussion

Accuracy:
The test accuracy is presented in Table 1. It can be observed from Table 1 that random forest (RF) method achieves higher accuracy against all the embeddings when the local embeddings from CBOW and global embedding vectors from BERT are com-

Table 1 Average test accuracy of different models with various embeddings

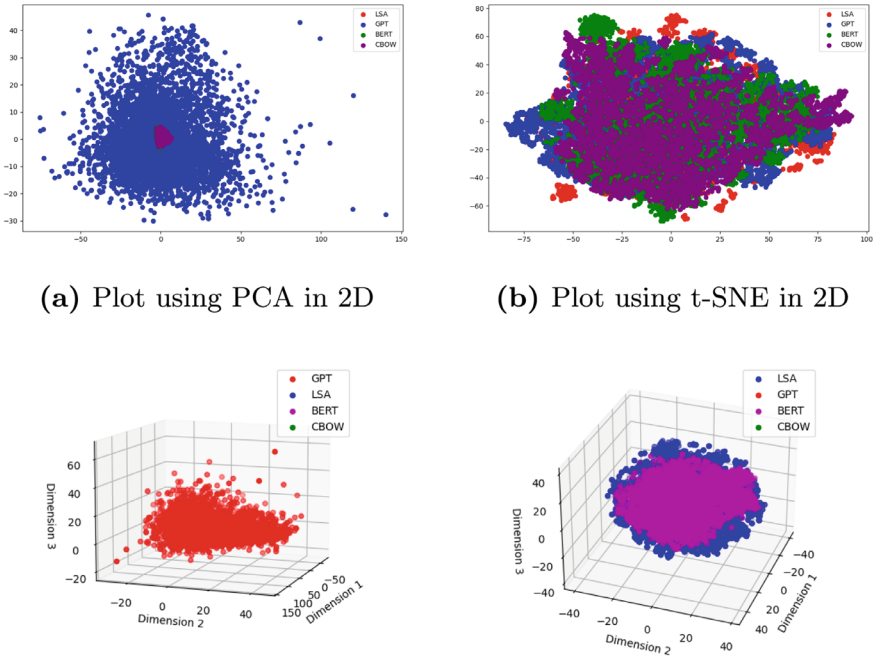
Embeddings ↓ Models →	RF	SVM	MLP
LSA [16]	61.33	62.78	70.74
CBOW [17]	58.85	62.95	69.54
BERT [15]	67.06	70.74	72.88
GPT [14]	65.78	69.88	71.08
Proposed SGPT models			
LSA+BERT	66.72	71.17	70.74
LSA+GPT	65.69	70.31	69.54
BERT+GPT	66.89	67.75	72.88
LSA+CBOW	60.73	67.06	61.76
CBOW+BERT	**67.15**	71.59	70.23
CBOW+GPT	65.09	69.54	69.71
LSA+BERT+GPT	66.72	68.17	71.08
LSA+BERT+CBOW	67.40	**71.94**	69.54
LSA+GPT+CBOW	65.44	70.14	69.37
BERT+GPT+CBOW	66.89	67.66	72.02
BERT+GPT+CBOW+LSA	66.29	67.92	**73.13**

bined. It can also be observed from Table 1 that linear support vector machine (SVM) method achieves higher accuracy when the local embeddings from LSA and CBOW, and global embedding vectors from BERT are combined. Further, the multilayer perceptron (MLP), a neural network model, achieves the best set of accuracies when all local context embedding vectors and all global context embedding vectors are combined. It can be observed from the results that the large embedding models such as GPT and BERT capture the global semantics effectively and lack on local semantics while learning. From the results, it is advisable to integrate local embeddings with the embeddings generated by large language models.

Embeddings: In order to investigate the embeddings generated by LSA, CBOW, GPT, and BERT, the embeddings using t-SNE and PCA are plotted in Fig. 6a, b for 2D, Fig. 6c, d for 3D, respectively. These plots (Fig. 6) are generated using Embedding Projector tool available at https://projector.tensorflow.org/. From these figures, it can be observed that these embeddings from LSA and CBOW follow different distributions when compared with GPT and BERT.

5 Conclusion

This paper proposed an improved sentiment classification approach (SGPT) using local and global context information in the text data. This method tries to reduce the challenges incurred by less focusing on local context in the state-of-the-art word



(a) Plot using PCA in 2D

(b) Plot using t-SNE in 2D

(c) Plot using PCA in 3D

(d) Plot using t-SNE in 3D

Fig. 6 Training sample plot using emdeddings

embedding algorithms based on GPT- and BERT-based frameworks for sentiment classification. The proposed SGPT method can be used for most of the text classification tasks without any changes. In the future, we would like to integrate the local context information while training the large language models to enhance the text classification task.

References

1. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and trends® in information retrieval
2. Liu B (2012) Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies
3. Socher R et al (2013) Recursive deep models for semantic compositionality over a sentiment treebank. Conference on empirical methods in natural language processing (EMNLP)
4. LeCun Y, Zhang X, Zhao J (2015) Character-level convolutional networks for text classification. Adv Neural Inf Process Syst 28:649–657
5. Wei C, Wang B, Kuo CC (2022) Task-specific dependency-based word embedding methods. Pattern Recogn Lett 159:174–180

6. Cambria E, White B (2014) Jumping nlp curves: a review of natural language processing research. IEEE Comput Intell Mag
7. Wang Y, Kuo C-CJ, Wang B, Chen F (2021) Efficient sentence embedding via semantic subspace analysis. 25th ICPR, 119–125
8. Korhonen P, Wallenius J, Malo P, Sinha A, Takala P (2014) Good debt or bad debt: Detecting semantic orientations in economic texts. J Am Soc Inf Sci 65(4):782–796
9. Manandhar S, Komninos A (2016) Dependency based embeddings for sentence classification tasks. Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies, pp 1490–1500
10. Gao Y, Wei X, Tian Y, Liu L, Liu Q, Huang H-Y (2018) Task-oriented word embedding for text classification. 27th ACL, pp 2023–2032
11. Hutto CJ, Gilbert E (2014) Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Eighth international conference on weblogs and social media
12. Manning CD, Pennington J, Socher R (2014) Glove: global vectors for word representation. Empirical methods in natural language processing (EMNLP), pp 1532–1543
13. Vaswani A et al (2017) Attention is all you need. Advances in neural information processing systems (NeurIPS)
14. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2018) Language models are unsupervised multitask learners. OpenAI blog, 1(8)
15. Chang MW, Devlin LK, J, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
16. Foltz Landauer PW, Thomas K, Laham D (1998) An introduction to latent semantic analysis. Discourse processes, pp 259–284
17. Chen K, Corrado G, Mikolov T, Dean J (2013) Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR

Deep Learning-Based Classification of Dietary Restricted Genes

Sunidhi Priyadarshini and Shyantani Maiti

Abstract Dietary restriction (DR) is a well-researched nutritional manipulation of ageing that involves limiting an organism's food intake without causing starvation and has been demonstrated to delay ageing in experimental species. DR can extend life by promoting adaptive alterations in gene expression. In this aspect, this paper presents the deep learning method's applicability to studying ageing genes related to dietary restriction. Then, the accuracy of the train and test datasets is analysed. Further, the AUC score is computed to determine the feasibility of the model for each dataset.

Keywords Ageing · Dietary restriction · Deep learning · Confusion matrix · Auc score

1 Introduction

Ageing is associated with a greater risk of illness and mortality because it alters homeostasis and reduces one's ability to react to environmental cues [1]. Understanding the essential molecular pathways behind ageing has gathered significant research interest due to the widespread desire to reverse and eliminate its harmful effects [2]. It is well-established that numerous processes, including telomere erosion, epigenetic modifications, proteostasis decline, cellular senescence, mitochondrial dysfunction, impaired nutrient detection, stem cell depletion, altered intracellular communication, and genomic instability, contribute to the changes associated with ageing [3]. A gradual deterioration in physiological health that results in diminished functioning and an increased risk of mortality defines the ageing process.

S. Priyadarshini (✉) · S. Maiti
Department of CSE, National Institute of Technology, Durgapur, WB, India
e-mail: sp.23cs1504@phd.nitdgp.ac.in

S. Maiti
e-mail: smaiti.cse@nitdgp.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_49

The most crucial factor leading to illness and mortality is ageing. Ageing has always been an inevitable aspect of life, but is it something that cannot be stopped or even reversed? Although many believe that becoming older is a disease, this is still a mystery. Environmental factors, including lifestyle choices, can significantly impact the pace of ageing. For example, a higher prevalence of diseases associated with age like type II diabetes, cardiovascular disease, stroke, and cancer has been linked to obesity or excess calorie consumption [4]. Several studies suggest that some lifestyle choices, including regular exercise and heart-healthy diets, might enhance our health and lengthen our lives. Dietary/Caloric Restriction (DR), which involves consuming fewer calories and nutrients without becoming hungry, is one such intervention. More than seven decades have passed since McCay et al. [5] first noted that limiting laboratory rats' food consumption increased their lifespan. Since dietary restriction (DR) is still the only environmental intervention to show the extended lifespan of both vertebrates and invertebrates, it has acquired a reputation as the gold standard in ageing research [6]. Therefore, DR is considered a valid public mechanism for extending life duration.

The observation [5, 6] has been confirmed recently in other model organisms [7, 8]. These findings showed that these organisms had longer lifespans and longer health spans (the amount of time they lived in a healthy state), with a marked decline in age-related diseases, including diabetes, heart disease, and several types of cancer. For example, mice who had been fed a diet high in vitamins and minerals, which comprised 35% of their ad libitum consumption, lived an average of fifty-three months, as opposed to thirty-five months in the control ad libitum-fed group [9].

Reduced nutritional intake without resulting in starvation is a calorie restriction (CR), and it has been demonstrated to increase longevity and general health in a variety of species, including humans. Numerous medications and other naturally occurring substances found in food (known as nutraceuticals) have been suggested to mimic the effects of calorie restriction, which has sparked a wave of studies looking into the effectiveness of these substances in stopping ailments related to age and encouraging longer lifespans and improved health.

In recent times, the most promising intervention for extending life and improving health is dietary restriction (DR). Research involving a range of species has shown that DR promotes lifespan and slows the ageing process. Reducing overall dietary calorie intake while preserving sufficient vitamin and mineral levels is the goal of diet resistance [2]. Based on the available data, DR appears to do the following three things: (1) lessen oxidative damage by decreasing Reactive Oxygen Species (ROS); (2) decrease circulating insulin and glucose levels, which results in a decrease in cell growth and division towards maintenance and repair; and (3) decrease levels of growth hormone and insulin-like growth factor [10]. It is yet unclear what mechanism underlies these pro-longevity benefits.

Additionally, new high-throughput sequencing technologies have led to the emergence of an abundance of publicly accessible omics data on ageing [11]. As a result, machine learning (ML) techniques are the foundation of a new method of studying

the ageing process. These methods analyse features of genes and proteins, such as protein-protein interactions, metabolic pathways, and Gene Ontology (GO) terms, to uncover patterns in gene or protein function. In this context, several supervised algorithms are employed in [12], while Random Forests and Easy Ensemble Classifier are used in [10]. Previous studies [13] indicate that deep learning techniques hold tremendous potential for bioinformatics applications. Deep learning algorithms have yet to be used in much research on ageing and dietary restrictions.

2 Proposed Methodology

2.1 Datasets Construction

The ML terminology used in this work: A feature is any measurable attribute of an instance; an instance is any age-associated gene or protein found in the datasets. Five different datasets are used in this work which are discussed below. These datasets offer a plethora of data to represent every ageing-related gene that is being studied (i.e. genes extracted from the GenAge "Model organisms" database; a gene is related to ageing if a minimum of one wet-lab research shows that modifying the gene alters the model organism's longevity or ageing phenotype noticeably [10]). The number of features and instances used in datasets are discussed below.

2.1.1 PathDIP Dataset

The features in this dataset are binary, meaning that each attribute shows whether or not an ageing gene belongs to the corresponding PathDIP pathway. To do this, a dataset was downloaded from the PathDIP [14] database, where the features were biological pathways, and the instances were genes associated with ageing. 1640 pathways and 986 ageing-related genes were found in the resulting dataset; 110 of these genes were identified as AgeingDR-related and 876 as AgeingNotDR-related.

2.1.2 KEGG-Pertinence Dataset

Here each attribute is binary and shows whether an occurrence of an ageing-related gene is associated with a specific KEGG pathway or not. The [15] command of the R language package Lima was used to obtain a data frame that linked KEGG pathways to the genes they included, which was then used to generate this dataset. Subsequently, 312 KEGG pathways were obtained by keeping only those pathways connected to at least one gene relevant to ageing. 799 age-related genes were found

in the resulting dataset, of which 705 were identified as AgeingNotDR-related and 94 as AgeingDR-related.

2.1.3 GTEx Dataset

This database [16] was consulted for the retrieval of the median expression levels of human genes across fifty-five distinct anatomical tissues. Next, only genes associated with ageing were kept, yielding 1111 genes related to ageing: 114 were associated with AgeingDR-related, and 997 were related to AgeingNotDR-related.

2.1.4 GO Terms Dataset

Each binary attribute denotes a distinct GO term. A BioMart query produced 8640 different GO terms that were listed as Biological Process GO terms linked to the genes associated to ageing. All GO terms' ancestors were initially retrieved using the GO.db package in R [17]. Then, as new predictive features, those ancestors were combined with this dataset. Through this process, the total number of GO terms associated with ageing-related genes increased from 4877 (i.e. without ancestors) to 8640 (i.e. containing both the ancestors and the original retrieved GO terms). The resultant dataset contains 1124 genes linked to ageing: 1010 identified as AgeingNotDR-related and 114 as AgeingDR-related.

2.1.5 Protein-Protein Interaction (PPI) Adjacency Dataset

The PPI network of human physical interactions was obtained by downloading data from BioGrid [18]. A graph was constructed from this network, which contained 850 genes associated with ageing. Out of 850 genes, 86 were identified as AgeingDR-related and the rest, 764, as AgeingNotDR-related. Next, 5718 features were obtained by extracting a dataset including binary features from the adjacency matrix of the PPI-graph.

2.2 Deep Learning

New techniques for data analysis are being introduced as the complexity and multi-dimensionality of the data being produced in the field of nutrition continue to rise. Because of its inherent qualities, machine learning (ML) is a suitable tool for handling this data. Neural networks that use numerous operators placed in nodes to assist in breaking down the problem into smaller portions that are solved independently are used in deep learning. However, implementing neural networks can be very chal-

lenging. A deep learning framework called Keras handles this issue. A high-level deep learning API called Keras is created to implement neural networks.

This work uses the Keras framework to analyse the accuracy of different datasets. The class label of the loaded datasets contains categorical values such as NotDR for Ageing Genes, which are not related to dietary restriction, and DR for Ageing Genes, which are associated with Dietary Restriction. As the prediction in deep learning requires the data to be numeric (binary form), the categorical values present in the dataset must be pre-processed into corresponding numerical values. In this regard, AgeingDR-related genes are represented as 1, and AgeingNotDR-related genes are represented as 0. Similarly, gene names are converted into corresponding numerical values using the OneHotEncoder class from SciKit. The total number of columns for "PathDip Dataset" is 2626 after applying OneHotEncoder. After this, the loaded dataset is split into a training dataset and a testing dataset with a split percentage of 70% and 30%, respectively. The simplest neural network model provided by Keras is Sequential API, which is a linear stack of layers. The initial layer of a sequential model has to know its input shape; subsequent layers can automatically infer shapes. The Sequential() class generates sequential models, and layers are added using the add() method. Activation functions relu, and sigmoid are used since this is a binary classification problem. The model's last layer is a dense layer with a single neuron and a sigmoid activation function that permits a single output neuron. The model is visualized in Fig. 1 for "PathDip Dataset". This model has 2626 input neurons (as total number of columns is 2626) for the input layer. The first, second, and third hidden layers have 256, 64, and 8 neurons, respectively.

Then, the model is compiled, and cross-entropy loss between true and predicted labels is computed. Finally, the fit() function is used to train the model using epochs ranging from 05 to 20 for different datasets with a batch size of 10. The accuracy of the training and testing dataset is computed. The confusion matrix is calculated to calculate the classification accuracy and plotted to visualize the confusion matrix. Finally, the AUC-ROC curve is plotted, and the AUC score is computed to identify the feasibility of the model for each dataset.

3 Results and Discussion

This paper presents a detailed analysis of 5 different datasets related to dietary-restricted genes using the deep learning technique. The datasets are analysed by computing accuracy, confusion matrix, and AUC score. In this regard, Table 1 depicts the accuracy of training and testing datasets. Table 2 represents the confusion matrix and AUC-ROC curve for each dataset. Table 3 represents the AUC score and feasibility of the model.

The performance of each built model is evaluated using the evaluate() function, which will return the loss and accuracy of the model for each dataset. Then, the accuracy_score() function from the sklearn.metrics module is imported to make predictions on the test data using the kerasmodel. According to Table 1, the KEGG-

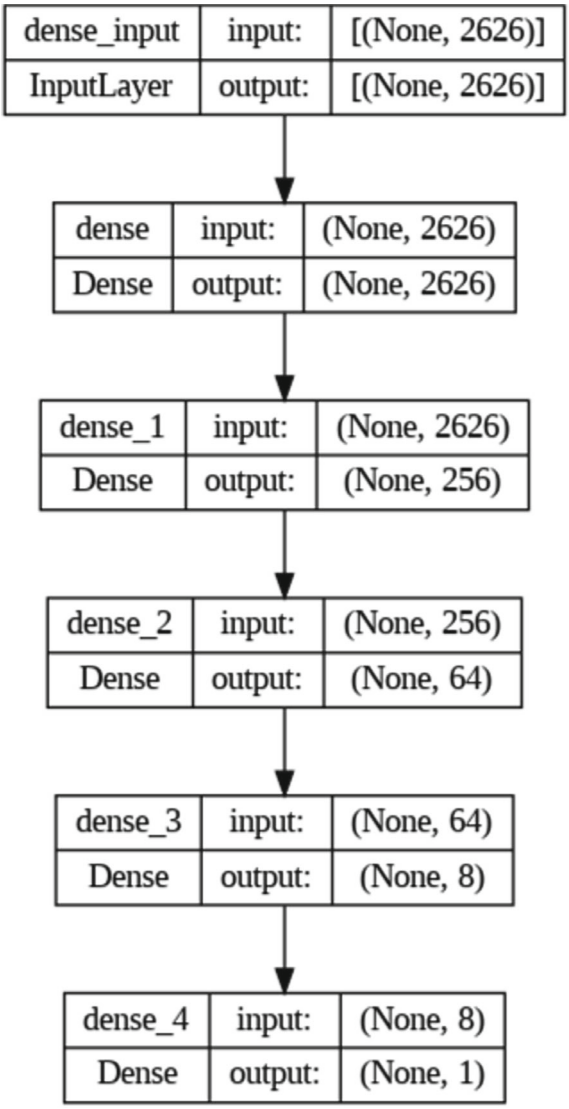


Fig. 1 Visualization of Network Model for PathDIP Dataset

Pertinence feature type shows the highest train accuracy percentage of 99.84, and the GTEx feature type shows the lowest train accuracy percentage of 91.12. Similarly, the GO terms feature type shows the highest test accuracy percentage of 95.56, and the KEGG-Pertinence feature type shows the lowest test accuracy percentage of 87.50.

Then, the ConfusionMatrixDisplay() function is utilized to visualize the confusion matrix for each dataset, illustrating the count of correct and incorrect predictions generated by the model. The roc_curve() function imported from sklearn.metrics module takes the true outcomes (0,1) from the test set and the predicted probabilities

Table 1 Train and test accuracy for different datasets

Name of the dataset	No. of features	No. of instances (Genes)	Train accuracy (%)	Test accuracy (%)
PathDIP	1640	986	99.57	89.86
KEGG- Pertinence	312	799	99.84	87.5
GTEx	55	1111	91.12	89.52
GO terms	8640	1124	98.35	95.56
PPI-Adjacency	5718	850	99.83	91.37

for the positive class. The function gives the thresholds as well as the true positive and false positive rates for each threshold. The corresponding confusion matrix and ROC curve are plotted for each dataset in Table 2.

The roc_auc_score() function is then used to compute the AUC for each ROC. This function requires the true outcomes (0,1) from the test set in addition to the predicted probabilities for the positive class. When the AUC falls within the range from 0.5 to 1, there is a strong likelihood that the classifier can effectively discriminate between positive class values and negative ones. The area under the receiver operating characteristic curve (AUC) shows the expected accuracy in a broad sense. As shown in Table 3, the GO terms feature type has an AUC of 0.78, indicating good prediction performance. In contrast, the PathDIP feature type has an AUC of 0.58, meaning uncertain prediction performance.

4 Conclusion

This article demonstrates the application of the deep learning technique to age-related dietary restriction research. In this regard, the training and test accuracy of five distinct datasets are compiled and examined. At the same time, the confusion matrix and the AUC-ROC curve are presented for each dataset. It is discovered by AUC score analysis that the PathDIP dataset has an uncertain prediction performance, while the GO terms dataset has a good prediction performance. According to the presented research, deep learning can be utilized to analyse dietary-restricted genes. Therefore, deep learning approaches can be used in future research to predict new dietary-restricted ageing genes. Additionally, deep learning methods can be used to identify the most significant features among all the features available in the datasets.

Table 2 Confusion matrix and AUC-ROC curve for each dataset

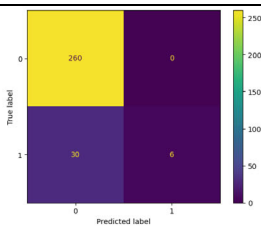
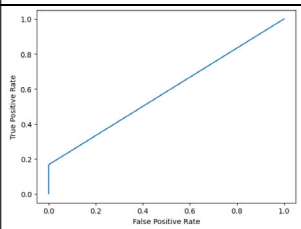
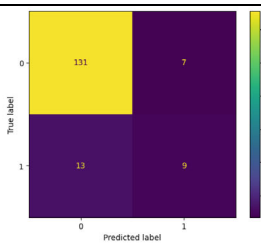
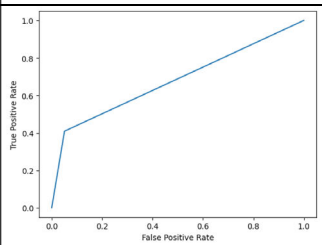
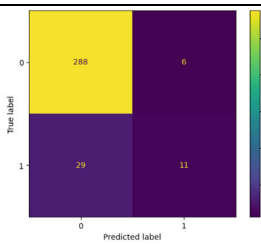
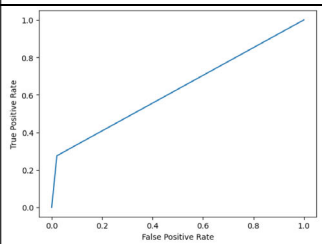
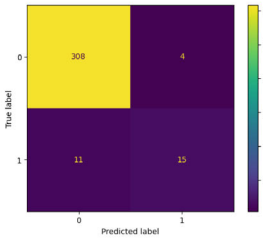
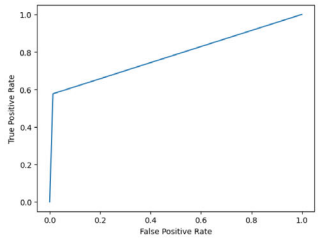
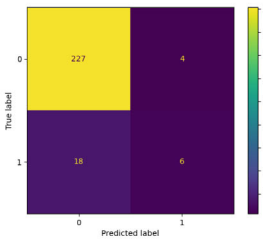
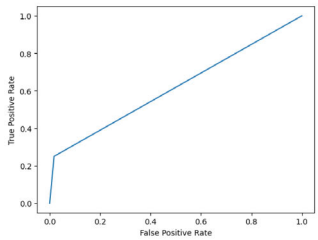
Name of the Dataset	Confusion Matrix	AUC-ROC Curve
PathDIP		
KEGG-Pertinence		
GTEx		
GO terms		
PPI-Adjacency		

Table 3 AUC score and feasibility of model for each dataset

Name of the dataset	AUC score	Prediction capability of model
PathDIP	0.58	Uncertain
KEGG-Pertinence	0.68	Satisfactory
GTEx	0.63	Satisfactory
GO terms	0.78	Good
PPI-Adjacency	0.62	Satisfactory

References

1. MacNee W (2016) Is chronic obstructive pulmonary disease an accelerated aging disease? Ann Am Thorac Soc 13(5):429–437
2. Magalhaes JP, Wuttke D, Wood S, Plank M, Vora C (2011) Genome-environment interactions that modulate aging: Powerful targets for drug discovery. Pharmacol Rev 64(1):88–101
3. Lopez-Otın C, Blasco M, Partridge L, Serrano M, Kroemer G (2013) The hallmarks of aging. Cell 153(6):1194–217
4. Fontana L, Klein S (2007) Aging, Adiposity, and Calorie Restriction. JAMA 297(9):986–994
5. Simpson SJ, Couteur DGL, Raubenheimer D, Solon-Biet SM, Cooney GJ, Cogger VC, Fontana L (2017) Dietary protein, aging and nutritional geometry. Ageing Res Rev 39:78–86
6. Weindruch R (1996) The retardation of aging by caloric restriction: studies in rodents and primates. Toxicol Pathol 24(6):742–745
7. Gillespie Z, Pickering J, Eskiw C (2016) Better living through chemistry: caloric restriction (cr) and cr mimetics alter genome function to promote increased health and lifespan. Front Genet 7
8. Wieser D, Papatheodorou I, Ziehm M, Thornton JM (2011) Computational biology for ageing. Philosophical Trans Royal Soc B: Biol Sci 366(1561):51–63
9. Fabris F, Magalhaes JP, Freitas A (2017) A review of supervised machine learning applied to ageing research. Biogerontology 18(2):177–188
10. Vega Magdaleno G, Bespalov V, Zheng Y, Freitas A, Magalhaes JP (2022) Machine learning-based predictions of dietary restriction associations across ageing-related genes. BMC Bioinform 23
11. Longo VD (2009) Linking sirtuins, igf-i signaling, and starvation. Exp Gerontol 44(1):70–74
12. Lee C, Longo V (2016) Dietary restriction with and without caloric restriction for healthy aging. F1000Res
13. Sapoval N, Aghazadeh A, Nute MG, Antunes DA, Balaji A, Baraniuk R, Barberan C, Dannenfelser R, Dun C, Edrisi M, Elworth RAL, Kille B, Kyrillidis A, Nakhleh L, Wolfe CR, Yan Z, Yao V, Treangen TJ (2022) Current progress and open challenges for applying deep learning across the biosciences. Nat Commun 13(1):1728
14. Rahmati S, Abovsky M, Pastrello C, Jurisica I (2016) Pathdip: an annotated resource for known and predicted human gene-pathway associations and pathway enrichment analysis. Nucleic Acids Res 45:419–426
15. Ritchie M, Phipson B, Wu D, Hu Y, Law C, Shi W, Smyth G (2015) Limma powers differential expression analyses for rna-sequencing and microarray studies. Nucleic Acids Res 43(7)
16. Carithers L, Ardlie K, Barcus M, Branton P, Britton A, Buia S, Compton C, DeLuca D, Peter-Demchok J, Gelfand E, Guan P, Korzeniewski G, Lockhart N, Rabiner C, Rao A, Robinson K, Roche N, Sawyer S, Segr'e A, Taherian M (2015) A novel approach to high-quality postmortem tissue procurement: the gtex project. Biopreserv biobank 13(5):311–319

17. Marc C (2019) Go.db: a set of annotation maps describing the entire gene ontology. R Package Version 3.11.4
18. Stark C, Breitkreutz B-J, Reguly T, Boucher L, Breitkreutz A, Tyers M (2006) Biogrid: a general repository for interaction datasets. Nucleic Acids Res 34(suppl1):535–539

A Preliminary Work on Tenyidie Stemming Corpus Creation and Deep Learning Applications

Teisovi Angami and Themrichon Tuithung

Abstract Tenyidie language belonging to the Tibeto-Burman family is one of the major languages of Nagaland in northeastern India. It is a low-resource, tonal, SOV, and a high agglutinative language. Stemming is an important Natural Language Processing (NLP) task wherein the stem or the root word is identified and extracted. To the best of the authors knowledge, there has been no work reported on corpus creation and stemming using deep learning for the Tenyidie language. In this paper, we build an annotated corpus of 5400 manually stemmed words in Tenyidie to apply deep learning to the stemming task. We applied state-of-the-art deep learning models, such as the Encoder-decoder models, and compared our results with the rule-based approach, which uses both affix stripping and root word matching methods. Our experimental result shows that the deep learning models outperform the rule-based approach. We achieved the highest accuracy of 85.55% with Encoder-decoder (BLSTM) with attention. We also provide an analysis of the overstemmed and understemmed words in our test dataset.

Keywords Tenyidie · NLP · Stemming · Encoder-decoder

1 Introduction

The Tenyidie language of Nagaland in northeastern India is a Tibeto-Burman Language which is spoken by the community mostly in Nagaland and parts of Manipur and Assam known as *Tenyimia*. Tenyidie is a low-resource, tonal [1–4], SOV [3–5], and a high agglutinative [3] language. Stemming is an important Natural Language

T. Angami · T. Tuithung
Department of Computer Science and Engineering, National Institute of Technology Nagaland, Dimapur, India
e-mail: ttuithung@nitnagaland.ac.in

T. Angami (✉)
Department of Information Technology, School of Engineering and Technology, Nagaland University, Lumami, India
e-mail: teisovi@gmail.com; teisovi@nagalanduniversity.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_50

Processing (NLP) task wherein the stem or the root word is identified and extracted. Stemming is useful in many other NLP tasks, such as understanding language morphology, information retrieval, etc.

This paper aims to create a 5400 words manually stemmed dataset for building an automatic stemmer for the Tenyidie Language. In this work, we have applied state-of-the-art deep learning models such as the Encoder-decoder models and compared these results with the rule-based approach. To the best of the authors knowledge, this is the first work on building an automatic stemmer for Tenyidie.

The remaining of this paper is structured as follows: Sect. 2 provides an overview of the Tenyidie Language, Sect. 3 gives a review of the related works, Sect. 4 describes the corpus creation and statistics of the manually created stemming corpus for the Tenyidie Language, Sect. 5 discusses the deep learning models experimental setup, Sect. 6 provides a discussion of the experimental results and Sect. 7 concludes and presents future works.

2 Overview of Tenyidie Language

In this section, we provide an overview of the Tenyidie Alphabet and its morphology.

2.1 Tenyidie Alphabet

Tenyidie Alphabet comprises 25 letters, 19 consonants, and 6 vowels. It is similar to the Latin Alphabet except that the letters 'Q' & 'X' are excluded and include an additional letter 'Ü'. A Tenyidie sentence[1] is shown here:

'Ga cüyakezhamia shürhokuoya.'
(Those who eat vegetable are healthier.)

2.2 Morphology

By nature, Tenyidie is highly agglutinative. Due to this behavior, it possesses high inflected forms from its stem word by combining with prefix(es), infix(es), and suffix(es) [3]. A few example illustrations showing affixation in Tenyidie are shown in Table 1.

[1] *Note* The tone is not marked in the orthography (written form).

Table 1 Illustrations of affixation in Tenyidie

Tenyidie word	Stem	Prefix	Suffix
Kikruko (the family)	kikru	–	ko
Katiwate (cut)	kati	–	wa+te
Kethepfu (to be brave)	thepfu	ke	–

Table 2 Unstemmed dataset statistics

Particulars	No. of characters
Max. word length	19
Min. word length	2
Average word length	8.85

Table 3 Unstemmed dataset affix statistics

Particulars	No. of characters
Min. prefix character length	0
Max. prefix character length	14
Min. suffix character length	0
Max. suffix character length	14

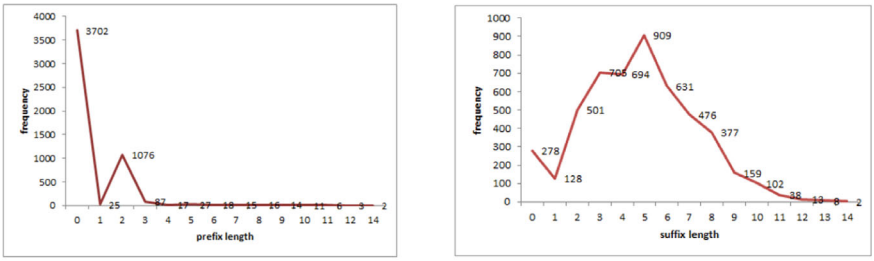
3 Related Works

Substantial work on stemming is available for resource-rich languages. However, for Tenyidie language, very limited work on NLP research has been done. Teisovi et al. [6] worked on building a rule-based stemmer for Tenyidie, in which the affixation rules were reviewed. These rules were used in building suffix combinations of 1206 and 11 prefixes. A root word dictionary of 766 unique words was also used along with the generated affixes, achieving an accuracy of 60.69% on 430 Tenyidie words.

4 Corpus Creation

To apply any learning model, we need to create a training dataset to train the learning model. This section describes the manual annotated corpus creation process and the statistical analysis of the Tenyidie stemmed words appearing in our annotated dataset. The corpus creation process starts with data acquisition in which we collected a total of 16,022 unique words from news data (Tenyidie news daily *"CAPI"*). Next, we removed the non-Tenyidie words and other tokens (numbers, symbols, etc.) from this unique word list. From the remaining unique words, we manually stemmed 5400 words. One annotator carried out the manual stemming process by following the stemming guidelines given by the language expert.

Table 2 provides details of the unstemmed dataset, and Table 3 provides the affix details of the unstemmed dataset. Figure 1a shows the prefix character length versus



(a) Prefix character length versus frequency

(b) Suffix character length versus frequency

Fig. 1 Prefix/suffix character length versus frequency

Fig. 2 Tenyidie stem words word-cloud

frequency. We observed the highest frequency with a prefix length of 0 (words with no prefix) and the lowest with a prefix length of 14. Figure 1b shows the suffix character length versus frequency. We observed the highest frequency with a suffix length of 5 and the lowest with a suffix length of 14. The word cloud for the stem words in the manually stemmed dataset is shown in Fig. 2.

5 Deep Learning Experimental Setup

This section discusses the deep learning architectures used and the experimental setup. We model our stemming task as a machine translation task since the input sequence length (unstemmed word) is not equal to the output length (stemmed word), as is observed in language translation. Our experiments use the Encoder-decoder

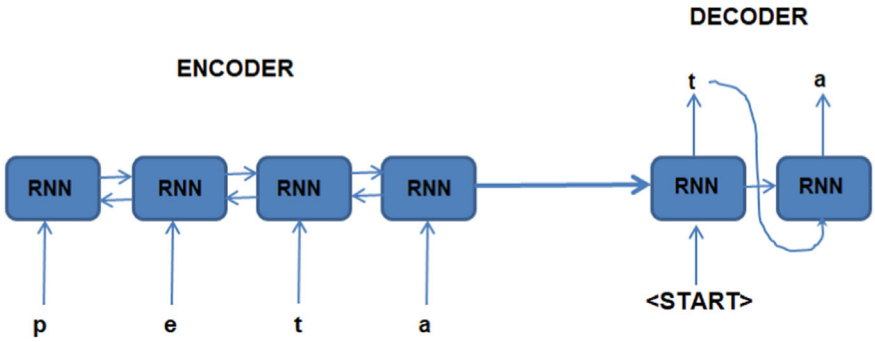


Fig. 3 Encoder-decoder model illustrating input and output

Table 4 Training, validation and test dataset

Particulars	No. of words
Total	5400
Training	4320
Validation	540
Test	540

model with attention mechanism [7]. The models we implemented are: *Encoder-decoder (GRU*[2]*) with attention* and *Encoder-decoder (BLSTM*[3]*) with attention*. The first model implementation is available in tensorflow.org[4], and for the second, we used the OpenNMT [8] toolkit. Figure 3 shows an overview of the Encoder-decoder model with the input and the output. On the left side, we have the encoder part; on the right, we have the decoder part. The encoder receives the unstemmed word input, i.e., *"p e t a"*, and the decoder outputs the stemmed word, i.e., *"t a"*. To perform our experiments, we choose the following hyperparameters:

GRU with attention: batch_size = 16, embedding_dim = 128, units = 512, epochs = 40.

BLSTM with attention: batch_size = 64, embedding_dim = 128, units = 512, epochs = 40.

To evaluate our learning models, we split the annotated stemming dataset into 80, 10, and 10 % words for training, validation, and testing, respectively, as shown in Table 4. These models are then compared with the rule-based approach as proposed in [6].

[2] Gated Recurrent Unit.

[3] Bidirectional long short-term memory.

[4] https://www.tensorflow.org/text/tutorials/nmt_with_attention.

Table 5 Performance evaluation of stemming experiments

Stemming technique	Words stemmed	Accuracy (%)
Rule-based	291/540	53.88%
Encoder-decoder (GRU) with attention	398/540	73.70%
Encoder-decoder (BLSTM) with attention	462/540	**85.55**%

The highest accuracy obtained is indicated in bold

Table 6 Evaluation of the best learning model

Details	No. of words	Percentage (%)
Correctly stemmed	462	**85.55**%
Over-stemming	7	1.29%
Under-stemming	42	7.77%
Wrongly stemmed	29	5.37%

The overall stemming accuracy is indicated in bold

6 Experiment Results & Discussions

This section reports the results of the experiments that we conducted. The stemming accuracy formula is expressed as:-

$$\text{Accuracy} = \frac{Number_of_words_stemmed_correctly}{Total_number_of_words}$$

We also compare our results with the rule-based approach proposed in [6], which is based on the Tenyidie language affixation rules by applying both affix stripping and root word matching.

The test dataset is comprised of 540 Tenyidie words. Table 5 reports the performance of the different techniques. From the experiments performed, we achieved the highest accuracy of 85.55% using Encoder-decoder (BLSTM) with attention, as compared to Encoder-decoder (GRU) with attention achieving 73.70% and rule-based achieving 53.88%. Table 6 reports the stemming evaluation results with the best learning model (i.e., BLSTM with attention). We provide stemming details such as stemmed correctly or incorrectly, and details of overstemming and understemming. Table 7 provides the first 10 test words stemming results for the best learning model. The statistics of the overall stemmed words results for the best model are shown in Table 8. The results show that the deep learning approach performs well with any number of affixes combinations. The model loss for both the Encoder-decoder models are shown in Fig. 4. Additionally, example heatmaps of two words are given in Fig. 5.

Table 7 First 10 test words stemming results

Word	Approx. translation to English	POS tag	Affixes details	Predicted output	Stemming results	
					stemmed correctly	overstemmed/ understemmed
kikruko	family	noun	**kikru**+ko	kiku	No	stemmed incorrectly
katiwate	cut	verb	**kati**+wate	kati	Yes	–
lashüketa	again	verb	**la**+shüketa	la	Yes	–
tuketa	burnt	verb	**tu**+keta	tu	Yes	–
tezeketa	captured	verb	**teze**+keta	te	No	understemmed
rhipie	drawn (sketch)	verb	**rhi**+pie	rhie	No	overstemmed
tezeketau	the person who got captured	verb	**te**+zeketau	te	Yes	–
khashütaketuo	what will be given	verb	**kha**+shütaketuo	kha	Yes	–
pushüko	what has been told	verb	**pu**+shüko	pu	Yes	–
kekhotaketuo	to separate	verb	**kekho**+taketuo	kho	No	understemmed

The stem (root) of the word is indicated in bold

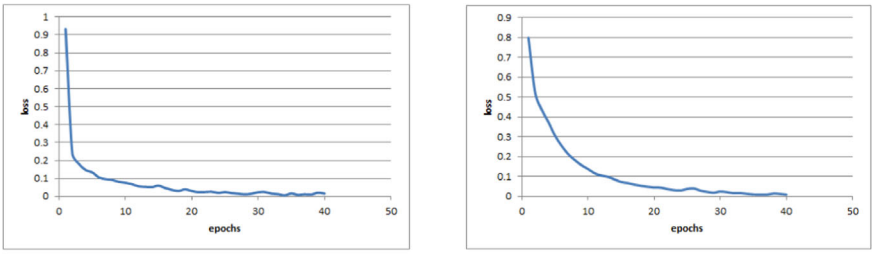
Table 8 Overall stemmed words results

	Prefix/Suffix length	No. of words stemmed	Total no. of words	% (percentage)
Stemmed correctly	prefix (1 char.)	64	70	**91.43**
	suffix (1 char.)	131	153	**85.62**
	suffix (2 char.)	123	131	**93.89**
	suffix (3 char.)	68	76	**89.47**
	suffix (4 char.)	11	15	73.33
	suffix (5 char.)	2	2	**100**
Stemmed incorrectly	prefix (1 char.)	6	70	8.57
	suffix (1 char.)	22	153	14.38
	suffix (2 char.)	8	131	6.1
	suffix (3 char.)	8	76	10.53
	suffix (4 char.)	4	15	26.67
	suffix (5 char.)	0	2	0

Accuracy > 80% for different prefix and suffix length is indicated in bold

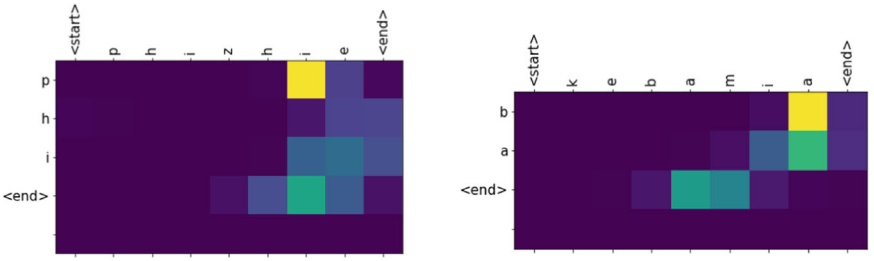
7 Conclusion & Future Works

This work focuses on corpus creation for the stemming task for the Tenyidie Language. In this paper, we build an annotated corpus of 5400 manually stemmed words



(a) Encoder-decoder (GRU) model loss (b) Encoder-decoder (BLSTM) model loss

Fig. 4 Encoder-decoder model loss



(a) Tenyidie stemming heat map 1 (b) Tenyidie stemming heat map 2

Fig. 5 Tenyidie stemming heat maps

in Tenyidie to apply deep learning to the stemming task. We applied the state-of-the-art deep learning models, such as Encoder-decoder models with attention, and compared our results with the rule-based approach, which uses both affix stripping and root word matching methods. Both the deep learning models (Encoder-decoder with attention) outperform the rule-based approach. We achieved the highest accuracy of 85.55% for BLSTM with attention. We also provide an analysis of overstemmed and understemmed words in our test dataset. In future work, we will test on a larger dataset since the deep learning models perform well on larger datasets.

Acknowledgements We are grateful to Dr. Keneichanuo Mepfhüo for her immense help during the creation of the Tenyidie stemming corpus.

References

1. Burling R (1960) Angami Naga phonemics and word list. Indian Linguistics. 21:51–60
2. Giridhar PP (1980) Angami Grammar. CIIL, Mysore

3. Kuolie D (2006) Structural description of Tenyidie—A Tibeto-Burman Language of Nagaland. Ura Academy
4. Ravindran N (1974) Angami Phonetic Reader. CIIL, Mysore
5. Ezung MK (2007) The Structure of Tenyidie (Angami) Syntax. PhD Thesis
6. Angami T, Tuithung T (2023) Towards Building Stemmer for Tenyidie Language. In: 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA), pp 1-6. IEEE. doi:https://doi.org/10.1109/ICAICTA59291.2023.10389862
7. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, and Polosukhin I (2017) Attention is all you need. In: 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, pp 6000-6010. https://dl.acm.org/doi/10.5555/3295222.3295349
8. Klein G, Kim Y, Deng Y, Senellart J, Rush AM (2017) Opennmt: Open-source toolkit for neural machine translation. arXiv preprint. https://doi.org/10.48550/arXiv.1701.02810

On Building a Firewall with White Hat Bots in the Neighborhood of Malicious Bots in Large-Scale Networks

Ryutaro Matsumoto and Shingo Yamaguchi

Abstract This paper proposes a firewall that prevents the spread of malicious bot infections by a small count of white-hat bots targeting large networks. By strategic placing bots, we identify bottlenecks in the network structure and minimize the count of bots deployed to prevent the spread of malicious bot infections. The proposed method is a malicious bot neighborhood-based method for large networks that considers the trade-off relationship between minimum firewall and minimum count of infections. The method builds a firewall by identifying bottlenecks using minimum node cuts at the edges of the edges of malicious bots and firewall search areas and placing white hat bots at those locations. The algorithm determines the firewall search area by its distance from the malicious bot. This was implemented as an algorithm and evaluated. indicator were set, with lower scores meaning better firewall performance. In comparison to the surround tactic, the count of white-hat bots was reduced by 50% and the count of infections increased by 24%.

Keywords Firewall · White-hat botnet · Neighborhood-based · Heuristic solution · Large-scale network

1 Introduction

In recent years, IoT devices have enriched our lives. A common example is the ability to remotely control household appliances by connecting them to IoT devices. While this enhances the convenience of our daily lives, there have been significant incidents targeting IoT devices, posing a substantial threat from botnets specifically designed for IoT devices [1]. In response to malicious bot attacks targeting IoT devices, our

R. Matsumoto (✉) · S. Yamaguchi
Graduate School of Sciences and Technology for Innovation, Yamaguchi University,
2-16-1 Tokiwadai, Ube-shi Yamaguchi, 755-8611, Japan
e-mail: c123vgw@yamaguchi-u.ac.jp

S. Yamaguchi
e-mail: shingo@yamaguchi-u.ac.jp

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_51

research group is actively developing a security system called the Botnet Defense System (BDS) [2], which utilizes a white-hat botnet [3] as a means to counteract such threats.

Simultaneously, IoT devices are highly vulnerable due to hardcoded security, making them prime targets, given their widespread presence. A significant challenge arises from the rapid proliferation of IoT devices, with the global count continually increasing each year [4]. In other words, the count of devices connecting to the Internet is on the rise, leading to dynamic changes in network structures. Consequently, security systems leveraging reinforcement learning, as proposed in [5], have been suggested. In order to cope with botnets in such a changing network structure, we are focusing on research to eliminate botnets or to improve the security of IoT devices. However, focusing on the spread of infection as one of the threats of botnets, we believe it is more important to contain and minimize the damage than to improve device security and eliminate malicious bots.

In this paper, we propose a firewall which utilizes white hat bots to protect devices with the fewest count of firewalls possible. The method uses the maximum flow minimum cut of graph theory to search for bottlenecks in order to construct the minimum firewall. In addition, in order to construct a minimum firewall while mitigating the count of infections, we propose a method of restricting the region and searching for the maximum-flow minimum cut within the region.

Section 1 of this paper provides background on this research. Next, Sect. 2 describes the previous studies and the problems with the firewall algorithm. In Sect. 3, the trade-offs of firewalls and the proposed method are illustrated with examples, and an algorithm is proposed. In Sect. 4, we perform an experiment to proposed method is evaluated and compared with conventional methods. Finally, Sect. 5 presents the conclusions of this paper.

2 Related Work

2.1 BDS and White-Hat Botnets

In the Botnet Defense System (BDS), a white-hat worm is utilized to eliminate malicious bots, featuring two key characteristics: secondary infectivity and a limited lifespan. This worm initiates secondary infections on compromised IoT devices, effectively supplanting the malicious bots with their white-hat counterparts. Moreover, the worm is designed with a finite lifespan, ceasing to exist once it reaches the end of its pre-determined duration. Even after the white-hat worm dissipates, the white-hat bots, now without worm capabilities, persist. They continue safeguarding devices from malicious bot infections until the devices undergo a reboot.

Studies related to BDS include the following. Kageyama et al. [6] proposed a method for building a white hat botnet to surround a malicious botnet. This strategy aims to systematically eliminate the malicious botnet without exposing its presence

by surrounding it with a white-hat botnet. Furthermore, the tactical deployment of a white-hat botnet around the malicious botnet not only serves the purpose of eradication but also plays a role similar to a firewall. This dual functionality not only conducts removal operations but also acts as a preventive measure against the expansion of infections caused by the malicious botnet.

Pan et al. [7] proposed a novel machine learning-infused method for tactical response through zoning in BDS. The concept of zoning was introduced to understand malicious botnets spread unevenly across an IoT network. This allows us to divide the network into zones and to perform a tactical response for each zone. However, depending on the distribution of malicious bots, many white hat bots may be needed to enclose them. Therefore, Matsumoto et al. [8] formalized a problem of building a firewall with a fewer bots than enclosing and then provided a solution. However, the proposed method is an exact solution method, which requires an enormous amount of time to find an answer among all possible solutions and cannot support large-scale network (Fig. 1).

2.2 Firewall and Brute-Force Method

A firewall is a white hat botnet that surrounds the malicious bots. The firewall is a white hat bot botnet placed around the white hat botnet. This prevents further infection of the malicious bots. The quality of the firewall should be such that the fewer the number of white hat bots and the fewer the number of malicious bots infected, the better. Trivial firewalls are made up of white hat bots with white bots placed in close proximity to and surrounding the malicious bots [6]. In this case, the number of infections of malicious bots is zero. Using this as a baseline, we propose a method to reduce the number of white hat bots while allowing the number of infections to increase. Figure 2 shows an example of a fire wall with surround. However, the number of white hat bots can be further reduced by allowing the number of normal devices to be infected. This is the proposed method shown in Fig. 1.

2.3 Firewall Building Problem

We define a problem to build a firewall with white-hat bots as follows:

Definition 1 *(Firewall building problem)* Here the targeting IoT system infected by a malicious botnet is given as follows:
IoT Network $G(V, E)$, Malicious bots distribution M, White-hat bots used count n, Nodes protected by the firewall m.

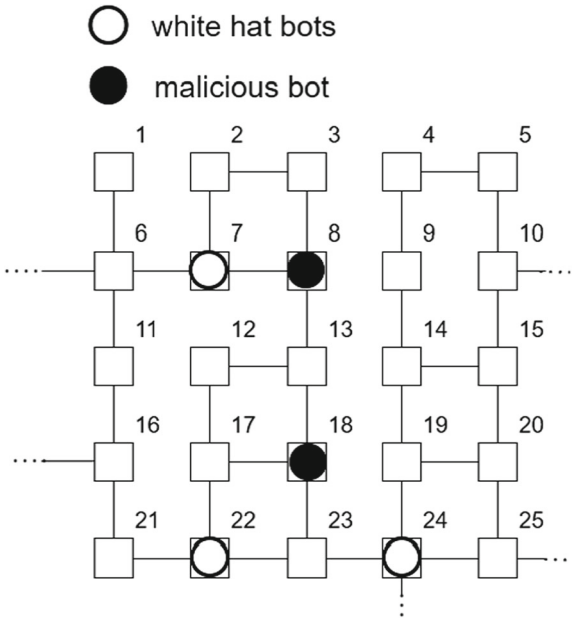


Fig. 1 Proposed fire wall shape

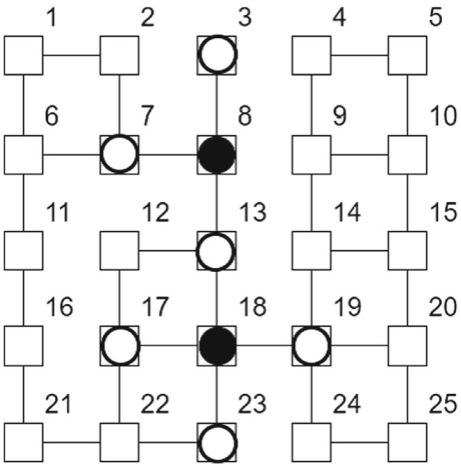


Fig. 2 Surround tactic

Question.
What is a WH bot distribution W that forms the firewall? Here

$$W : v\ (\in V) \mapsto \begin{cases} 1 & \text{Infectedwithwhite} - \text{hatbots} \\ 0 & \text{uninfected} \end{cases}$$

□

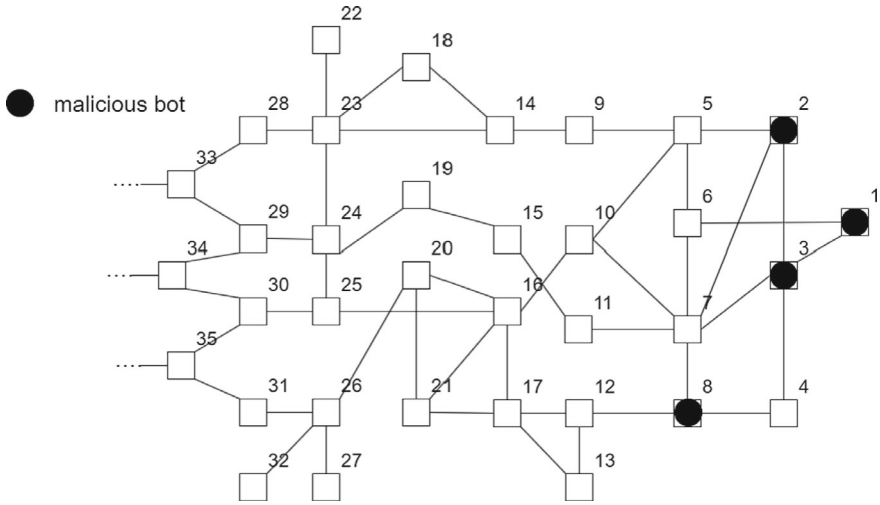


Fig. 3 Question : Find the distribution of white hat bots to protect more than half of the nodes in the IoT network (V, E) from a malicious botnet spreading from v_1, v_2, v_3, v_8

2.4 Examples of Problem

The target network is shown in Fig. 3. In this question, there are edges at nodes v_{33}, v_{34}, v_{35}. Therefore, conventional methods cannot solve this problem.

Example 1 *What is a WH bot distribution W that forms the firewall? Here The solution to the answer is shown in Fig. 4.*

$$W_1 : v\ (\in V_1) \mapsto \begin{cases} 1 \ \textit{if}\ v \in \{v_{18}, v_{19}, v_{20}\} \\ 0 \ \textit{otherwise} \end{cases}$$

In this problem, the upper limit is the number of surround, 5. Find the firewalls that can protect more than half of the houses with $n = 3$. As a result, the 18 node can be protected by placing white hat bots at node counts v_{18}, v_{19}, v_{20}. The procedure to output this answer is to set the firewall search area and search for the minimum cut inside it. The firewall search area is set in terms of the radius distance from one point of the malicious bot. The details of the procedure are as follows. Examples of firewall failures and extending the firewall search area are explained in the next Section.

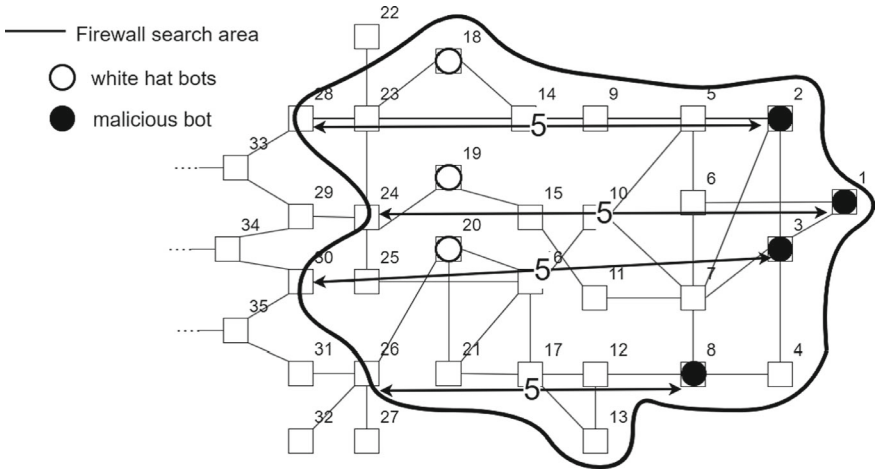


Fig. 4 Answer for Question of Fig. 3: The firewall protects more than half of the nodes in the network from a malicious botnet

Algorithm 1 White hat bot minimum method algorithm

Input: Certain distance from malicious bots r, Network $G(V, E)$, Distribution of malicious bots M.

Output: $n = |W|, m^*$

1: ▷ Contraction of malicious bots
$M_c \leftarrow$ CONTRACTED NODES$(\sum_{v \in V} M(v))$

2: ▷ Malicious bot distance r to node, condensing the edge of the Firewall search area
$H \leftarrow$ CONTRACTED NODES$(\{v \in V \mid r(M, v)\})$

3: ▷ Minimum cut of malicious bot contraction point and edge of Firewall search area
$W \leftarrow$ MINIMUM NODE CUT(M_c, H)

4: ▷ Find the count of nodes between M and W, find the count of infections, and find m^*
$m^* \leftarrow |V| -$ NODES BETWEENNESS$(G, \sum_{u \in V} W(u), \sum_{v \in V} M(v))$

5: **return** $n \leftarrow |W|, m^*$

3 Heuristic Solution Method

3.1 Trade-Off Between White-Hat Bots and Infected Devices

As mentioned earlier, the primary objective of firewalls is to build firewalls with fewer white hat bots than surround tactics. But there is a trade-off to this objective. That trade-off exists between a minimal firewall and a minimal count of infections. The key issue here is determining how far to allow the infected area to go. This balances the trade-offs. Determine areas that may be infected and build a firewall at the bottleneck between the malicious bots and the set of devices that protect the devices from infection. Bottlenecks are identified by using existing minimum node cuts [9]. Therefore, the firewall construction algorithm can be divided into two main stages.

- Determine how far to allow infection.
  - Determination of the maximum allowable range of infection.
- Determination of white hat bot placement location.
  - Find the minimum node cut between the edge of the maximum firewall search area and the malicious botnet. Applying existing algorithms.

First, a simple algorithm is used to illustrate this trade-off. The white hat bot minimum method in Algorithm 1 sets up a firewall search area and performs minimum node cuts at its edges and infection points to find bottlenecks. Please see Fig. 5. The count of surroundings in this infection situation is 10. This is an example of a case in which the Firewall search area is taken far from the infection point. We look for bottlenecks in the Firewall search area, and the edges are v_{39}, v_{50}, v_{59}, v_{63}, v_{68}, v_{72}, v_{83}, v_{85}, v_{86} and v_{94}. If we find the minimum node cut between this edge and the malicious bots, the answers are v_{17}, v_{46}, and v_{74}. Next, see Fig. 6. It can be seen that the firewall search area is smaller than that in

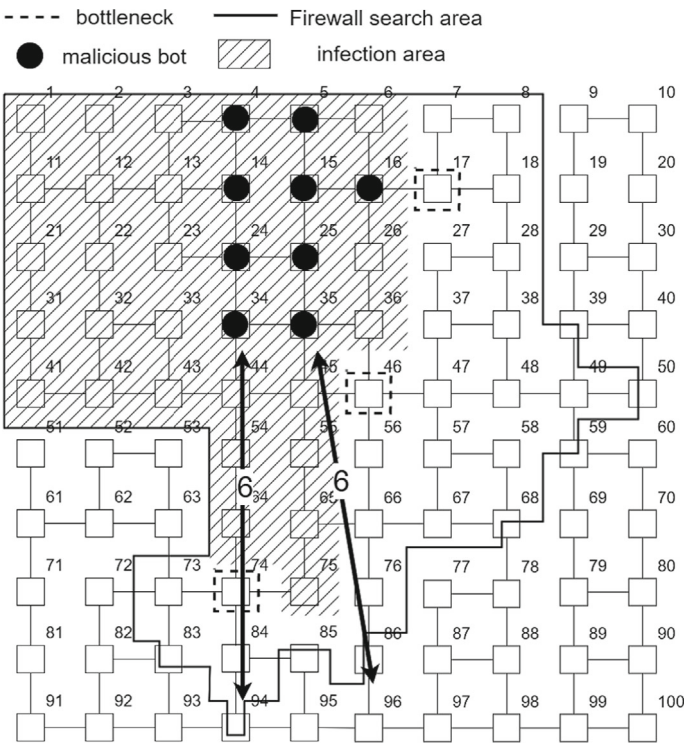


Fig. 5 White hat bot minimum method with malicious bots:search area is taken large and the bottlenecks are v_{17}, v_{46},and v_{74}

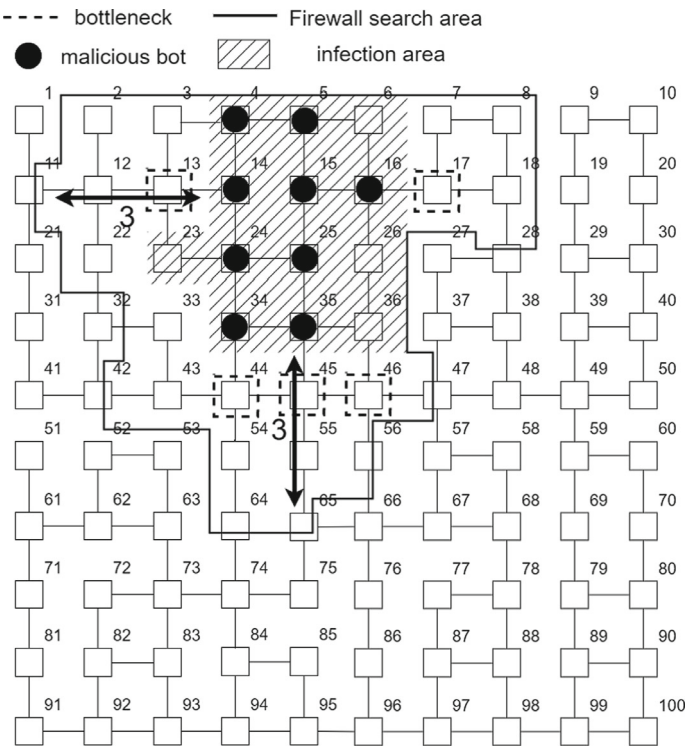


Fig. 6 White hat bot minimum method with malicious bots. Search area is taken small and the bottlenecks are v_{13}, v_{17}, v_{44}, v_{45}, v_{46}

Fig. 5. The edges of the firewall search area at this time are v_{11}, v_{28}, v_{42}, v_{47}, v_{56}, v_{64}, v_{65}, and the bottleneck locations are v_{13}, v_{17}, v_{44}, v_{45}, v_{46}. When comparing the two, it is worth noting the count of bottlenecks and the size of the infected area. Infected areas are shaded. In Fig. 5, the count of bottlenecks is smaller, but the infected area is larger in Fig. 6, the count of bottlenecks is larger than in Fig. 5, but the infection area is smaller. As can be seen from this example, the relationship between minimal firewall and a minimal count of infections is a trade-off relationship: a larger Firewall search area results in a larger infection area at the expense of a larger range of bottleneck selection, while a smaller Firewall search area results in a smaller infection area at the expense of a larger count of bottlenecks. In both cases, suppressing the bottleneck will result in a firewall with fewer than the count of surrounded, but the expected characteristic is to build a firewall that can limit the count of infections as much as possible with a small count of white-hat bots, rather than with surround tactics.

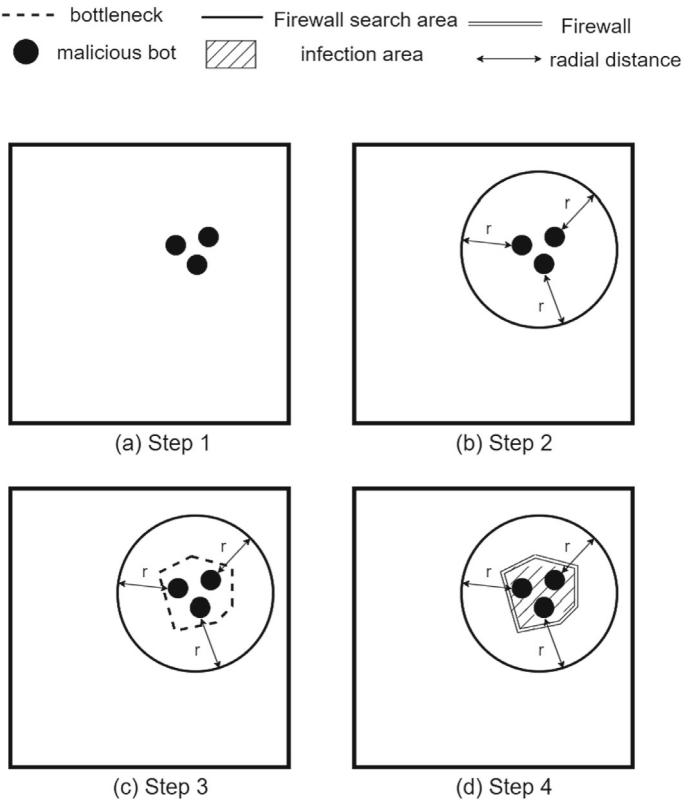


Fig. 7 Diagram of the new method step by step

3.2 New Method Concept

From the previous section, we consider new methods. New methods should focus on the extent to which they allow infection. Although there are various possible shapes for the infection area, this paper uses concentric circles with the parameter r as the distance from malicious bots. The proposed method focuses on the distance from the malicious bots in order to minimize the count of infections with emphasis on minimizing the count of infections. We give up minimizing the count of white-hat bots based on the network shape and search for bottlenecks by making the acceptable range of infection as small as possible. This allows firewalls to be constructed even in large networks. It also makes it possible to construct firewalls in a more practical time than the exact solution method. Figure 7 shows a schematic of the concept of the new method step by step. In Step 1, a malicious bot is infecting the workstation. Step 2 defines a Firewall search area from malicious bots. The radius distance in the neighborhood of malicious bots is an important point of this new method. In Step 3,

Algorithm 2 Overall algorithm, network firewall search algorithm

Input: (V, E), Distribution of malicious bots M, Count of white-hat bots of white-hat bots launched by the BDS n, Count of nodes to be protected from infection m^*.
Output: $n = |W|, m^*, R_{proposed}$.
1: $r = 0$
2: **do**
3: Enter n and find m^*
4: $r \leftarrow r + 1$
5: **if** $K = r$
6: **break**
7: $n, m^* \leftarrow$ Malicious bot neighborhood-based algorithm(r)
8: **while**$(n < n^*$ and $m^* \geq |v|/2)$
9: ▷ Find the distance r of the minimum valuation value.
$R_{proposed} \leftarrow \underset{r \in \{1,2,\ldots K\}}{\text{argmin}} \{f(r)\}$
10: **return** $n \leftarrow |W|, m^*, R_{proposed}$

the minimum node cut between the edge of the Firewall search area and the malicious bots is determined and is used as the bottleneck. In Step 4, white hat bots are output to the bottleneck to build a firewall. These are the steps of the new method.

3.3 Malicious Bot Neighborhood-Based Solution

Algorithm 2 is the overall algorithm. The while statement requires that n be less than the count of surroundings and that the count of nodes to be protected, m^*, be half of the nodes in the entire network. Finally, the $f()$ valuation function is used to find the $R_{proposed}$ of the minimum valued distance. The restricted distance in the neighborhood of r in Algorithm 1 is the neighborhood-based algorithm for malicious bots. This algorithm is a part of Algorithm 2 and outputs n and m when r is determined.

The definition of K is as follows:

$$K(M) = max\,\{\text{shortest-path}(M, u) \mid u \in V(G)\} \cdot 0.5$$

Let K be half the eccentricity from malicious bots.

If the bottleneck of the minimum cut is less than or equal to the input n and satisfies m^*, then that is the answer.

4 Evaluation

4.1 Valuation Indicators

The valuations in this paper are based on an originally defined valuation metric. The distance of the minimum value of the evaluation of the proposed method is $R_{proposed}$. The white hat bot minimum method r is $R_{conventional}$. We evaluate the performance of a new method for minimizing the count of infections, the malicious bot neighborhood-based method. In this experiment, we show how the proposed malicious bot neighborhood-based method varies in terms of the count of infections, the count of white-hat bots, and the search distance in the neighborhood, and describe the validity of $R_{proposed}$.

Next, we compare the performance of the malicious bot neighborhood-based method with that of the basic method, the white hat bot minimum method with malicious bots. This experiment compares the proposed method with the conventional white hat bot minimum method. The purpose is to show the superiority of the proposed method in terms of the count of bots, count of infections, and indicator.

The count of firewalls and the count of infections are the key indicator for evaluation. The main objective of this evaluation is to reduce the count of firewalls n (#WH bot). While placing importance on the count of firewalls, the count of infections (#Infection) is also taken into account in the valuation. The indicator ε is as follows.

$$\varepsilon = \#\text{WH bot} \cdot \delta + \#\text{Infection} \cdot \omega$$

δ and ω are weights, in this case $\delta = 30$ and $\omega = 3$.

4.2 Experimental Results

$R_{conventional}$ is set to 10 this time. The maximum radius K of the proposed method is set to 6. Valuation of the performance of the proposed method. Figure 8 plots #infection and #WH bot for each radius of the proposed method. Focusing on the count of infections and the count of white-hat bots, the count of white-hat bots decreases in an s-shape as the radius increases. The count of infections increases accordingly. The minimum firewall for the proposed method is when the radius is 6, the count of white-hat bots is 5, and the count of infections is 51. However, when the valuations for a radius $r = 2 \sim 6$ are calculated, the scores are 339, 309, 312, 300, 303, with $R_{proposed} = 5$ having the highest valuation of 300. Thus, it is important not only to simply reduce the count of firewalls required but also to pay attention to the count of infections. In addition, the valuation shows that the valuation value is able to pay attention to both the count of infections and the count of bots.

Next, the proposed method is compared with the white hat bot minimum method. Table 1 shows the comparative results of the white hat bot minimum method (con-

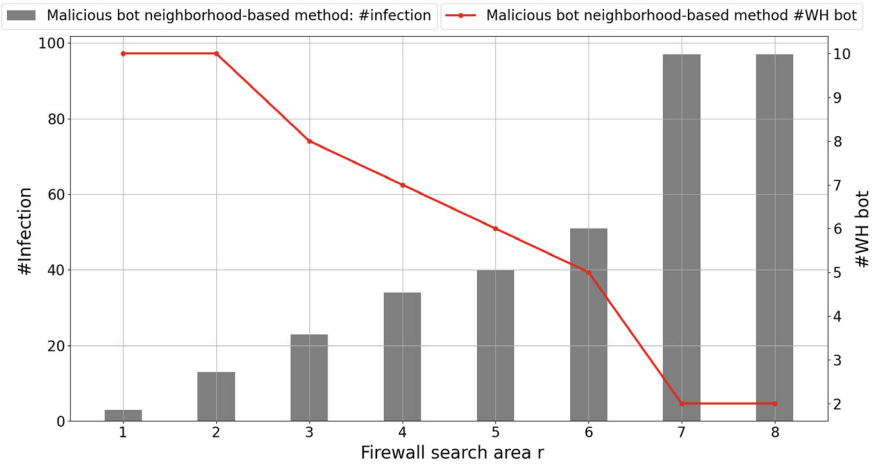


Fig. 8 Performance Valuation Graph for Malicious bot neighborhood-based method. Firewall search area r on the horizontal axis, #WH bot on the right vertical axis, and #Infection the left vertical axis

Table 1 Comparative valuation of conventional and proposed methods

	#WH bot	#Infection	ε(Indicator)
Proposed method	6	40	300
Conventional method	2	97	351

ventional method) and the malicious bot neighborhood-based method (proposed method). The count of infections for the conventional method is as high as 97, while the count of white-hat bots is 2. The proposed method has 40 infections but requires 5 bots. Lower values indicate better firewall performance. The minimum valuation of $R_{proposed}$ is 300 and the minimum valuation of $R_{conventional}$ is 351.

This value indicates that the proposed method is superior to the conventional method. This result indicates that even if the count of white-hat bots required for a firewall is extremely small, the performance of the firewall is not sufficient if the #infection is large. In other words, it is more important to search for bottlenecks in the vicinity of malicious objects than to search for simple bottlenecks in the network, which allows us to find a firewall with a reduced count of white-hat bots required while keeping the count of infections low.

5 Conclusion

In this paper, we proposed a method of building a firewall using fewer white hat bots than the method of enclosing malicious botnets. It was found that there is a trade-off in building firewalls. The trade-off was the relation between the minimum

firewall and the minimum count of infections. To address this trade-off, we proposed a malicious bot neighborhood-based solution that emphasizes minimizing the count of infections by specifying the radius of the firewall search area that could be the infection area in the neighborhood of malicious bots. In indicator ε, we used a new evaluation metric to evaluate the designed firewall. The indicator took into account the count of infections while focusing on the count of white-hat bots. The lower the score, the better performing the firewall. The evaluation result of the proposed malicious bot neighborhood-based was 303, which was lower than the evaluation result of the base white hat bot minimum of 351. These results show that the firewall can be constructed in a practical timeframe in a large network with a smaller count of firewalls than the surround tactic, which is the objective of this paper.

Future work includes comparing the malicious bot neighborhood-based algorithm and New methods to be developed, and investigating what kind of network is appropriate. We will also develop an underlying algorithm that can appropriately choose between these two heuristic methods based on the network structure and the distribution of malicious bots.

Acknowledgements This work was supported by JSPS KAKENHI Grant Numbers JP19K11965 and JP22K12028, and Interface Corporation, Japan.

References

1. Kolias C, Kambourakis G, Stavrou A, Voas J (2017) DDoS in the IoT: Mirai and Other Botnets. IEEE Computer 50(7):80–84
2. Yamaguchi S (2020) Botnet Defense System: Concept, Design, and Basic Strategy. Information 11:516
3. Stephen H, Harvey K, Hughey G, Roberts R, Levin D (2019) Measurement and analysis of hajime, a peer-to-peer IoT botnet, Proc. of 2019 Network and Distributed System Security Symposium
4. IoT Analytics, State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally, https://iot-analytics.com/number-connected-iot-devices/ (Accessed 5 Dec 2023)
5. Zhang Z et al (2022) Artificial intelligence in cyber security: research advances, challenges, and opportunities, Artif Intell Rev 1–25
6. Kageyama T, Yamaguchi S (2021) On Tactics to Deploy White-Hat Worms in Botnet Defense System. Proc. of IEEE GCCE 2021:294–297
7. Pan X, Yamaguchi S, Kageyama T, Kamilin M (2022) Machine-Learning-Based White-Hat Worm Launcher in Botnet Defense System. International Journal of Software Science and Computational Intelligence (IJSSCI) 14(1):1–14
8. Matsumoto R, Yamaguchi S (2023) On building firewall with white-hat bots to prevent the infection spread of malicious botnets, 2023 11th International Conference on Information and Education Technology (ICIET), Fujisawa, Japan, pp 539–543
9. Esfahanian A (2013) Topics in structural graph theory (Connectivity algorithms), Cambridge University Press, pp 268–281

Deep Neural Network-Based Yoga Posture Classification Using Human Pose Estimation

Rohan Pramod Jadhav and Sunita Dhavale

Abstract Yoga is an ancient practise that has been followed to build up our physical and mental fitness. Yoga has been a form of treatment in clinical therapy, improving immunity and assisting the recovery of various chronic conditions. If performed incorrectly in the absence of a yoga expert, there is a risk of injuries such as strains, sprains, slip discs. Hence, there is a need for accurate real-time yoga posture classification techniques to be implemented that can be used as part of any automated yoga posture monitoring applications. This paper aims to perform Deep Learning based automated recognition of yoga asanas in real time from two different views accurately. We prepared a yoga posture custom video dataset titled "DIAT-YogaDS" consisting of five unique asanas from “Suryanamaskar”, a modern-day commonly followed yogic practise. These unique asanas contain occluded portions, and it is found that occlusion sensitive Human Pose Estimation (HPE) models (OpenPose, Detectron2, and YOLOv8-Pose) fail to recognize all key points, while DensePose HPE takes more time compared to MoveNet HPE to extract these features. In this work, to achieve real-time performance, we used MoveNet to obtain joint vector cosine angle features from extracted skeletal body key points. The proposed Deep Neural Network (DNN) model trained on these angle features is lightweight, providing robust detection invariant to both scale and distance. Experimental results show the precision of the proposed DNN to be 0.9428. The proposed DNN model can be implemented on edge devices, and it requires very low computational resources.

Keywords Yoga posture classification · Human pose estimation (hpe) · Skeletal body key points · Cosine angle · Deep neural network (dnn)

R. P. Jadhav (✉) · S. Dhavale
School of Computer Engineering and Mathematical Sciences, Defence Institute of Advanced Technology, Pune, Maharashtra, India
e-mail: rohanpj@rediffmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_53

1 Introduction

Due to the current fast-moving world with tonnes of development and progress, we tend to overwork, face a lot of anxiety, and go through lots of stress and depression that affects us mentally as well as physically, which leads to various stress-based diseases such as hypertension, coronary heart diseases, and cancer. Hence, yoga has become popular recently due to its multiple health benefits, like physical, mental, and spiritual well-being [1]. Performing yoga daily helps to have a healthy lifestyle, especially in the current era where chronic diseases such as COVID-19 can affect a human body and cause various life-threatening issues. After the Indian Government worked towards initiatives like Yoga Day, it was observed that there was an increase in the number of people from various age groups keen to learn yoga.

Recently, a large population is adapting to yoga, while there are very few yoga trainers/experts are available for supervision via physical or online modes. Hence, there is a need for automated yoga posture monitoring applications that allow individual participants to make sure that it is performed in the correct manner, avoiding any injuries [2]. Human Pose Estimation (HPE), along with Deep Neural Network (DNN), is a promising and significant field of research in computer vision and can be used successfully in yoga posture monitoring applications too. HPE can be employed for single or multiple person detection, that follows either a top-down or bottom-up approach method. There are various aspects to human body modelling. A body model has information such as the shape of the human and texture, which can be used to predict body point coordinates and the distance of the body key points for joint estimation. The human body models can be of various types, such as Skeletal-based model, Contour-based model, and Volume-based model. Skeletal-based models are computationally less intensive and hence suitable for real-time applications.

In this work, we adopted a skeletal-based model for the implementation of the real-time yoga posture classification technique. Further, we propose a custom dataset titled "DIAT-YogaDS", consisting of five unique yoga asanas (Ashava Sanchalanasana, Bhujangasana, Hasta Uttanasana, Parvatasana, and Tadasana) from "Suryanamaskar", a modern-day commonly followed yogic practise. These selected yoga asanas are challenging to detect as they can exhibit occlusions while performing. Six different individuals of different heights and physical builds are used to perform these asanas in a closed environment to collect a sample dataset using two RGB cameras placed at two different angles. The extracted frames are resized to 540×960, and background audio noise is removed. Further, we made a detailed analysis of various HPE algorithms (OpenPose, DensePose, Detectron2, YOLOv8-Pose, and MoveNet) on the DIAT-YogaDS dataset and selected MoveNet for our proposed work as it exhibits low sensitivity towards occlusion and faster computation properties. The preprocessed frames act as input to MoveNet, which generates body key points in real time. These key points are processed further to calculate joint vectors and cosine angle features. The proposed DNN model is trained with the extracted angle features for the implementation of the real-time yoga posture classification technique.

2 Related Work

There has been some research performed on using HPE in various applications that allow access to the correctness of movements [3]. Some of the ideas include a marker-based approach that makes use of sensors attached to the human body that track human movements. In 2018, the authors in [4] developed the OpenPose model, which performed multi-person skeletal-based body pose estimation. OpenPose made use of two parallel division approaches of conditional network layers, where one predicted a confidence map of key points and the other predicted Part Affinity Fields (PAFs) to define associations between points, so using stronger links based on PAFs, a bipartite graph is calculated to provide body key points. In 2018, the author in [5] developed the DensePose model, trained on the COCO dataset, that provides a surface-based representation of a contour-based body model with dense correspondences between an RGB image of the human body. In 2021, the authors in [6] introduced MoveNet, a deep neural network for joint profile prediction across variable walking speeds and slopes. This was further improved by Tensorflow [3] to build an ultra-fast and accurate pose detection model that offered two variant models, known as Lightning and Thunder, which were intended for either latency-critical applications or applications that require high accuracy. In 2020, the authors in [7] introduced Detectron2 as part of Facebook AI Research (FAIR) to support the rapid implementation and evaluation of novel computer vision research.

In 2012, the authors in [8] proposed a method to recognize body parts of human depth silhouettes for residents monitoring services at smart homes using trained Random Forests for Human Activity Recognition (HAR). In 2015, the authors in [9], proposed an activity recognition framework based on randomized trees and guided random forests to identify human body activities based on wearable sensors. In 2019, the authors in [10] performed a survey on deep learning approaches for camera pose estimation of RGB images, aiming to improve the deep pose regression solution. In 2020, the authors in [11] proposed a three-dimensional convolutional neural network (3D CNN) architecture for yoga pose recognition. In 2021, the authors in [12] introduced the HSiPu2 dataset, which consisted of a physical fitness action dataset for recognition, in case of HPE, an architecture was developed called the Two-Branch Multi-Stage CNN and LSTM Attention-Based Human Recognition model, which outperformed various traditional machine learning models such as Decision Tree, Random Forest, SVM, Bagging, GBDT, AdaBoost, XGBoost, and Voting. In 2021, the authors in [13] performed an analysis of human-robot interaction for precise analysis with marker-less human pose estimators. This research involved four upper-limb rehabilitation exercises using two HPE models, that is, OpenPose and Detectron2, with two RGB-D cameras. Classification for this paper was performed manually by calculating the angle for each exercise. In 2022, the author in [14] describes feature processing that can be performed on open pose key points for further classification.

In 2023, the author in [15] performed a survey on yogic posture recognition, which involved a study of yoga pose prediction and automatic movement analysis.

Fig. 1 Side-view yogic asana—Ashava Sanchalanasana, Bhujangasana, Hasta Uttanasana, Parvatasana and Tadasana

It involves the study of various ways of local feature extraction, various tool usage, machine learning, and deep learning techniques using transfer learning. The authors highlighted that the current approaches are not capable of predicting if there exists concealed body parts while performing asanas. Also, most large-scale pose estimate datasets comprise typical poses used daily and are not related to yoga asanas, hence making it difficult to use them directly for yogic posture recognition. Hence, there is a need to create a yoga asanas custom dataset along with robust lightweight DNN models to classify yoga poses for successful yoga posture monitoring applications that can run on edge devices.

3 Methodology

3.1 Dataset Acquisition

A custom Yoga asanas dataset, DIAT-YogaDS, is prepared using two RGB cameras placed at two different angles, 90° from each other, to capture front and side views of any asana. A total of five unique yoga asanas from "Suryanamaskar" are captured sequentially (Ashava Sanchalanasana, Bhujangasana, Hasta Uttanasana, Parvatasana, and Tadasana) performed by six different individuals of various height groups and physical builds (see Figs. 1 and 2). A total of 30 videos are captured of yoga asanas being performed for 1 min in mp4 format with a resolution of 1920 × 1080 in landscape and portrait form. Video is resized by 50% to 540 × 960 resolution, and background audio noise is removed as part of data preprocessing, which reduced the size from 9.63GB to 300MB. Further, image generation is performed on the preprocessed data by frame extraction, resulting in a total of 118,108 images.

3.2 Human Pose Estimation

A few of the best HPE algorithms are tested on DIAT-YogaDS to evaluate a good HPE algorithm to proceed.

Fig. 2 Front-view yogic asana—Ashava Sanchalanasana, Bhujangasana, Hasta Uttanasana, Parvatasana and Tadasana

3.2.1 OpenPose

OpenPose [4] is tested on DIAT-YogaDS, found sensitivity towards occlusion while detecting body points (see Fig. 3) and it also exhibits higher computation time.

3.2.2 DensePose

DensePose [5] is tested DIAT-YogaDS and found very low sensitivity towards occlusion while detecting body points (see Fig. 4) but it exhibits higher computation time.

3.2.3 Dectectron2

Detectron2 [7] is tested on DIAT-YogaDS and found a large amount of sensitivity towards occlusion while detecting body points (see Fig. 5), which suggests it is not suited for DIAT-YogaDS.

3.2.4 YOLOv8-Pose

YOLOv8-Pose P6 is tested on DIAT-YogaDS and found some sensitivity towards occlusion while detecting body points (see Fig. 6) but it exhibits a lesser computational time than the above methods.

3.2.5 MoveNet

MoveNet [3] is tested on DIAT-YogaDS and found very low sensitivity towards occlusion while detecting body points (see Fig. 7), also exhibits very low computational time.

Based on testing the best HPE algorithms, we concluded to utilize MoveNet for this paper's further processing for yoga posture recognition and classification.

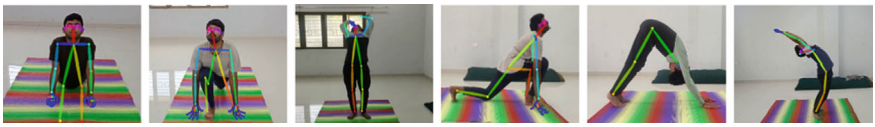


Fig. 3 OpenPose results

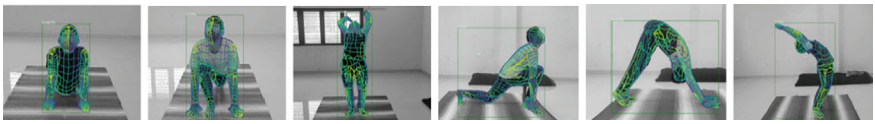


Fig. 4 DensePose results

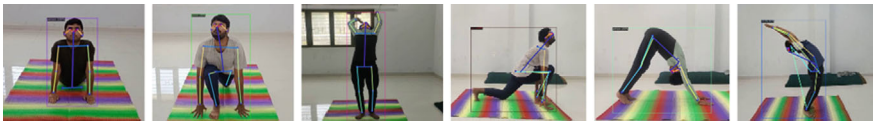


Fig. 5 Detectron2 results

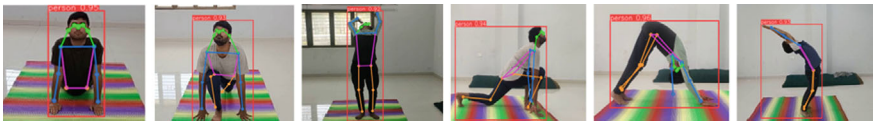


Fig. 6 YOLOv8 pose P6 results

3.3 Feature Processing

Using MoveNet as the HPE algorithm for recognizing the asana, key points are generated. A total of 17 body key points are generated, which are the nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, and right ankle. The data generated consisted of x and y coordinates along with the confidence score of each body key point.

In feature processing, the extracted key points, left ear, right ear, left eye, and right eye, are not considered in further processing. Those with confidence score less than 0.3 are removed for accurate classification. Body joint vectors are generated from

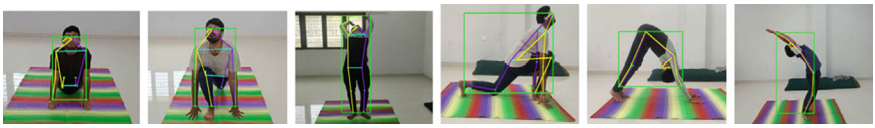


Fig. 7 MoveNet results

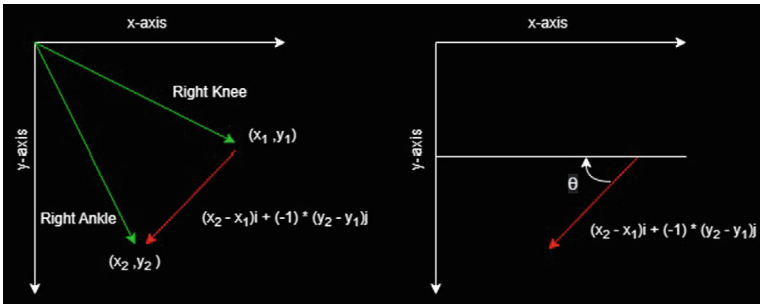


Fig. 8 Joint vector and cosine angle calculation

the 13 body key points. The vector joints are (nose, left shoulder), (left shoulder, left elbow), (left elbow, left wrist), (left shoulder, left hip), (left hip, left knee), (left knee, left ankle), (nose, right shoulder),(right shoulder, right elbow), (right elbow, right wrist), (right shoulder, right hip), (right hip, right knee), (right knee, right ankle), (left shoulder, right shoulder) and (left hip, right hip). A total of 14 body joints are generated. To generate joint vector, let us consider if right knee is given by coordinates (x_1, y_1) and right ankle is given by coordinates (x_2, y_2), then their corresponding vectors are $x_1 i + y_1 j$ and $x_2 i + y_2 j$. To get the joint that connects the two body key points, we subtract the right knee vector from right ankle vector, which gives $(x_2 - x_1)\, i + (y_2 - y_1) j$, which is a joint vector. Since, the centre origin of the image is at the top left corner, hence -1 should be multiplied to the vertical point which is $(y_2 - y_1) j$. Hence, the joint vector will be $(x_2 - x_1)\, i + (-1) * (y_2 - y_1) j$. From this joint vectors, angle with respect to horizontal axis is calculated, which is the cosine angle of the joint vector with respect to horizontal x-axis. If the joint vector is $(x_2 - x_1)\, i + (-1) * (y_2 - y_1) j$ (see Fig. 8), the corresponding cosine angle equation will be given by:

$$\cos\theta = \frac{(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_1 - y_2)^2}} \tag{1}$$

Using selected 14 body joint vectors, 14 cosine angle features are calculated, which is used for further classifications. Angle feature allows cameras to be placed at varying distance and scale, as it makes features invariant to scale and rotation of camera, that is, if two body key points, such as the left wrist and the left elbow, make an angle of 45 with the horizontal axis, it will remain the same, irrespective of distance, as the angle will always remain the same. Hence, angles are used as a key feature for accurate classification in this paper.

The 14 angle key features are further normalized using the MinMax Scaling approach for better training, and the equation is given by:

$$X_{\text{sc}} = \frac{(X - X_{\min})}{(X_{\max} - X_{\min})} \tag{2}$$

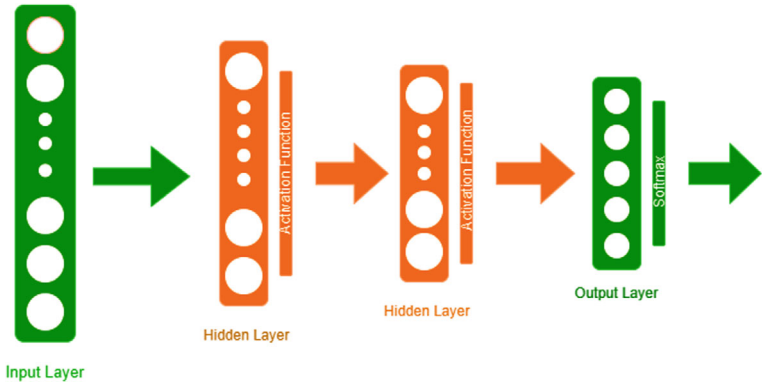


Fig. 9 Artificial neural network architecture

3.4 Classification

Using DIAT-YogaDS body key points are generated, which is processed to fetch body joint vectors, and then normalized cosine angles features, resulting to 14 key features. For the classification, we trained DNN models (see Fig. 9) with three different variants of DIAT-YogaDS, which are:

– Side-view dataset - 59,832 set of features.
– Front-view dataset - 58,776 set of features.
– Combined dataset - 118,108 set of features.

Initially, the model is trained on a raw dataset of 17 body key points, which is 34 features, without scores. The dataset is normalized, and a split is performed, where out of six different individuals, we split the train+validation to five individuals and the test to one individual. The validation split is 20%. This resulted in a train set of 89,945 features, a validation set of 9,983 features and a test set of 18180 features. The model resulted with an accuracy of 85.43%, with test 69.78 which confirmed the requirement of feature processing. The prepossessed feature set is passed through DNN. The features are categorized into five classes (Ashava Sanchalanasana, Bhujangasana, Hasta Uttanasana, Parvatasana, and Tadasana). Those data with missing or null cosine values are removed from training to achieve high accuracy. For a side-view dataset, split of five individuals for train+validation and one individual for test is performed, also the validation split is set to 20%. This resulted into a train set of 39,462 features, a validation set of 9,866 features and a test set of 9,932 features. The architecture consists of an input layer, two hidden layers of 32 neurons in first and 16 neurons second layer. LeakyReLU, activation function is utilized, as features consist of negative values, and an output dense layer of five neurons with a softmax activation function is built. Learning rate of 1-e3, Adam optimizer and Negative Log-Likelihood Loss Function is set. Class weights are passed. The model is trained for 20 epochs, resulting an accuracy of 95.74%.

Similarly, for the front-view dataset, split of five individuals for train+validation and one individual for test is performed, also the validation split is set to 20%. This resulted in a train set of 39,172 features, a validation set of 9,793 features, and a test set of 9,820 features. The architecture consisted of an input layer, two hidden layers of 32 neurons in first and 16 neurons second layer. LeakyReLU, activation function is utilized, as features consist of negative values, and an output dense layer of five neurons with a softmax activation function is built. Learning rate of 1-e3, Adam optimizer and Negative Log-Likelihood Loss Function is set. Class weights are passed. The model is trained for 20 epochs, which resulted in an accuracy of 90.83%

Finally, for combined dataset, as well, split of five individuals for train+validation and one individual for test is performed, also the validation split is set to 20%. This resulted in a train set of 89,945 features, a validation set of 9,983 features and a test set of 18180 features. The architecture consisted of an input layer, two hidden layers of 64 neurons in first and 32 neurons second layer. LeakyReLU, activation function is utilized, as features consist of negative values, and an output dense layer of five neurons with a softmax activation function is built. Learning rate of 1-e3, Adam optimizer and Negative Log-Likelihood Loss Function is set. Class weights are passed. The model is trained for 25 epochs, which resulted in an accuracy of 87.45%.

4 Results

An ablation study is carried out using different key feature extraction HPE techniques on NVIDIA GeForce GTX 1650 Ti GPU 4GB system, as shown in Table 1 and finally, MoveNet is found to be a suitable feature extraction HPE model, based on low sensitivity towards occlusion and low computational time. The author in [14] utilized OpenPose, for yoga pose estimation, and the author in [13] utilized OpenPose and Detectron2 for rehabilitation exercises, but MoveNet is comparatively 4 times more faster and has low sensitivity towards occlusion.

Table 1 HPE model results

S. no.	Human pose estimation models	Key point features	Average time taken (ms)	Sensitivity towards occlusion
1	OpenPose	135 (18 body)	498.898ms	Medium
2	DensePose	(Surface-based representation)	700.137ms	Low
3	Detectron2	17	348.371ms	High
4	YOLOv8 -Pose-P6	17	359.707ms	Medium
5	MoveNet	17	105.505ms	Low

Table 2 DNN model results

S. no.	Models	Train accuracy (%)	Test accuracy (%)
1	Simple DNN model— raw dataset	85.43	69.78
2	DNN model—side processed	95.74	94.26
3	DNN model—front processed	90.83	84.39
4	DNN model—combined processed	87.45	79.34

	precision	recall	f1-score	support
Ashava Sanchalanasana	0.99	0.99	0.99	2273
Bhujangasana	1.00	0.99	0.99	2584
Hasta Uttanasana	0.86	0.87	0.87	2327
Parvatasana	0.99	0.99	0.99	2223
Tadasana	0.88	0.87	0.87	2445
accuracy			0.94	11852
macro avg	0.94	0.94	0.94	11852
weighted avg	0.94	0.94	0.94	11852

Fig. 10 Classification report DNN model—side processed

An ablation study is also carried out with respect to, proposed DIAT-YogaDS with different views using proposed DNN models as shown in Table 2, and it is found that the dataset from side camera provided higher accuracy, and precision of 0.9428, which suggests the placement of camera on side position for HPE monitoring.

Based on the results observed, raw dataset trained model did not perform well, hence it cannot be considered for yoga posture classification. The 3 models were trained and tested, it is observed that DNN Model - side processed performed well, and it is most suitable for our DIAT-YogaDS, the classification report given by the model (see Fig. 10). DNN Model—front processed model, also gave good results with the classification report (see Fig. 11). Finally, the combined processed—DNN model, gave the classification report (see Fig. 12). Also, the precision for side processed DNN model was 0.9428.

Tests performed on DIAT-YogaDS by our proposed DNN is shown in Fig. 13a, b, and c, where key points detected by MoveNet is visualized and the yoga asana predicted by the DNN is displayed in the image.

5 Conclusion

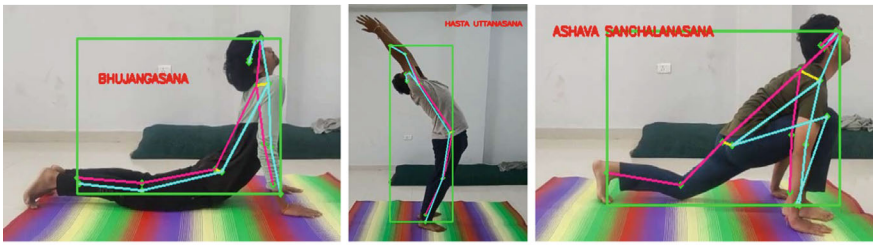
We prepared DIAT-YogaDS of five unique asana which are part of Suryanamaskar from two different angles (side-view and front-view) performed by six different individuals. The video dataset is resized and preprocessed. An analysis on HPE

	precision	recall	f1-score	support
Ashava Sanchalanasana	0.90	0.77	0.83	2253
Bhujangasana	0.90	0.96	0.93	2481
Hasta Uttanasana	0.72	0.88	0.80	1971
Parvatasana	0.83	0.83	0.83	2243
Tadasana	0.88	0.78	0.82	2433
accuracy			0.84	11381
macro avg	0.85	0.84	0.84	11381
weighted avg	0.85	0.84	0.84	11381

Fig. 11 Classification report DNN model—front processed

	precision	recall	f1-score	support
Ashava Sanchalanasana	0.85	0.83	0.84	2263
Bhujangasana	0.89	0.97	0.93	2473
Hasta Uttanasana	0.78	0.84	0.81	1969
Parvatasana	0.89	0.75	0.81	2209
Tadasana	0.84	0.84	0.84	2482
accuracy			0.85	11396
macro avg	0.85	0.85	0.84	11396
weighted avg	0.85	0.85	0.85	11396

Fig. 12 Classification report DNN model—combined processed



(a) (b) (c)

Fig. 13 Classification results—**a** Bhujangasana **b**Hasta Uttanasana **c** Ashava Sanchalanasana

models is performed on DIAT-YogaDS, and MoveNet is suitable for DIAT-YogaDS, as it exhibits results with less computational time and low sensitivity towards occlusion. The resultant body key point from MoveNet is preprocessed to obtain cosine angle of each body joint vector as it makes data, scale and rotation invariant, allowing to obtain angle features. The features obtained based on DIAT-YogaDS are trained on our proposed DNN model, and result obtained confirmed that cosine angle feature processing gave accurate results and side-view dataset trained model provided accurate results. The trained DNN model is lightweight, performs robust

detection invariant to both scale & distance and can be implemented on edge devices as part of implementing automated real-time yoga posture monitoring applications.

6 Future Work

Apart from classifying the asana, we will customize the model to provide scoring for the asana performed. This will allow the individual user to get feedback on the correctness of the asana performed. Also, a recommendation system for yoga asana based on the asana score will be worked upon, to provide suggestions for new asana and improvement in current asana.

References

1. Büssing A, Michalsen A, Khalsa SB, Telles S, Sherman KJ (2012) Effects of yoga on mental and physical health: a short summary of reviews. Evidence-based complementary and alternative medicine
2. Irani GS (2020) Injuries/harms resulting from incorrect adjustments/alignments performed by yoga asana practitioners. Arch Pharm Pract 1(1):38
3. Tensorflow. MoveNet: Ultra fast and accurate pose detection model. https://www.tensorflow.org/hub/tutorials/movenet. Last accessed on 29/01/2024
4. Cao Z, Simon T, Wei SE, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp 7291-7299)
5. Güler RA, Neverova N, Kokkinos I (2018) Densepose: dense human pose estimation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp 7297-7306)
6. Bajpai R, Joshi D (2021) Movenet: a deep neural network for joint profile prediction across variable walking speeds and slopes. IEEE Trans Instrum Meas 16(70):1–1
7. Wu Y, Kirillov A, Massa F, Lo WY, Girshick R. Detectron2. https://github.com/facebookresearch/detectron2. Last accessed on 29 Jan 2024
8. Jalal A, Sarif N, Kim JT, Kim TS (2013) Human activity recognition via recognized body parts of human depth silhouettes for residents monitoring services at smart home. Indoor Built Environ 22(1):271–9
9. Uddin MT, Uddiny MA (2015) Human activity recognition from wearable sensors using extremely randomized trees. In 2015 International conference on electrical engineering and information communication technology (ICEEICT) (pp. 1-6). IEEE
10. Shavit Y, Ferens R (2019) Introduction to camera pose estimation with deep learning. arXiv preprint arXiv:1907.05272
11. Jain S, Rustagi A, Saurav S, Saini R, Singh S (2021) Three-dimensional CNN-inspired deep learning architecture for Yoga pose recognition in the real-world environment. Neural Comput Appl 33:6427–41
12. Zhang C, Liu L, Yao M, Chen W, Chen D, Wu Y (2021) HSiPu2-a new human physical fitness action dataset for recognition and 3D reconstruction evaluation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp 3172-3181)
13. Hernández ÓG, Morell V, Ramon JL, Jara CA (2021) Human pose detection for robotic-assisted and rehabilitation environments. Appl Sci 11(9):4183

14. Anand Thoutam V, Srivastava A, Badal T, Kumar Mishra V, Sinha GR, Sakalle A, Bhardwaj H, Raj M (2022) Yoga pose estimation and feedback generation using deep learning. Comput Intell Neurosci 24:2022
15. Rajendran AK, Sethuraman SC (2023) A survey on yogic posture recognition. IEEE Access 30(11):11183–223

Next-Gen Driver Safety: An Intelligent System Framework for Multi-Factor Authentication in Distraction Detection

Sailee Pramod Shewale

Abstract Driver Distraction Detection (DDD) enhances driver safety by identifying and alerting the distraction of drivers while driving a car. However, the existing works did not concentrate on multi-factor authentication for identifying authorized, drunken, distracted, and drowsy drivers. Thus, this paper presents an effective framework for DDD. At first, the Real-Time (RT) videos of the car drivers are pre-processed. Then, Karladall Correlation You Only Look Once Version 3 (KC-YOLOV3) is used for driver detection. In phase I, the face and eye detected results using Linear Gini Scaling Viola Jones (LGSVJ) are given to Hotfix-FTS Mish-Residual Attention Network (HT-FTSM-RAN) technique, which classifies the authorized and unauthorized drivers along with the drunken and non-drunken drivers. In phase II, the authorized and non-drunken drivers are checked for drowsiness and distraction using the HT-FTSM-RAN classifier. This gives an alert signal to the drivers to prevent the risk of accidents. The proposed work gives maximum accuracy of 98.97 and 99.21% to detect authenticated and drunken drivers, respectively, and gives 99.78% accuracy to detect drowsy and distracted drivers. Thus, this framework outperformed the existing methodologies.

Keywords Driver distraction detection framework · Driver safety · Intelligent Systems · Multi-factor authentication · Technology

1 Introduction

Why is advanced technology not ensuring driver and road safety? Technology is known to solve many of our problems, but does it also create new problems? As technology becomes more integrated into cars, driving has become more convenient; however, it also surfaces a serious issue of driver distraction [1]. Accidents resulting from distracted driving are increasingly common and carry grave consequences [2],

S. P. Shewale (✉)
Symbiosis Institute of Design (SID), Symbiosis International (Deemed University) (SIU), Viman Nagar, Pune, Maharashtra, India
e-mail: sailee.shewale@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_54

exacerbated by instances of driving under the influence of alcohol [3]. To address these risks, Driver Distraction Detection (DDD) technology has been developed [4]. DDD systems issue warnings through sound or visual signals to alert drivers when distractions are detected.

Existing DDD methods employ Deep Learning (DL) techniques like Convolutional Neural Networks (CNN) [5] and Optimized Residual Neural Networks (O-RNN) [6]. Other methods, such as Support Vector Machine (SVM) [7] and Long Short-Term Memory (LSTM) [8], are also used to alert the DDD system. Various Machine Learning (ML) approaches are employed for DDD [9], yet these methods often lack multi-factor authentication capabilities.

To address this gap, the proposed work utilizes advanced approaches such as You Only Look Once Version 3 (YOLO-V3), Linear Gini Scaling Viola Jones (LGSVJ), and Hotfix-FTS Mish-Residual Attention Network (HT-FTSM-RAN). These methods aim to efficiently detect authorized, drowsy, intoxicated, and distracted drivers, enhancing the effectiveness of DDD systems.

1.1 Research Gaps

The gaps present in existing literature are mentioned as follows:

- Firstly, none of the existing works have prioritized multi-factor authentication within Driver Distraction Detection (DDD) systems. This oversight fails to address issues such as car theft and driving under the alcohol influence simultaneously.
- Secondly, the current literature neglects incorporating driver drowsiness into the definition of distraction [10], overlooking a critical aspect of driver behavior.
- Thirdly, the extraction of irrelevant features as observed in a specific study [11] has been found to diminish the accuracy of DDD systems, highlighting a technical limitation.
- Lastly, prior research efforts grappling with biometric and authentication factors have encountered challenges in accurately identifying the authenticated car owner, revealing a significant hurdle in ensuring system reliability and security.

1.2 Research Objectives

The objectives of the proposed study are delineated as follows:

- Firstly, the primary aim is to address the identified deficiencies by emphasizing multi-factor authentication within Driver Distraction Detection (DDD) systems.
- Secondly, the proposed methodology seeks to broaden the definition of distraction to include driver drowsiness. This will be achieved through the implementation of facial point extraction techniques and analysis of eye and mouth aspect ratios.

- Thirdly, an innovative feature extraction process will be employed, focusing on skeletonized image analysis, with the objective of enhancing classification accuracy.
- Finally, to ensure precise authentication, advanced techniques such as KC-YOLOV3 and LGSVJ will be utilized for the detection of the driver's eyes and facial features. These methodologies are chosen for their demonstrated effectiveness in accurately identifying and classifying key facial attributes.

The rest of the paper firstly addresses the study of related literature; secondly, explains the proposed methodology; then, the results and discussion are presented; and lastly, the proposed work is concluded with future recommendations.

2 Literature Review

2.1 Need of DDD

Data suggests the importance of a trustworthy real-time driver distraction system, capable of monitoring and alerting drivers to distractions before accidents occur, reducing roadway disaster expenses [12]. Machine learning and deep learning can be instrumental in identifying diverse distracted behaviors such as cell phone use, conversation, eating, sleeping, or driver inattentiveness [13]. The driver's attention buffer diminishes when they look away and replenishes upon returning gaze; AttenD2.0 enhances this by incorporating multiple buffers, transparently integrating situational dependence and visual multitasking behavior [14]. This article addresses these issues by proposing a deep learning method that correlates driver distraction with bioelectric electroencephalography EEG signals unaffected by traffic conditions, suitable for integration into driver assistance and autonomous vehicle systems for managing emergencies [15].

2.2 Driver and DDD

Globally increasing road accidents highlight driver distraction as a major factor. National Highway Traffic Safety Administration research suggests that distractions like mobile use, eating, and operating in-vehicle systems contribute to accidents [16]. Current distracted driving models have limited generalization. This study aims to enhance their generalization by considering factors like the driver, background, monitoring angle, etc. It introduces a new method called multi-scale domain adaptation network (MSDAN) to enhance model adaptability [17]. Lately, car manufacturers have been incorporating systems to detect driver distraction. This study proposes a system for detecting mobile phone usage in images captured from inside

vehicles [18]. The emergence of new technology has significantly increased distractions from handheld devices, contributing to traffic accidents by impairing drivers' decision-making and reducing their reaction time. Multitasking further diminishes attentiveness, leading to avoidable accidents [19].

2.3 Vehicle Technology and DDD

The placement of cameras in order to always capture the driver's face and the in-vehicle microphone to capture voice and sound was proposed for improved accuracy for driver distraction detection [20]. Driver distraction is increasing due to the widespread use and complexity of in-vehicle technology and portable devices, posing a significant threat to road safety as distractions and inattention are leading causes of accidents [21].

2.4 DDD with Intelligent Systems

A Hybrid-DL approach for mitigating driver distraction was presented [10]. This study analyzed driver actions as well as head pose and provided voice commands for distraction elimination. Although this methodology effectively classified the driver's actions, it didn't encompass the detection of retinal abnormalities for DDD. A lightweight Octave-Like Convolutional Mixed block Network (OLCM-Net) for DDD was accomplished [11]. For expanding feature maps, OLCMNet employed low-frequency as well as high-frequency branches. Detailed information for alerting distracted drivers was captured by the high-frequency branches. Nevertheless, the processing speed of OLCM was reduced owing to the addition of more low-frequency branches. A 2-stage approach for the detection of distracted drivers was formulated [22]. Initially, Transfer Learning (TL) and fine-tuning models were applied to vision-centered CNN methodologies. Then, for accurate DDD, LSTM-Recurrent Neural Networks (LSTM-RNN) was used. Nevertheless, this methodology didn't detect drowsiness as a distraction, thus hindering the system's behavior. An automatic DDD system by employing Deep-CNN was established [13]. Four TL-centered architectures like CNN, Visual Geometry Graph (VGG-16), Residual Network-50 (ResNet50), and Mobile Network-Version2 (MobileNetV2) were utilized in this methodology for identifying distraction. However, this work didn't alert the distracted drivers, which caused higher accidental risks. A CNN framework named DistractNet for the effective DDD was introduced [23]. Initially, DistractNet was trained on a dataset that captured diverse distracted driving activities and then was tested to accurately classify drivers into diverse distractions. However, the entire process was degraded by the high error rate in the DistractNet framework.

3 Proposed Methodology of Multi-factor Authentication for Safe Driving

Multi-factor authentication using the HT-FTSM-RAN classifier is done in the proposed work to detect authorized, drunken, drowsiness, and distracted drivers for safe driving. Figure 1 illustrates the proposed work's structural diagram.

3.1 Video Processing

Initially, the k number of RT videos (V) captured after the drivers enter the car is denoted as,

$$V = V^1, V^2, ...V^k \tag{1}$$

The (V) is then converted to frames (V_f), and the n number of keyframes (V_{key}) extracted from (V_f) is given as,

$$V_{\text{key}} = V_{\text{key}}^1, V_{\text{key}}^2, ...V_{\text{key}}^n \tag{2}$$

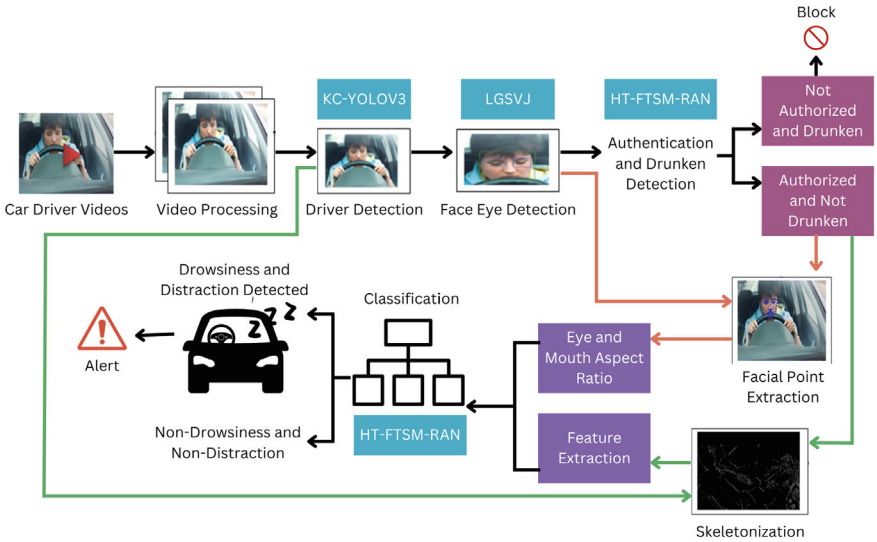


Fig. 1 Structural diagram of the proposed work

After that, the noise from is removed and then contrast is enhanced to improve the visual clarity. The $\left(k'\right)$ numbers of $(V_{\text{constract}})$ are signified as,

$$V_{\text{contrat}} = V^{1}_{\text{constract}}, V^{2}_{\text{contrat}}, \ldots\ldots V^{k'}_{\text{constract}} \tag{3}$$

By using KC-YOLOV3, the drivers $(D_{d'})$ are detected from the enhanced images $(V_{\text{constract}})$.

3.2 Driver Detection

For faster detection, the $(D_{d'})$ is detected using You Only Look Once Version 3 (YOLOV3) after video processing. However, the softmax activation struggles in predicting the humans whenever the image pixels are non-linearly separable. Thus, Karladall Correlation (KC)-based softmax activation, which effectively predicts humans even in non-linear pixel images, is used. The KC-YOLOV3 algorithm is described below,

- Initially, $(V_{\text{constract}})$ is partitioned into grids (Y).
- Then, centered on certain parameters, the bounding box $\left(B'\right)$ is predicted for each (Y). Therefore, the parameters for each $\left(B'\right)$ in the cell are given as,

$$B' = m, n, o, p \tag{4}$$

$$m = \sigma(f_m) + k_m \tag{5}$$

$$n = \sigma(f_n) + k_n \tag{6}$$

$$o = u'_o e^{(f_o)} \tag{7}$$

$$p = u'_p e^{(f_p)} \tag{8}$$

where, the centers of $\left(B'\right)$ relative to (Y) are represented as (m, n), the predicted offsets of (m, n) are represented as (f_m, f_n), (k_m, k_n) represents the top-left corner of (m, n) in (Y), the width and height of $(B\prime)$ relative to the entire image are represented as (o, p), $\left(f_o, f_p\right)$ represents the predicted scales for $(B\prime)$ dimensions, the width and height of the anchor box (B'^{A}) are represented as $\left(u'_o, u'_p\right)$, (e) represents the exponential function that modifies the dimensions of the anchor box (B^{a}), and (σ) represents the sigmoidal function, which is computed as,

$$\sigma\left(B'\right) = \frac{1}{1 + e^{-B'}} \tag{9}$$

- Class probabilities $\left(C_p\right)$ for each $\left(B'\right)$ are predicted as,

$$C_p = \delta(C_1, C_2.....C_{l'}) \tag{10}$$

Here, $\left(p = 1 \text{ to } l'\right)$ signifies the class scores, and the softmax activation is represented as δ. In this, KC-centered δ is employed, which is given as,

$$\delta\left(C_p\right) = \frac{\sqrt{\sum\left(e^{C_p} - e^{\overline{C}_p}\right)}}{\sqrt{\sum\left(e^{C_p} - e^{\overline{C}_p}\right)^2}} \tag{11}$$

- Now, the confidence score (C_s) is predicted for each $(B\prime)$ centered on the representation of $\left(C_p\right)$ and inverse of $\left(\overline{C}_p\right)$. This is articulated as,

$$C_s = \sigma\left(f_{C_s}\right) \tag{12}$$

Here, the sigmoidal function with predicted score values is represented as $\sigma\left(f_{C_s}\right)$.

Pseudo-code of KC-YOLOV3

Input: Enhanced images, $\left(V_{contrast}\right)$

Output: Detected drivers, $\left(D_{d'}\right)$

Begin

Initialize iterations, $t, t^{\max}$

While $t < t^{\max}$

Divide grid cell, $\left(Y\right)$

Predict $\left(B'\right)$ for each $\left(Y\right)$

Calculate $C_p = \delta\left(C_1, C_2 \ldots\ldots C_{l'}\right)$

Compute $\left(C_s\right)$

End while

Return $\rightarrow \left(D_{d'}\right)$

End

After that, for efficient authentication and identification of drunken $(D_{d'})$, the face and eyes of $(D_{d'})$ are identified.

3.3 Face and Eye Detection

By utilizing LGSVJ, the face and eyes of $(D_{d'})$ are identified. Viola Jones (VJ) is a within cascade object detector, which is known for its rapid identification of faces. In this, the weights are randomly initialized, which affect the prediction accuracy. Thus, the Linear Gini Scaling (LGS) centered weights are initialized that accurately detect the driver's face and eyes. The working steps of LGSVJ are detailed below,

Haar-Like Feature Selection. Primarily, to capture patterns like edges, lines, and corners in the image, haar-like features are selected from $(D_{d'})$. It is articulated as,

$$H^f = \sum_{g'} v_{g'} * \ell_{g'}\left(j', k'\right) \tag{13}$$

where, the haar-like feature value at position $\left(j', k'\right)$ is represented as $\left(H^f\right)$, g' refers to the index representing the individual $\left(H^f\right)$, the weight in the rectangular region is signified as $\left(v_{g'}\right)$, and the pixel intensity is notated as $\left(\ell_{g'}\right)$.

Integral Image Generation. To accelerate the computation, the integral image $\left(I^f\right)$ is generated from $\left(H^f\right)$,

$$I^f = \sum_{J=1}^{j'} \sum_{K=1}^{k'} I^f\left(j', k'\right) \tag{14}$$

Here, the feature points of pixel intensities within the rectangular region are represented as (J and K).

AdaBoost Calculation. Weak classifiers are combined by AdaBoost to form a strong classifier. The weights are initialized utilizing LGS rather than random initialization to obtain accurate outcomes, which are computed as,

$$A^b\left(I^f\right) = 1 - \sum_{e'=1}^{E} \left(w_e b_{e'}\left(I^f\right)\right)^2 \tag{15}$$

where, the final strong classifier is signified as $A^b\left(I^f\right)$, which determines whether a given region of the image contains a face or not, the LGS weights in the e'^{th} weak classifier are symbolized as $w_{e'}$, and the number of weak classifiers is notated as $b_{e'} \rightarrow$ where $e' = 1$ to E.

Haar-Cascade Classifier. The chosen H^f is organized into a cascade of classifiers in which each classifier is trained to detect non-face and non-eye regions rapidly. Therefore, the final cascade classifier (F_c) is given as,

$$F_c = \sum_{i'=1}^{n} w_i * H^s\left(j', k', s'\right) \tag{16}$$

where, the number of stages in cascade is notated as $(i' = 1 \text{ to } n)$, the i'^{th} weighted function is represented as (w_i), and $H^s(j', k', s')$ represents the decision function of the i'^{th} stage at the position (j', k') with scale factor (s'). Finally, $(D_{\text{face,eye}})$ denotes the detected face and eye regions.

3.4 Phase I

In Phase I, by using HT-FTSM-RAN, the authorized and drunken drivers are detected from $(D_{\text{face,eye}})$. Residual Attention Network (RAN) encompasses a soft mask branch, which extracts many irrelevant features, thereby making the process complex and slow. Hence, the soft mask branch with softmax activation is replaced with the Hotfix branch with FTS Mish activation, which concentrates on important features, thus making faster detection. Figure 2 exhibits the HT-FTSM-RAN classifier.

The working of HT-FTSM-RAN is detailed below,

Step 1: Initialization. Initially, for detecting the authorized person, the $(D_{\text{face,eye}})$ is given as input, which is signified as,

$$D_{\text{face,eye}} = D_1, D_2, ...D_{v'} \text{ where } \quad \text{face,eye} = 1 \text{ to } v' \tag{17}$$

Here, the total number of $(D_{\text{face,eye}})$ is notated as (v').

Step 2: Attention Network. The RAN generally comprises 2 branches, namely the trunk branch and the mask branch.

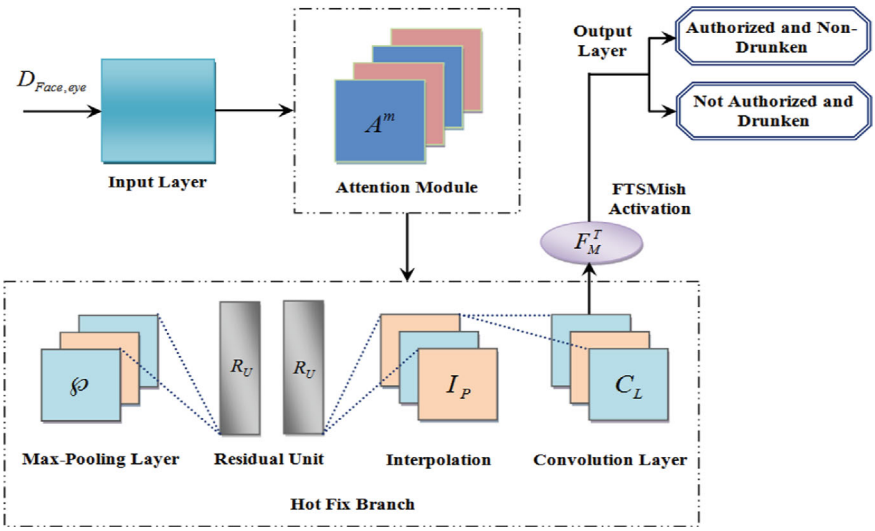


Fig. 2 HT-FTSM-RAN classifier

- The trunk branch $\left(T\left(D_{\text{face,eye}}\right)\right)$ is the upper branch with a pre-defined number of residual units (R_u) to extract features.
- To learn a mask, the mask branch $\left(M\left(D_{\text{face,eye}}\right)\right)$ uses a bottom-up top-down structure. Therefore, the Attention module (A^m) is articulated as,

$$A^m_{u,c}\left(D_{\text{face,eye}}\right) = M_{u,c}\left(D_{\text{face,eye}}\right) * T_{u,c}\left(D_{\text{face,eye}}\right) \tag{18}$$

where, $(u = 1 \text{ to } U)$ represents the range of spatial locations in (A^m) and the channel index is notated as $(c = 1 \text{ to } C)$. The parameters of the trunk (φ) and mask (φ) are updated during backpropagation as,

$$\frac{\partial M\left(D_{\text{face,eye}}, \phi\right) * T\left(D_{\text{face,eye}}, \varphi\right)}{\partial \varphi} = M\left(D_{\text{face,eye}}, \phi\right)\frac{\partial T\left(D_{\text{face,eye}}, \varphi\right)}{\partial \varphi} \tag{19}$$

Here, the differential calculation for (φ) and (φ) is signified as $(\partial\varphi, \partial\varphi)$.

Step 3: Attention Residual Learning. Attention Residual Learning assists in keeping the significant feature from (A^m), which is given as,

$$A^m_{u,q}\left(D_{\text{face,eye}}\right) = \left(1 + M_{u,q}\left(D_{\text{face,eye}}\right)\right) * F_{u,q}\left(D_{\text{face,eye}}\right) \tag{20}$$

Here, the original features obtained from $\left(D_{\text{face,eye}}\right)$ are represented as $F_{u,q}\left(D_{\text{face,eye}}\right)$.

Step 4: Hotfix Branch. The Hotfix branch extracts significant features and makes faster classification. This is computed by,

$$\wp\left(D_{\text{face,eye}}\right) = \text{avg}\left(\gamma\left(D_{\text{face,eye}}\right)\right) \oplus \max\left(\gamma\left(D_{\text{face,eye}}\right)\right) \tag{21}$$

where, the pooled output is represented as $\wp\left(D_{\text{face,eye}}\right)$ and (avg, max) represents the average and maximum values for $\left(D_{\text{face,eye}}\right)$. In this, FTS-Mish activation $\left(F^T_M\right)$ is used to attain the self-residual spatial attention matrix (A_s),

$$F^T_M = \frac{\sigma\left(D_{\text{face,eye}}\right) + \log\left(1 + e^{-D_{\text{face,eye}}}\right)\left(D_{\text{face,eye}}\right)\left(1 - \sigma\left(D_{\text{face,eye}}\right)\right)}{d\left(D_{\text{face,eye}}\right)} \tag{22}$$

Here, the standard deviation and differential calculation of $\left(D_{\text{face,eye}}\right)$ are notated as (σ),$d\left(D_{\text{face,eye}}\right)$, respectively.

Step 5: Output Layer. The authorized $\left(A^{\text{authorized}}\right)$ and non-authorized persons $\left(A^{\text{non - authorized}}\right)$ are classified from (A_s). If the person is not authorized, then an alert is sent to the owner who can either accept or decline the request for providing access for $\left(A^{\text{non - authorized}}\right)$ to utilize the vehicle.

- If the owner accepts the request, then the new user is redirected for registration.
- If the owner declines the request, then the person is not allowed to access the car. Therefore, car theft can be prevented or notified.

- If the authentication is successful, then the drunken detection system is enabled to verify whether the accessing person is drunk or not, to prevent accidents.
- If the person is classified as drunken $\left(F^{\text{drunk}}\right)$, then he/she is not allowed to access the car. Or else, the person is allowed to access the car followed by drowsiness and distraction detection.

3.5 Phase II

Here, by using HT-FSTM-RAN, the drowsiness and distraction of drivers are detected.

- **Drowsiness Detection.** For analyzing drowsiness, the haar features $\left(F_{ps}\right)$ are extracted from $\left(D_{\text{face,eye}}\right)$. Then, the mouth and eye aspect ratios (F_{ratio}) are taken and given to HT-FSTM-RAN. This detects and classifies the drowsiness drivers (F_{drows}) and non-drowsiness drivers $(F_{\text{non - drows}})$ and immediately sends alert (ψ) to (F_{drows}) for safe driving. The working steps of HT-FTSM-RAN are discussed in (Sect. 3.4).
- **Distraction Detection.** At the same time, the detected drivers are skeletonized (S_{skeleton}) from $(D_{d'})$ for enhancing the RT analysis in detecting distracted drivers (F_{distract}). After that, features like motion features, trajectory features, and depth features are extracted and given to HT-FSTM-RAN, which is shown in Sect. 3.4. This classifies the types of (F_{distract}) and sends (ψ) to (F_{distract}) for safe and secure driving. From this approach, the authentication, drunken drivers, drowsiness, and distracted drivers are identified and notified effectively for safe driving as well as averting unnecessary road accidents.

4 Results and Discussion

To highlight the proposed DDD model's efficacy, the proposed model's performance is analogized with other traditional works. The proposed work is implemented using the PYTHON platform.

4.1 Dataset Description

The drowsiness detection dataset utilized for training the proposed model comprises 1,00,000 images with 12 classes, namely: normal driving, texting—right, talking on the phone—right, texting—left, talking on the phone—left, operating the radio, drinking, reaching behind, hair and makeup, talking to passenger, drowsiness, and non-drowsiness. Likewise, the drunken driver detection dataset encompasses 2 classes, namely drunken and non-drunken; there are 1,00,000 images for each

class. To ensure robust model training and evaluation, the proposed methodology employs an 80–20 split, allocating 80% of the dataset for model training and the remaining 20% for testing purposes. This partitioning strategy enables comprehensive model training while providing a rigorous evaluation framework to assess the model's performance on unseen data.

4.2 Performance Analysis

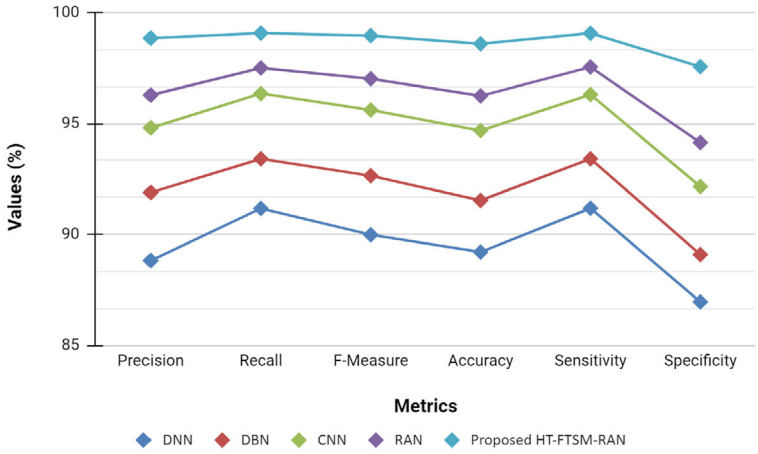
Table 1 shows the image results of the proposed work for image enhancement, driver detection, face and eye detection, skeletonization, and facial point extraction.

The performance comparison of each phase (KC-YOLOV3, LGSVJ, and HT-FTSM-RAN) with prevailing techniques is discussed further.

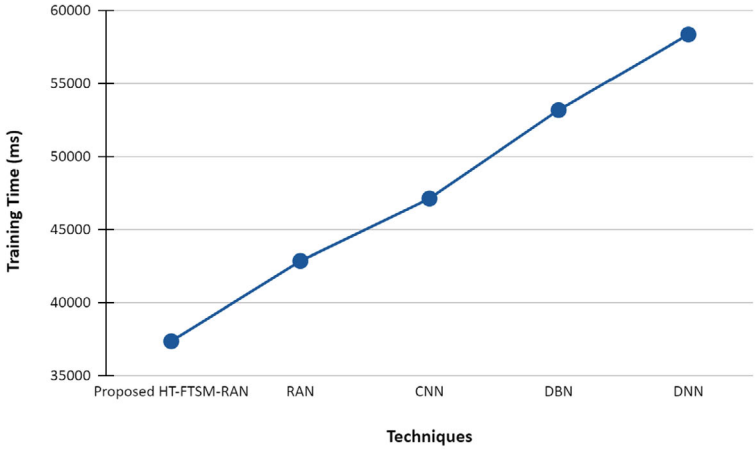
The comparative performance analysis of the proposed HT-FTSM-RAN model against established techniques, including RAN, CNN, Deep Belief Network (DBN), and Deep Neural Network (DNN), is depicted in Fig. 3a and b.

Table 1 Image results of the proposed framework

Input Image	Enhanced Image	Driver Detection	Face and Eye Detection (Drunken)	Skeletonization (DDD)	Facial Point Extraction (Drowsiness)
			-		-
			-		-
			-		-
			-		-
				-	-
				-	



(a)



(b)

Fig. 3 Performance analysis based on **a** Evaluation metrics and **b** Training time

The proposed HT-FTSM-RAN model, incorporating the FTS-Mish Activation function rather than softmax activation, demonstrated superior performance metrics. Specifically, the HT-FTSM-RAN achieved an accuracy of 98.5974%, precision of 98.8505%, recall of 99.0723%, specificity of 97.5609%, sensitivity of 99.0712%, and f-measure of 98.9643%, with a minimal Training Time (TT) of 37,348 ms.

In contrast, the existing methodologies exhibited longer Training Times (TT) while achieving lower accuracies, with maximal TT observed. These techniques yielded accuracies of 96.2441%, 94.6791%, 91.5328%, and 89.2018%, respectively, among RAN, CNN, DBN, DNN, along with diminished precision, recall, specificity,

sensitivity, and f-measure scores. This disparity underscores the proposed HT-FTSM-RAN model's superior performance compared to prevailing methodologies.

Table 2 presents a comparative performance analysis between the proposed KC-YOLOV3 and several established approaches, including YOLOV3, Faster Region-CNN (Faster R-CNN), R-CNN, and CNN.

The results indicate that the proposed KC-YOLOV3 model achieved superior performance metrics compared to existing methodologies. Notably, the proposed approach attained an average precision of 98.3658%, mean average precision of 98.6574%, recall of 98.7485%, and f1-score of 98.8475%. This outperformance can be attributed to the utilization of KC-based softmax activation within the YOLOV3 framework, facilitating accurate driver identification.

These findings underscore the effectiveness of the proposed KC-YOLOV3 model in enhancing driver detection accuracy, thus contributing to advancements in driver safety technology.

Figures 4a–c present a comprehensive comparison of key performance metrics, including True Positive Rate (TPR), True Negative Rate (TNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), False Positive Rate (FPR), and False Negative Rate (FNR), between the proposed HT-FTSM-RAN model and existing methodologies.

The HT-FTSM-RAN model exhibited superior performance across multiple metrics, achieving a TPR of 99.0783%, TNR of 97.5609%, PPV of 98.8584%, NPV of 98.0336%, and demonstrating a lower FNR of 2.4391% and FPR of 0.9216%. In contrast, existing methodologies yielded a lower average TPR of 94.62%, TNR of 90.58%, PPV of 93.02%, and NPV of 92.70%, along with elevated FPR and FNR rates.

The observed effectiveness of the proposed HT-FTSM-RAN model can be attributed to the integration of the Hotfix branch and FTSM activation, which contributed to enhanced performance compared to traditional methods.

These findings underscore the efficacy of employing advanced techniques in improving the accuracy and reliability of driver distraction detection systems, thereby promoting enhanced road safety.

Table 3 provides a comparative analysis between the proposed HT-FTSM-RAN model and several established methodologies. While the proposed approach adopts

Table 2 Performance assessment of the proposed KC-YOLOV3

Techniques	Average Precision (%)	Mean Average Precision (%)	Recall (%)	F1-Score (%)
Proposed KC-YOLOV3	98.3658	98.6574	98.7485	98.8475
YOLOV3	96.4841	96.3258	96.3254	96.3247
Faster R-CNN	94.2145	94.2154	94.2147	94.2748
R-CNN	92.3247	92.3695	92.3258	92.2335
CNN	90.3247	90.2347	90.3248	90.1475

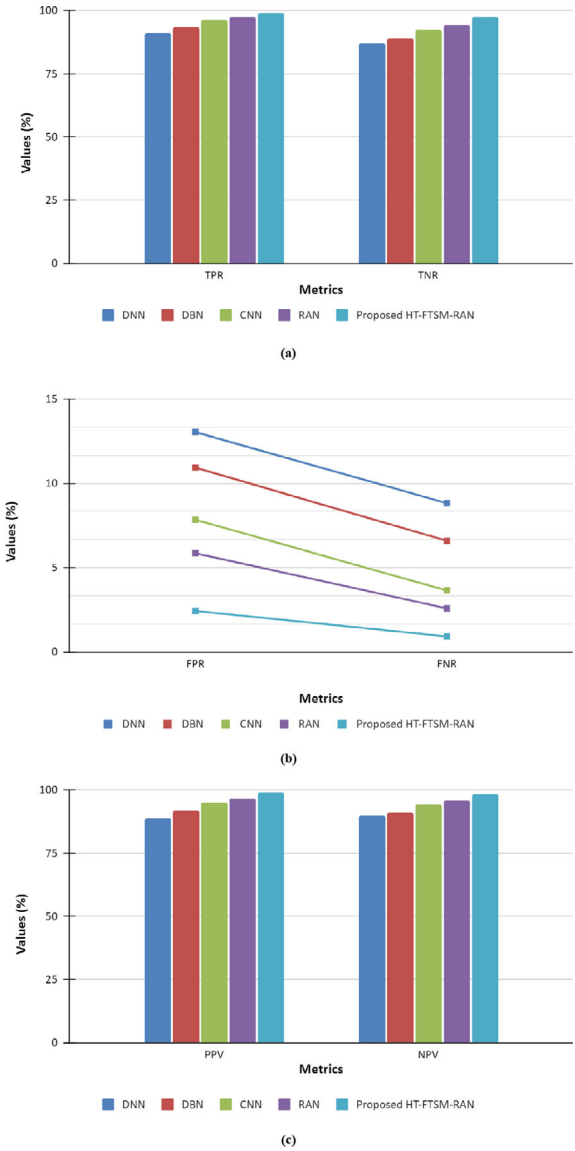


Fig. 4 Graphical comparison based on **a** TPR and TNR, **b** FPR and FNR, and **c** PPV and NPV

the HT-FTSM-RAN methodology, existing techniques include You Only Look Once (YOLO), VGG-19, Unsupervised Multi-Modal Fusion Network (UMMFN), ResNet Support Vector Machine (ReSVM), and Decreasing-HOG-based CNN (D-HCNN).

The proposed HT-FTSM-RAN model achieved a notably higher accuracy of 98.59%, compared to the lower accuracies ranging from 82 to 97.73% attained by prevailing techniques. Specifically, YOLO and UMMFN demonstrated lower f1-scores of 74% and 97.39%, respectively, while the proposed HT-FTSM-RAN model

Table 3 Comparison analysis of the proposed and related works

Study	Techniques Used	F1-score (%)	Accuracy
Proposed Work	HT-FTSM-RAN	98.96	98.59
[24]	YOLO	74	82
[25]	VGG-19	–	95.77
[26]	UMMFN	97.39	97.47
[27]	ReSVM	–	95.50
[28]	D-HCNN	–	97.73

achieved a superior f1-score of 98.96%.The observed superiority of the proposed framework can be attributed to the utilization of FTSM activation within the Residual Attention Network (RAN), facilitating faster and more accurate classification of authorized, drunken, drowsy, and distracted drivers. These results underscore the effectiveness of the proposed framework in surpassing existing methodologies in classifying authorized, drunken, drowsy, and distracted drivers, thereby contributing to advancements in driver safety technology.

5 Conclusion

This approach effectively detected the authorized drivers and gave alert signals to drunken, drowsy, and distracted drivers. RT videos were collected to detect and alert. After that, the drivers were detected by the KC-YOLOV3 with average precision and mean average precision of 98.3658% and 98.6574%, respectively. Lastly, the Hotfix branch with FTS-Mish is utilized in HT-FTS-RAN, which detected and classified drunken, drowsy, and distracted drivers with an accuracy rate of 98.5974% and a recall of 99.0723%. Therefore, this proposed framework assisted in reducing the accidental risk of authorized drivers and unauthorized access or theft of vehicles.

5.1 Future Recommendation

This study completely depended on camera devices that generated vulnerability in repaired cases. Therefore, to augment the detection of distracted and drunken drivers, additional sensor devices and voice assistants will be added in the future by utilizing numerous approaches.

Acknowledgements The author would like to thank Ms Stancy for her assistance in executing this paper's methodology.

References

1. Kashevnik A, Shchedrin R, Kaiser C, Stocker A (2021) Driver distraction detection methods: a literature review and framework. IEEE Access 9:60063–60076. https://doi.org/10.1109/ACCESS.2021.3073599
2. Ju Y, Chen F, Li X, Lin D (2023) Bibliometric study and critical individual literature review of driving behavior analysis methods based on brain imaging from 1993 to 2022. J Traffic Transp Eng (Engl Ed). https://doi.org/10.1016/j.jtte.2023.07.004
3. Halin A, Verly JG, Van Droogenbroeck M (2021) Survey and synthesis of state of the art in driver monitoring. Sensors 21:5558. https://doi.org/10.3390/s21165558
4. Vismaya UK, Saritha E (2020, July) A review on driver distraction detection methods. In: 2020 International conference on communication and signal processing (ICCSP). IEEE, pp 483–487. https://doi.org/10.1109/ICCSP48568.2020.9182316
5. Janet B, Reddy US (2020, August) Real-time detection of driver distraction using CNN. In: 2020 3rd International conference on smart systems and inventive technology (ICSSIT). IEEE, pp 185–191. https://doi.org/10.1109/ICSSIT48917.2020.9214233
6. Abbas T, Ali SF, Khan AZ, Kareem, I (2020, November) OptNet-50: An Optimized Residual Neural Network Architecture of Deep Learning for Driver's Distraction. In: 2020 IEEE 23rd international multitopic conference (INMIC). IEEE, pp 1–5. https://doi.org/10.1109/INMIC50486.2020.9318087
7. Chaudhary V, Dalwai Z, Kulkarni V (2021, June) Intelligent distraction and drowsiness detection system for automobiles. In: 2021 International conference on intelligent technologies (CONIT). IEEE, pp 1–4. https://doi.org/10.1109/CONIT51480.2021.9498562
8. Alkinani MH, Khan WZ, Arshad Q (2020) Detecting human driver inattentive and aggressive driving behavior using deep learning: recent advances, requirements and open challenges. IEEE Access 8:105008–105030. https://doi.org/10.1109/ACCESS.2020.2999829
9. Soultana A, Benabbou F, Sael N, Ouahabi S (2022) A systematic literature review of driver inattention monitoring systems for smart car. Int J Interact Mob Technol 16. https://doi.org/10.3991/ijim.v16i16.33075
10. AbouOuf A, Sobh I, Nasser M, Alsaqa O, Elezaby O, Zaki JF (2022) Multimodel system for driver distraction detection and elimination. IEEE Access 10:72458–72469. https://doi.org/10.1109/ACCESS.2022.3188715
11. Li P, Yang Y, Grosu R, Wang G, Li R, Wu Y, Huang Z (2021) Driver distraction detection using octave-like convolutional neural network. IEEE Trans Intell Transp Syst 23:8823–8833. https://doi.org/10.1109/TITS.2021.3086411
12. Taherisadr M, Asnani P, Galster S, Dehzangi O (2018) ECG-based driver inattention identification during naturalistic driving using Mel-frequency cepstrum 2-D transform and convolutional neural networks. Smart Health 9:50–61. https://doi.org/10.1016/j.smhl.2018.07.022
13. Hossain MU, Rahman MA, Islam MM, Akhter A, Uddin MA, Paul BK (2022) Automatic driver distraction detection using deep convolutional neural networks. Intell Syst Applications 14:200075. https://doi.org/10.1016/j.iswa.2022.200075
14. Ahlström C, Georgoulas G, Kircher K (2021) Towards a context-dependent multi-buffer driver distraction detection algorithm. IEEE Trans Intell Transp Syst 23:4778–4790. https://doi.org/10.1109/TITS.2021.3060168
15. Li G, Yan W, Li S, Qu X, Chu W, Cao D (2021) A temporal–spatial deep learning approach for driver distraction detection based on EEG signals. IEEE Trans Autom Sci Eng 19:2665–2677. https://doi.org/10.1109/TASE.2021.3088897
16. Mittal H, Verma B (2023) CAT-CapsNet: a convolutional and attention based capsule network to detect the driver's distraction. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2023.3266113
17. Wang J, Wu Z (2023) Driver distraction detection via multi-scale domain adaptation network. IET Intel Transport Syst 17:1742–1751. https://doi.org/10.1049/itr2.12366

18. Roy K (2022) Unsupervised sparse, nonnegative, low rank dictionary learning for detection of driver cell phone usage. IEEE Trans Intell Transp Syst 23:18200–18209. https://doi.org/10.1109/TITS.2022.3151752
19. Jain DK, Jain R, Lan X, Upadhyay Y, Thareja A (2021) Driver distraction detection using capsule network. Neural Comput Appl 33:6183–6196. https://doi.org/10.1007/s00521-020-05390-9
20. Tran D, Do HM, Lu J, Sheng W (2020, October) Real-time detection of distracted driving using dual cameras. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2014–2019. https://doi.org/10.1109/IROS45743.2020.9340921
21. Gusain A, Rawat AS, Bohra M, Kumar I, Singh T (2023, July) Distracted driver detection and driver rating system using deep learning. In: 2023 World conference on communication & computing (WCONF). IEEE, pp 1–6. https://doi.org/10.1109/WCONF58270.2023.10235211
22. Omerustaoglu F, Sakar CO, Kar G (2020) Distracted driver detection by combining in-vehicle and image data using deep learning. Appl Soft Comput 96:106657. https://doi.org/10.1016/j.asoc.2020.106657
23. Nasri I, Karrouchi M, Snoussi H, Kassmi K, Messaoudi A (2022) DistractNet: a deep convolutional neural network architecture for distracted driver classification. IAES Int J Artif Intell 11:494. https://doi.org/10.11591/ijai.v11.i2.pp494-503
24. Li L, Zhong B, Hutmacher C Jr, Liang Y, Horrey WJ, Xu X (2020) Detection of driver manual distraction via image-based hand and ear recognition. Accid Anal Prev 137:105432. https://doi.org/10.1016/j.aap.2020.105432
25. Ezzouhri A, Charouh Z, Ghogho M, Guennoun Z (2021) Robust deep learning-based driver distraction detection and classification. IEEE Access 9:168080–168092. https://doi.org/10.1109/ACCESS.2021.3133797
26. Zhang Y, Chen Y, Gao C (2021) Deep unsupervised multi-modal fusion network for detecting driver distraction. Neurocomputing 421:26–38. https://doi.org/10.1016/j.neucom.2020.09.023
27. Abbas T, Ali SF, Mohammed MA, Khan AZ, Awan MJ, Majumdar A, Thinnukool O (2022) Deep learning approach based on residual neural network and SVM classifier for driver's distraction detection. Appl Sci 12:6626. https://doi.org/10.3390/app12136626
28. Qin B, Qian J, Xin Y, Liu B, Dong Y (2021) Distracted driver detection based on a CNN with decreasing filter size. IEEE Trans Intell Transp Syst 23:6922–6933. https://doi.org/10.1109/TITS.2021.3063521

Dermatophytic Dynamics: A Holistic Approach to Superficial Mycosis Research

S. Abhishek, T. Anjali, and Rina Barouch Bentov

Abstract Human fungus infections are a serious public health issue that needs a prompt and correct diagnosis to cure effectively. Human fungal infections have historically been treated with in-person consultations and exams by trained laboratory mycologists. Lately, deep learning models-based computer-aided diagnosis systems have become more potent instruments, especially for late-stage mycological diagnoses. This research presents experimental results using a deep learning technique to classify different species of fungi. The work is noteworthy for using a vision transformer, demonstrating its effectiveness in classifying fungi. This work contributes to the ongoing advancement of diagnostic techniques and provides a feasible route for artificial intelligence integration in mycology. The results show a remarkable 97.3% accuracy rate, underscoring the potential of deep learning models—especially vision transformers—to enhance the efficacy and accuracy of fungal infection diagnosis. This study contributes to the growing body of research on artificial intelligence and medical mycology. The vision transformer's efficacy in classifying fungi confirms the transformative potential of deep learning in diagnostic methods.

Keywords Deep learning · Encoder · Flattened patches · Mycology · Neural networks · Transformer · Yeasts

S. Abhishek (✉)
Rolls-Royce Inc, Bangalore, India
e-mail: abhishekabi2002@gmail.com

T. Anjali
Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India
e-mail: anjalit@am.amrita.edu

R. B. Bentov
Department of Comparative Medicine, Stanford University School of Medicine, 300 Pastuer Drive, Edward Building, R320 Stanford, CA, USA
e-mail: rbarouch@stanford.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_55

1 Introduction

Fungal infections pose a serious risk to public health worldwide, impacting people from a variety of backgrounds and geographical areas. Surface-level or systemic infections are possible. The potential for serious illness and mortality from these conditions, particularly in those with weakened immune systems, highlights the need for early diagnosis and intervention. The knowledge of laboratory mycologists, who closely inspect and categorize fungus species, has historically served as the foundation for conventional mycological diagnostic techniques. This paradigm does have certain drawbacks, however, such as the labor-intensive procedures needed and the subjective nature of human assessment [1].

1.1 The Burden of Fungal Infections

Fungal infections cause a wide range of illnesses, from mild skin infections to serious and often deadly systemic mycoses, accounting for the majority of infectious disease cases worldwide. Effective diagnostic techniques are vitally needed, as shown by the rising number of immunocompromised persons, including organ transplant recipients, HIV/AIDS patients, and cancer patients, as well as the frequency of drug-resistant strains.

1.2 Traditional Mycological Diagnoses

The ability of mycology-focused lab specialists to carefully analyze fungal samples has long been essential to the diagnosis of fungal illnesses. Although this approach has shown to be successful, it is time-consuming and labor-intensive by nature, and it is only sometimes consistent among observers [1].

New technologies such as deep learning have the potential to improve patient outcomes, reduce the burden on healthcare systems, and speed up and improve diagnostic processes in the modern world [2]. The primary objective of this research is to improve the discipline of mycology by automating the categorization of fungal species using deep learning technology. The work intends to enhance patient outcomes, lessen the strain on healthcare systems, and develop diagnostic procedures related to fungus identification via the integration of artificial intelligence (AI) approaches.

To improve the objectivity, speed, and accuracy of fungal identification, we have set out on a novel path using state-of-the-art artificial intelligence (AI). Our method is unique in that it makes use of a cutting-edge deep learning model called a vision transformer. This divergence from conventional practices highlights the revolutionary potential of AI in diagnostic processes, especially in the field of mycology.

The results of our trial demonstrate how well the vision transformer can categorize fungus, underscoring its potential to reform mycological diagnostics and speed up processes that might change people's lives. These findings have far-reaching implications for the field of mycology as well as for the larger field of AI applications in healthcare.

Our results demonstrate the amazing power of artificial intelligence, with a 97.3% accuracy rate. This shows that AI has the potential to greatly improve the speed and precision of identifying fungal diseases in people. The foundation for a more sophisticated and effective way of diagnosing and treating fungal infections is laid by this study, which deftly incorporates deep learning techniques into the diagnostic framework. The use of vision transformers in particular demonstrates how deep learning has the ability to significantly improve diagnostic methods in a variety of disciplines.

2 Related Works

There are several obstacles in the way of the investigation into the automatic identification of fungi using deep learning in mycology. These difficulties include a dearth of real-world clinical data [1], issues with the models' interpretability [3], and a glaring lack of variety in the training datasets [4, 5]. But our novel strategy sticks out as a ground-breaking resolution to these common problems seen in earlier research on the identification and categorization of fungus.

For example, Essalat, M., Abolhosseini, M., Le, T.H. et al. [3] presents a unique architecture that aims to meet the important requirement for accurate and timely diagnosis of fungal and acanthamoeba keratitis via the use of this interpretable deep learning technique. Saliency maps highlight the portions of an image crucial for each diagnosis, providing light on the basic reasoning behind the model and fostering greater understanding and trust among medical professionals. Despite its apparent promise, this technique has shortcomings.

According to the study work [4], a deep learning method specifically, a Vision Transformer architecture can be used to develop an automated system for fungal detection. To increase the system's precision and generalizability further, the study proposes a crowdsourcing technique that leverages a smartphone app to collect additional tagged data from people discovering fungus in the wild. It's possible that the size of the dataset will limit the system's ability to identify rare or poorly reported fungal species.

Another research [5] proposes the automatic categorization of fungal genera from microscopic images using deep learning, which might transform the identification of fungal species in clinical settings. The procedure involves training Convolutional Neural Networks (CNNs) using a large number of fungal images. Although this method is fascinating, it has limitations, including hard data collection, unpredictable fungal morphology, and low computational and generalizability.

Convolutional Neural Networks (CNNs) are recommended for use in the microbiology field [6] for several purposes, including the automated identification of bacteria, viruses, and parasites and the measuring of target microbes. Despite the fact that the research is encouraging, the authors highlight a few drawbacks, including the underutilization of deep learning's potential in this area, challenges in handling large biological image datasets, and the need for additional work to address data scarcity, improve generalizability, and fully realize the potential of deep learning for the biological sciences.

The suggested approach [7] entails automatically identifying fungal species by using a deep learning model. The paper's authors mention a few systemic constraints. First, because of its interactive nature, new users must get training before submitting records of the highest caliber and actively participating in the validation process. Users may need to invest a lot of time and effort in this. Second, the data from Denmark utilized in the research could not apply to other areas where fungus species vary. Ultimately, different CNN and ViT designs could function better since the study only considers a small number of them.

Convolutional neural networks (CNNs) are used in the suggested approach [8] to categorize fungal genera from microscopic images. With an accuracy of 65.35% for the top 1 prediction and 75.19% for the top 3 predictions, the DenseNet model performed the best. Once unusual genera with low sample occurrence were eliminated, and data augmentation methods were used, the performance was further enhanced to exceed 80%. The authors discovered that their approach outperformed the model of Zielinski et al. (2020) on the same dataset. While additional training data is still needed for improvement, the proposed method performs well overall when categorizing fungal species from microscopic images.

Our approach addresses known problems with computing efficiency, erratic and fungal morphology. Our approach accounts for regional variations in fungal species and streamlines user training to overcome systemic constraints in automated fungal species identification. Our recommended method outperforms existing algorithms in the categorization of fungal taxa based on microscopic images.

3 Materials and Methods

3.1 Dataset

The collection includes more than five thousand high-quality images that were obtained by manual examination of microscopic fungi associated with fungal diseases of the skin, including yeasts, molds, and dermatophyte fungus [9]. Prominent experts in the area painstakingly categorized every picture, producing an accurate and comprehensive representation of a wide range of fungal diseases. This meticulous classification ensures that the dataset is accurate, making it suitable for academic research. The use of automated cropping algorithms has improved the dataset's effectiveness and produced a more standardized and well-organized set.

3.2 Data Pre-processing

One of the main goals in preparing the data for use in neural network models was to clean up and standardize the dataset. This included establishing consistent image dimensions, standardizing pixel values, and reducing the effects of lighting variances [10]. We used number of methods to remove irrelevant data, such as background and noise reduction, in order to increase the dataset's signal-to-noise ratio [11, 12]. The characteristics in the images were further improved by applying contrast and histogram equalization, which made the images ideal for accurate model training.

3.3 Data Augmentation

By using approaches for data augmentation, the dataset was improved, becoming more robust and diverse. Images were subjected to arbitrary flips, rotations, compressions, and displacements in order to replicate the real-world differences seen in fungal infections [12]. To the dataset were also included artificial lighting variations and simulations with varying degrees of magnification. These steps were intended to increase its variety and reduce the likelihood of overfitting.

3.4 Principal Component Analysis (PCA)

Using a linear technique, Principal Component Analysis (PCA) reduces the dimensionality of data in order to compress large datasets into a smaller format while maintaining maximum variability. To highlight the most notable variations in the data, the emphasis is on finding critical components and orthogonal directions within the original characteristics [12]. The scatter plot shown in Fig. 1 presents data in two dimensions, converting high-dimensional characteristics into a two-dimensional space by the use of PCA. Each point represents a data instance, with the x and y axes representing the first and second principal components. Class labels tint the points, highlighting correlations and trends. This succinct illustration facilitates comprehension of the dispersion and arrangement of data points in a space with fewer dimensions.

3.5 Model Evaluation

The effectiveness of machine learning models may be thoroughly evaluated using evaluation measures, such as confusion matrices, accuracy curves, loss curves, and the Matthews Correlation Coefficient (MCC) [13]. Confusion matrices provide comprehensive classification insights, confusion curves monitor training and validation

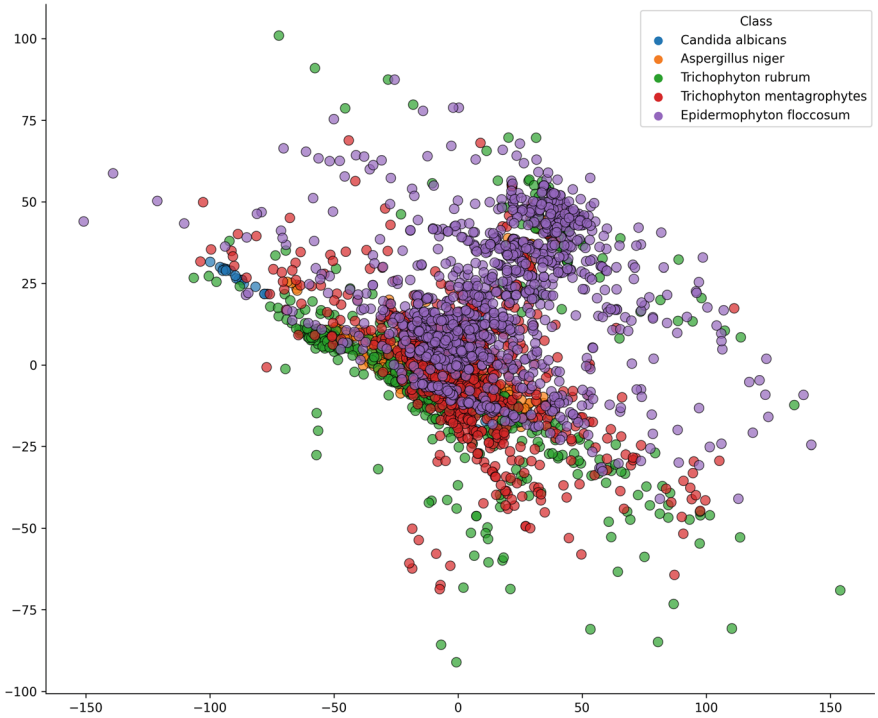


Fig. 1 Principal component analysis of the dataset

errors, and accuracy curves illustrate correctness trends. These factors are distilled into a single metric by MCC, which provides a concise assessment of overall classification efficacy and is particularly useful for unbalanced datasets.

4 Proposed Methodology

The objective of this study is to advance the area of dermatological image categorization by using the potential of Vision Transformers (ViTs). ViTs were originally designed for natural image processing, but they have shown outstanding performance in a wide range of computer vision applications.

4.1 Vision Transformer

The Vision Transformer (ViT) design offers a novel method of image categorization in dermatological image analysis for fungal diseases. By seeing pictures as a series of

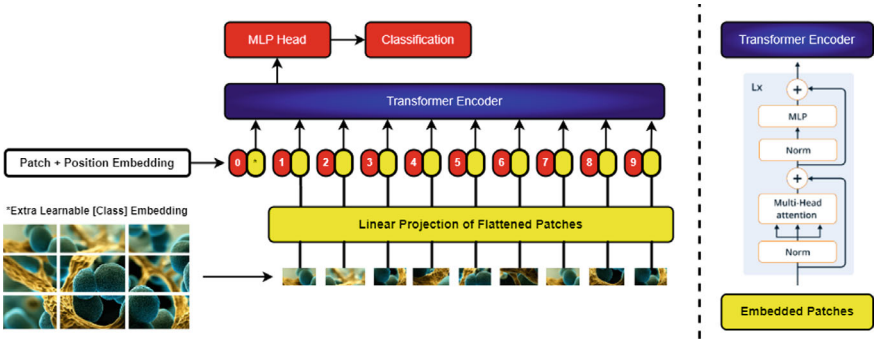


Fig. 2 Proposed system architecture of vision transformer

unique, non-overlapping patches, ViT transforms conventional methods [14]. This technique is quite good at capturing complex patterns and broad relationships in pictures. Several essential components make up the ViT architecture, as shown in Fig. 2.

Patch Embeddings The fixed-size patches of the picture are transformed into high-dimensional vectors by a linear embedding technique. This maintains important local data that is needed to diagnose disorders of the skin.

Positional Encoding Positional encodings are presented to provide spatial features to the patch embeddings. This stage is to make sure the model understands the spatial relationships between different patches.

Transformer Encoder Blocks ViT uses a set of transformer encoder blocks, each of which has a feedforward neural network and a multi-head self-attention mechanism. The model uses the self-attention mechanism to ascertain the importance of each patch, allowing it to incorporate contextual information from both local and global sources [15]. When the feedforward network processes the attended patches, it introduces nonlinearity into the model.

Layer Normalization and Residual Connections Layer normalization is applied to every sub-layer, and residual connections guarantee that data flows freely across the network.

Classification Head The last transformer block outputs a series of vectors denoting various patches. Once a single vector has been created using mean pooling, it is classified by passing it through a linear layer.

Output Layer The linear layer generates class probabilities, and the final output for the dermatological classification of images is obtained using a softmax activation function.

Pre-trained ViT models are refined on dermatological datasets using the suggested process, which also entails adding attention mechanisms to pick up on minor patterns and comparing the ViT's output to that of traditional convolutional neural networks

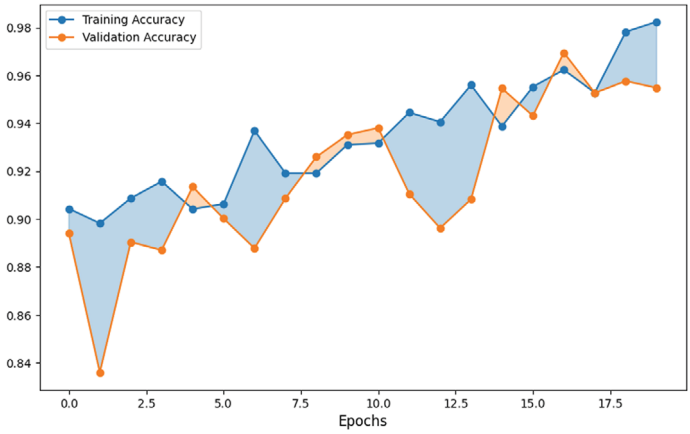


Fig. 3 Accuracy curves

(CNNs). ViT architecture's strength is its capacity to adjust to the particular features of dermatological images, which enables accurate and detailed diagnosis of fungal diseases. ViT is a potential method for improving computer-aided diagnosis in dermatology because of its attention processes and ability to collect local and global characteristics.

5 Experimental Results

The experimental findings show that the suggested vision transformer-based model performs very well in fungal infection diagnosis, with an astounding 97.3% accuracy. The assessment metrics—Matthews Correlation Coefficient (MCC), Balanced Accuracy, Loss, Precision, Recall, and F1 Score—offer a thorough grasp of the model's capabilities.

5.1 Accuracy Curve

The accuracy curve as shown in Fig. 3 depicts the range of thresholds across which the model can accurately categorize occurrences. The model exhibits a sharp increase in accuracy, culminating at a maximum of 97.3%, demonstrating its resilience in identifying genuine positive and negative situations.

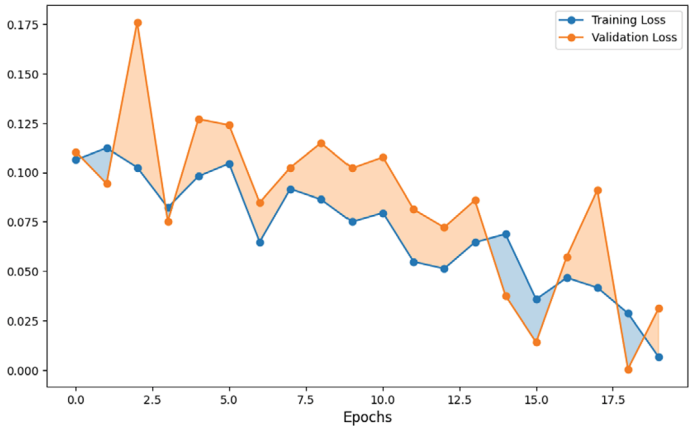


Fig. 4 Loss curves

5.2 Loss Curve

A steady decline can be seen in the loss curve as depicted in Fig. 4, which shows the model's convergence throughout training. The model's efficient learning process, which minimizes the difference between expected and actual results, is shown by the reported loss of 17%. This convergence is essential in order to ensure the model's stability and dependability in practical diagnostic applications.

5.3 Confusion Matrix

The model's classifications are broken down into detail in the confusion matrix, which also offers insights into the true positive, true negative, false positive, and false pessimistic predictions of the model which is shown in Fig. 5. A large number of true positives and true negatives are shown in the matrix, highlighting the model's capacity for precise diagnosis. With misdiagnoses having severe repercussions in real-world applications, a low percentage of false positives is shown by the accuracy value of 95%. The model's ability to reduce false negatives-a critical component in preventing the overlooking of real cases-is demonstrated by its recall value of 96%.

5.4 Precision, Recall, F1 Score, and MCC

The model's performance is detailed via the precision, recall, F1 score, and Matthews Correlation Coefficient (MCC) measurements. The recall value of 96% shows that

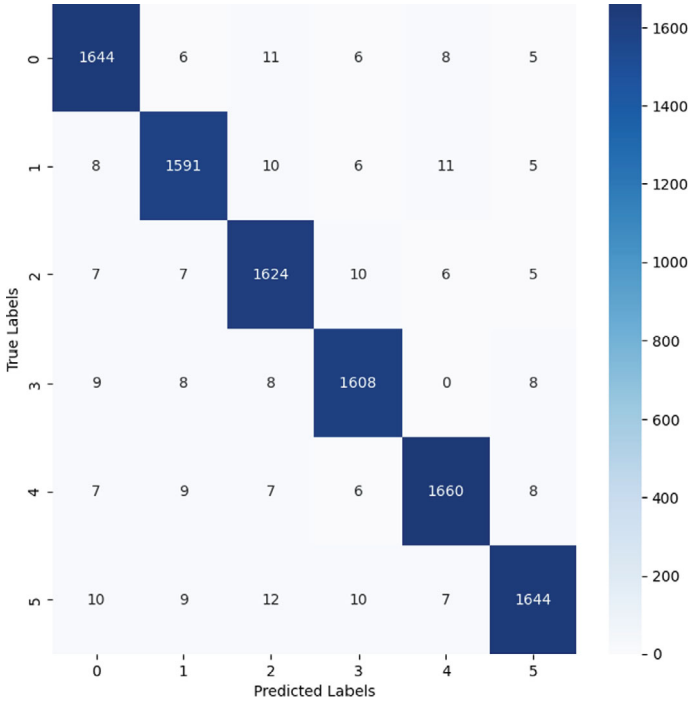


Fig. 5 Confusion matrix

the model was able to identify the majority of positive examples. In contrast, the accuracy value of 0.95 indicates high confidence in optimistic predictions. An even performance is shown by the F1 Score, which measures recall and accuracy and is 94%. Averaging 93%, the MCC further validates the model's efficacy in light of class distribution imbalances often seen in real-world datasets.

The accuracy, loss, and confusion curves may fully comprehend the model's behavior. Its capacity to handle the intricacies of dermatological image classification in the real world can be further understood by looking at precision, recall, F1 Score, and MCC [16]. The model's outstanding performance on all of these parameters as depicted in Fig. 6 indicates its suitability for real-world use in clinical settings, where precise and trustworthy diagnoses are critical. These comprehensive experimental findings support the vision transformer's promise as a reliable instrument for practical help in dermatological diagnosis.

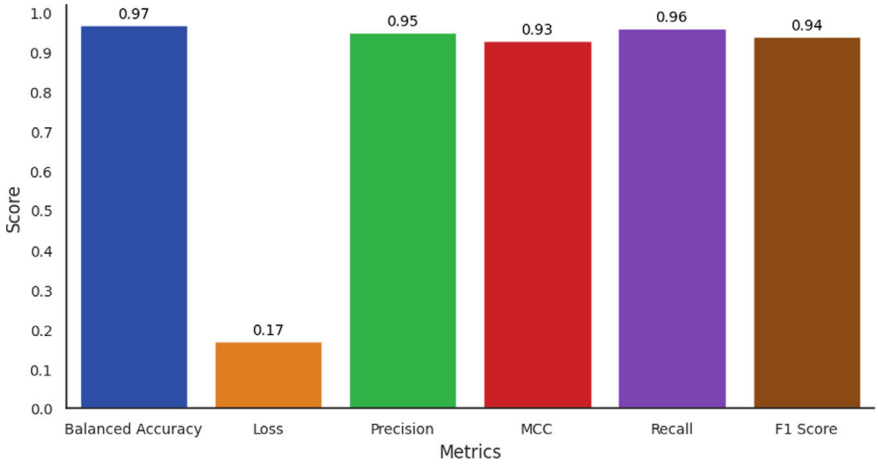


Fig. 6 Performance metrics

5.5 Early Detection of Fungal Infections

The vision transformer is an invaluable instrument for the early diagnosis of fungal diseases due to its remarkable accuracy and precision. Early detection and treatment of infections are made possible by the model's ability to recognize subtle patterns and minute features.

5.6 Clinical Diagnosis Enhancement

The accuracy of fungal infection diagnosis is improved in clinical settings by the use of vision transformers. Dermatologists and other medical practitioners may use the model's accurate and automated classification to make well-informed judgments quickly, resulting in more efficient treatment regimens.

5.7 Automation of Mycological Laboratories

Mycological labs may automate classifying fungal specimens by using vision transformers. Laboratory mycologists may concentrate on more intricate studies since this automation expedites diagnosis and ensures correct and consistent preliminary classifications.

6 Conclusion

In conclusion, our study demonstrates the revolutionary potential of deep learning-specifically, vision transformers in medical mycology. This work shows that artificial intelligence may significantly improve the accuracy and efficiency of detecting fungal infections in humans, as seen by its remarkable 97.3% accuracy rate. The proficient categorization of diverse fungal species shows deep learning models' versatility in handling the intricacies of mycological investigations. This study not only adds to the growing corpus of knowledge in medical mycology but also lays the groundwork for further developments in diagnostic techniques in future. Artificial intelligence's potential to transform diagnostic processes and enhance public health outcomes in mycology seems promising.

Future work should concentrate on providing an interface that is easy for physicians to use and evaluate real-time hospital apps as more practical. Mycologists, computer scientists, and other healthcare professionals must continue to collaborate to improve diagnosis accuracy and move artificial intelligence closer to being a standard part of medical procedures.

Acknowledgements We would like to acknowledge that the research presented in this paper does not involve or utilize any data or resources from Rolls-Royce India Private Limited, Bangalore, India. The work presented here is based solely on independent research and publicly available data sources.

References

1. Wickes BL, Romanelli AM (2020) Diagnostic mycology: xtreme challenges. J Clin Microbiol 58(4):10–1128. https://doi.org/10.1128/JCM.01083-20
2. Salil A, Joy N, Faizal B (2023) A prospective study comparing itraconazole alone versus systemic steroids alone as adjuncts to topical steroids in the post-operative management of allergic fungal rhinosinusitis. Clin Otolaryngol 48(2):356–362. https://doi.org/10.1111/coa.14014
3. Essalat M, Abolhosseini M, Le TH et al (2023) Interpretable deep learning for diagnosis of fungal and acanthamoeba keratitis using in vivo confocal microscopy images. Sci Rep 13:8953. https://doi.org/10.1038/s41598-023-35085-9
4. Picek L, Šulc M, Matas J, Heilmann-Clausen J, Jeppesen TS, Lind E (2022) Automatic fungi recognition: deep learning meets mycology. Sensors (Basel, Switzerland) 22(2):633. https://doi.org/10.3390/s22020633
5. Kumpatla G, Veresi H, P BS, Abhishek S, T A (2023) Revolutionizing brain tumour prediction: a pioneering GAN-based framework for synthetic data generation, 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, pp 548–553, https://doi.org/10.1109/I-SMAC58438.2023.10290627.
6. Zhang Y, Jiang H, Ye T, Juhas M (2021) Deep learning for imaging and detection of microorganisms. Trends Microbiol 29(7):569–572. https://doi.org/10.1016/j.tim.2021.01.006
7. Picek L, Šulc M, Matas J, Heilmann-Clausen J, Jeppesen TS, Lind E (2022) Automatic fungi recognition: deep learning meets mycology. Sensors 22(2):633. https://doi.org/10.3390/s22020633

8. Rahman MA, Clinch M, Reynolds J, Dangott B, Meza Villegas DM, Nassar A, Hata DJ, Akkus Z (2023) Classification of fungal genera from microscopic images using artificial intelligence. J Pathol Inf 14:100314. https://doi.org/10.1016/j.jpi.2023.100314
9. Sopo CJP, Hajati F, Gheisari S (2021) DeFungi: direct mycological examination of microscopic fungi images. arXiv preprint arXiv:2109.07322
10. Panichikkal J, Jose A, Sreekumaran S, Ashokan AK, Baby CS, Krishnankutty RE (2021) Biofilm and biocontrol modulation of Paenibacillus sp. CCB36 by supplementation with zinc oxide nanoparticles and chitosan nanoparticles. Applied Biochemistry and Biotechnology, 1–15
11. Nair SV, Baranwal G, Chatterjee M, Sachu A, Vasudevan AK, Bose C, Biswas R (2016) Antimicrobial activity of plumbagin, a naturally occurring naphthoquinone from Plumbago rosea, against Staphylococcus aureus and Candida albicans. Int J Med Microbiol 306(4):237–248
12. Abhishek S, Prakash P, Anjali T, Vijayaraghavan S (2023) Craniofacial and cervical aesthetic surgical outcome analysis using visual similarity metrics. In: Shakya, S., Tavares, J.M.R.S., Fernández-Caballero, A., Papakostas, G. (eds) Fourth International Conference on Image Processing and Capsule Networks. ICIPCN 2023. Lecture Notes in Networks and Systems, vol 798. Springer, Singapore. https://doi.org/10.1007/978-981-99-7093-3_34
13. Remya S, Anjali T (2023) An intelligent and optimal deep learning approach in sensor based networks for detecting microbes. IEEE Sensors Journal
14. Raj S, Vinod V, Jayakumar J, Suresh P, Kumar A, Biswas R (2021) Antifungal activity of Syzygium samarangense leaf extracts against Candida. Lett Appl Microbiol 73(1):31–38
15. T A, Abhishek S, S R (2023) NeuroDerm: pioneering dermatological diagnostics with advanced neural networks, 2023 7th international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 2023, pp. 1161-1168, https://doi.org/10.1109/I-SMAC58438.2023.10290631.
16. Mohammed N, Rejinold NS, Mangalathillam S, Biswas R, Nair SV, Jayakumar R (2013) Fluconazole loaded chitin nanogels as a topical ocular drug delivery agent for corneal fungal infections. J Biomed Nanotechnol 9(9):1521–1531

Deep Learning Models for Enhanced RUL Prediction in Turbofan Jet Engines

J. Judeson Antony Kovilpillai, Sulaiman Syed Mohamed, Pragya, Mahmood Hussain Mir, Tinka Singh, and Uday Kumar Singh

Abstract This paper presents an analysis of advanced deep learning techniques for determining the remaining useful life (RUL) of aircraft engine components, with a focus on turbofans and high-pressure compressors. Through this analysis the importance of RUL forecasting to ensure flight safety and efficiency has been emphasized. The study includes papers that use the C-MAPSS Aircraft Engine Simulator Data provided by NASA, which allows realistic simulation of engine damage over different periods of time. Four different datasets were simulated under different operating environments and fault mechanisms, related to engine health with multiple sensors to effectively predict the RUL of turbofan jet engines. Different learning techniques, including convolutional neural networks (CNN), long short-term memory (LSTM) networks, autoencoders, recurrent neural networks (RNN), gated recurrent units (GRU), together with sparse autoencoders were investigated and their applications in RUL prediction with respect to capabilities and limitations were examined. Furthermore, the paper provides an extensive analysis of these salient deep learning models, using the root mean squared error (RMSE) as an evaluation performance metric. This study provides extensive insights into the effectiveness of these methods in RUL prediction, and facilitates informed decisions for practical applications.

J. Judeson Antony Kovilpillai (✉) · S. S. Mohamed · Pragya · M. H. Mir · T. Singh · U. K. Singh
Alliance University, Bangalore, India
e-mail: judeson.kovilpillai@alliance.edu.in

S. S. Mohamed
e-mail: sulaiman.mohamed@alliance.edu.in

Pragya
e-mail: pragya@alliance.edu.in

M. H. Mir
e-mail: mahmood.mir@alliance.edu.in

T. Singh
e-mail: tinka.singh@alliance.edu.in

U. K. Singh
e-mail: udaykumar.singh@alliance.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_56

Keywords RUL prediction · NASA C-MAPSS Dataset · Deep learning · LSTM · Autoencoders · RMSE

1 Introduction

The turbofan engine (Fig. 1) represents the standard propulsion system used in commercial and military aircraft developed by NASA. Their operational principle involves the ingestion of ambient air at the engine's core by using a high-capacity fan, absorbing large amounts of air in the turbofan jet engine [1]. Predictive maintenance and condition monitoring of turbofan engines and high-pressure compressors in jet engines are very essential for maintaining safety, performance, comfort, and operating efficiency. These key components play a vital role in the generation of thrust and the overall performance of the engine. The progressive deterioration of turbofan components within jet engines necessitates early and reliable assessment of their operational status. Failure to timely detect incipient anomalies in turbofans and high-pressure injection systems can lead to significant economic consequences through escalating repair costs, compromised operational safety, and potentially catastrophic incidents [2]. Consequently, continuous and accurate estimation of the remaining usable life (RUL) of these critical systems emerges as a paramount concern. By adopting a proactive approach to RUL prognostication, comprehensive and timely maintenance regimens can be implemented, guaranteeing optimal engine dependabil-

Fig. 1 Turbofan of NASA jet engine

ity, efficiency, and safety throughout their projected lifespans. The field of prognostics is increasingly turning to deep learning and machine learning to predict when crucial aircraft engine components, like turbofans and high-pressure compressors, are nearing failure. This research dives deep into the potential of these tools to optimize engine lifespan and aircraft safety [3]. One key focus is remaining useful life (RUL) prediction, where sophisticated algorithms analyze sensor data to forecast impending component failure [4]. Convolutional neural networks (CNNs) extract valuable features from images while long short-term memory (LSTM) networks capture the complex temporal dynamics within sensor data, both contributing to accurate RUL estimates. To handle the vast amount of data, autoencoders efficiently compress information while preserving key features. Recurrent neural networks (RNNs) then model the inherent temporal dependencies, while the effectiveness of gated recurrent units (GRUs) and Sparse Autoencoders in reducing computational burden and enabling sparsity-based learning is also explored.

Simulation techniques come into play, emulating turbofan and compressor performance under various conditions, further refining RUL forecasts. These tailored models, specific to each component's characteristics, ultimately enhance aircraft safety. Beyond deep learning, the research delves into essential feature extraction and data manipulation techniques. The Kalman filter improves data quality by analyzing state statistics and suppressing noise, while Principal Component Analysis (PCA) reduces dimensionality for improved efficiency. Uncertainty is addressed through error prediction algorithms, and dependency analysis uncovers relationships between variables, informing feature selection and model development. Extended/residual convolutions extract more discriminative features from images, and Single Exponent Smoothing (SES) provides a resource-efficient method for time series prediction. Finally, Random Forest Feature Selection identifies the most impactful variables, optimizing feature selection strategies. All the mentioned deep learning techniques contribute to the RUL prediction process tools to evaluate, forecast, and enhance the performance of aircraft engine components.

2 Search Methodology Dataset Description

This paper used an advanced search strategy to identify relevant scholarly articles and documents in Scopus and SCI from popular databases such as IEEE Xplore, Springer, MDPI, and ScienceDirect and also consumed Google Scholar used to ensure proper analysis. The search was conducted focusing on keywords related to "RUL Prediction", "Deep Learning", "LSTM", "Autoencoders", "C-MAPSS NASA dataset", "Engine degradation simulation", etc. except information and documentation Referring to the integration of deep learning techniques to predict remaining useful life (RUL), especially for aircraft engine components, this design approach ensured a variety of materials and methods effective research methods were included.

All of the studies analyzed in this paper use an open NASA dataset [5] specifically designed for asset degradation modeling, specifically targeting turbofan jet engines

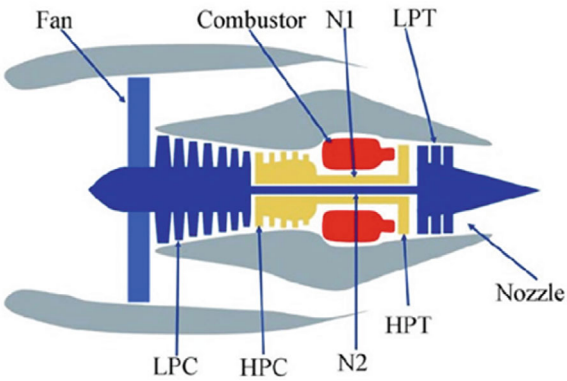


Fig. 2 Structure of simulated C-MAPSS engine

(Fig. 2.) Calling this dataset the C-MAPSS contains simulated Run-to-Failure statistics for detailed analysis. Engine degradation was simulated accurately, resulting in four distinct stages, each with different operating conditions and fault modes multiple sensor channels were recorded to capture the progression of faults within the engines. This dataset was provided by the Prognostics Center of Excellence (CoE) at NASA Ames Research Center. It consists of multiple multivariate time series, with each dataset further subdivided into training and test subsets. Notably, each time series corresponds to a different engine, collectively representing a fleet of engines of the same type. The dataset constitutes information regarding the engines' initial wear and manufacturing variation, as these factors are considered part of normal operating conditions. Additionally, the dataset encompasses three critical operational settings, each significantly affecting engine performance, and these settings are included in the data. It's important to mention that the data is subject to sensor noise.The dataset initially reflects engines operating under normal conditions at the beginning of each time series. However, faults develop at some point during the series. In the training set, these faults intensify until they lead to system failure, while in the test set, the time series concludes before system failure. True remaining useful life (RUL) values for the test data are also provided for reference.

The dataset is presented in a compressed text file format, comprising 26 columns of numerical data. Each row represents the simulated data taken during a single operational cycle. The dataset includes different variables like unit number, cycle time, operational settings, and sensor measurements. It is divided into four distinct subsets: FD001, FD002, FD003, and FD004. Each dataset consists different numbers of training and test trajectories, conditions, and fault modes as outlined in Table 1. This variational simulation provides different types of scenarios to analyze machine degradation and predict Remaining Useful Life of the same. Notably, Dataset FD001 is simulated in a conventional experimental setup, consisting of 100 training trajectories and 100 test trajectories. It simulates a single operational condition (Sea Level) and evaluates the degradation of the High-Pressure Compressor (HPC). The complex scenario in which the FD002 dataset is simulated is more intense than that of FD001

Table 1 Dataset details

Dataset	Train trajectories	Test trajectories	Conditions	Fault modes
FD001	100	100	ONE (Sea Level)	ONE (HPC Degradation)
FD002	260	259	SIX	ONE (HPC Degradation)
FD003	100	100	ONE (Sea Level)	TWO (HPC Degradation, Fan Degradation)
FD004	248	249	SIX	TWO (HPC Degradation, Fan Degradation)

as it comprises 260 training trajectories and 259 test trajectories. It has six different operational conditions but at different levels. As in FD001, the primary fault mode in FD002 is HPC degradation. This dataset, like FD003, has 100 Sea Level training and testing instances conducted using a standard experimental setup. This data collection includes two fault scenarios for HPC and Fan deterioration. FD004, unlike the other datasets, presents a different scenario and includes a comprehensive data set of 248 training and 249 test trajectories. It has six different operational conditions, comparable to FD003, with two types of fault modes connected with HPC deterioration and fan degradation included in the dataset.

The experimental data analysis (EDA) is utilized to evaluate sensor variation in turbofan jet engines, which is crucial for developing dependable deep learning models for effective remaining usable life (RUL) prediction. A systematic technique was used to determine the standard deviation of the sensors on a per-engine basis utilizing the unit number (engine identifier), as well as to calculate the standard deviation of each sensor and the average as it rose with all other sensors. The acquired numbers are then shown on a bar chart, which provides insight into how sensor aberrations distribute themselves in turbofan jet engines. Figure 3 shows a visual representation of the average number of sensors in an aircraft, as well as how many sensors are accessible per-engine on average. Every strip in Fig. 3 refers to one engine only, therefore making it feasible to investigate typical patterns or variations in sensor activity. These are essential considerations for training and optimizing deep learning models to improve RUL prediction accuracy for turbofanjet engines. The graph displays critical data on the average standard deviation and indicates any patterns or exceptions that could affect RUL forecasting ability. Each line in the graph depicts a specific engine; RUL is represented along the X-axis, and signal intensity of the corresponding sensor is shown along the Y-axis. For example, Figs. 4 and 5 displays temporal evolution of the sensor signals as fan speed and outlet temperature for every 10th engine to have a representative sample. Figure 6 indicates that the engine can achieve up to 210 cycles before failure.

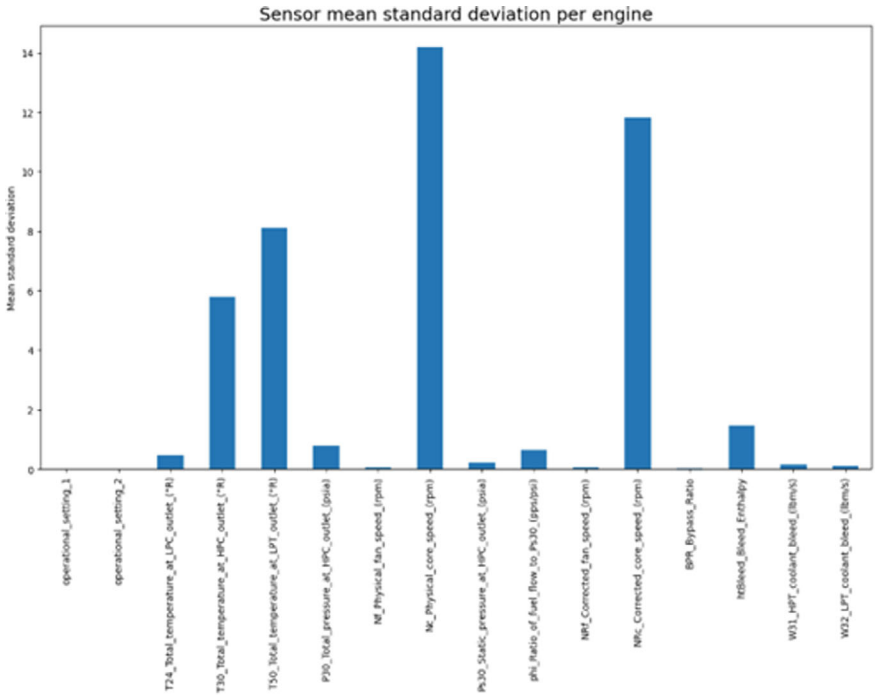


Fig. 3 Sensor mean distribution

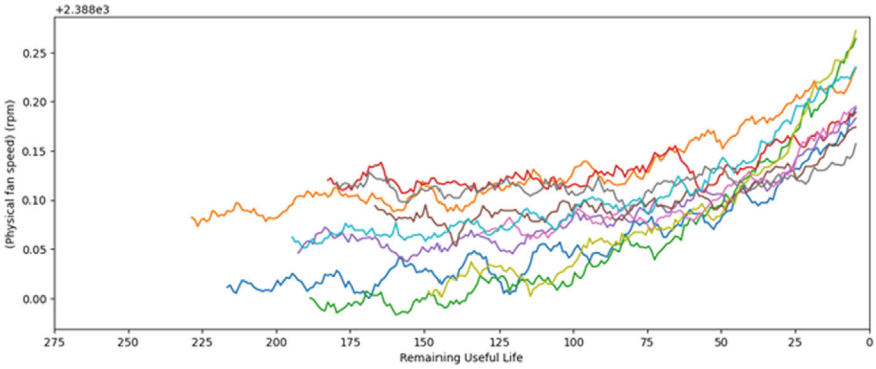


Fig. 4 EDA of fan speed in C-mapss dataset

3 Methodology

Remaining useful life (RUL) prediction of turbofan jet engines involves a methodical series of steps. It commences with data collection and preprocessing, where sensor data and operational information are gathered and pre-processed [6, 7]. Design of

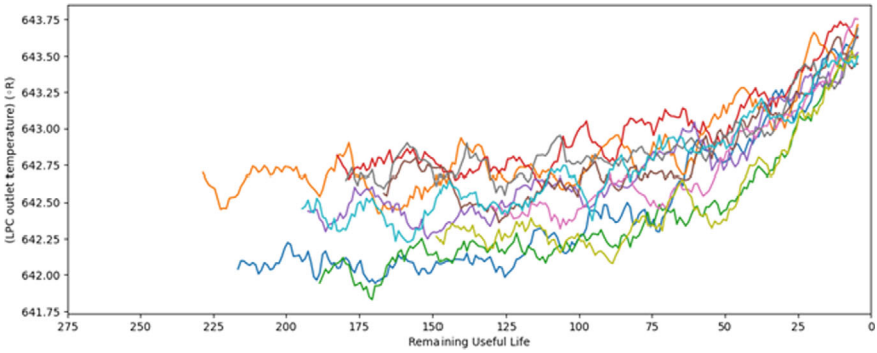


Fig. 5 EDA of outlet temperature

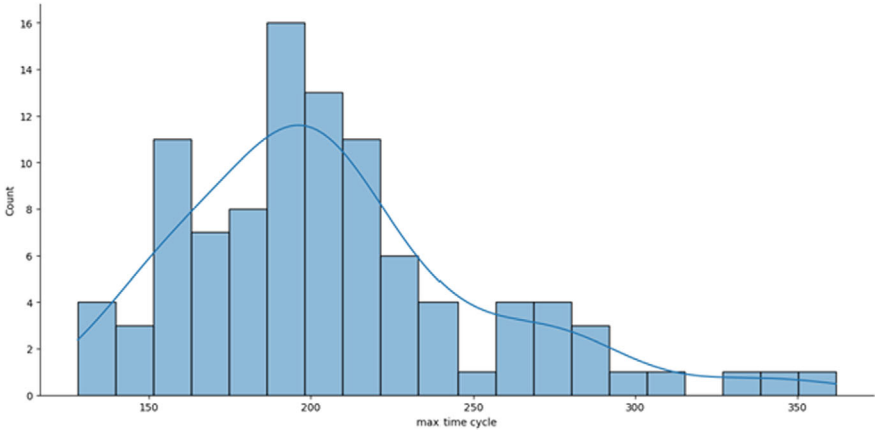


Fig. 6 EDA of max time cycle

architectures like CNN, SAE, and LSTM is done after preprocessing, encompassing the definition of layers, neurons, and activation functions. The layers of the model and deep learning concepts like Random Forest Feature Selection and Principal Component Analysis (PCA) are then utilized to automatically extract pertinent features. The model is trained on the training dataset, with hyperparameter adjustments to optimize its performance, and a validation set is used to fine-tune the model. Subsequently, the model is applied to the testing dataset for RUL prediction, where model accuracy is evaluated using metrics like Root Mean Square Error (RMSE) by comparing predicted RUL to actual values [8, 9]. Predictions are visualized and influential sensors and features are analyzed. For deployment, the model is used in real-time or monitoring systems, with periodic learning to ensure accuracy. The deep learning workflow for RUL estimation is depicted in Fig. 7.

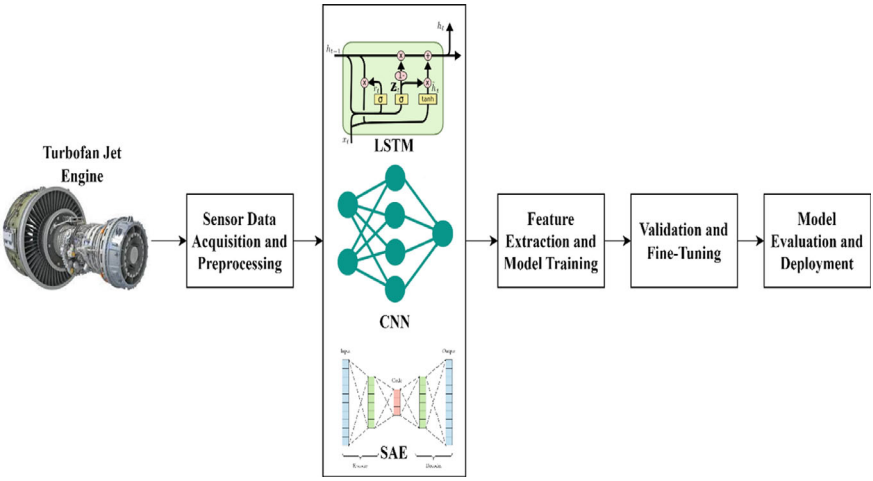


Fig. 7 RUL estimation using LSTM/CNN/SAE

Different deep learning architectures evaluated in this research investigation are as follows:

CNN + Bidirectional LSTM Architecture: Combining convolutional neural network (CNN) with bidirectional long short-term memory (LSTM) layers enhances the model's ability in forecasting Remaining Useful Life (RUL). CNN layers are used for retrieving features, making them suitable for discovering structural patterns in data. Similarly, bidirectional LSTMs deal with temporal relationships by interpreting patterns in both forward and backward directions. By combining these two layers, this method enables a thorough analysis of sequential and structural features of the data. This integrated deep learning model model is very effective at capturing complicated patterns, making it suitable for estimating RUL.

Autoencoder + LSTM Architecture without Elastic Weight Consolidation: The combination of Autoencoder and LSTM is also an efficient technique for forecasting RUL of the turbofan jet engines. It begins with an Autoencoder, which decreases data size and removes noise. This technique does not use Elastic Weight Consolidation and possesses a much simpler architecture than other deep learning models. Despite being less complex, it offers an effective balance between data compression and sequence modeling, resulting in an efficient RUL prediction technique.

CNN-RNN Architecture: The CNN-RNN technique focuses on using Convolutional Layers to predict RUL of the turbofan jet engines. Convolutional neural networks are efficient extracting features, making them suitable for structured data. This model simplifies the architecture by removing the iterative layer and provides a flexible mechanism for RUL prediction. The absence of repetitive deep learning layers emphasizes the performance of the model in effective RUL prediction.

Attention-based DCNN model: The attention-based DCNN model is an effective framework for predicting remaining useful life (RUL). This model efficiently recognizes fault conditions and analyzes RUL by combining convolutional layers, which is optimal for extracting critical features, with an attention mechanism that preferentially concentrates on critical data points. It provides crucial information about the performance of the turbofan jet engines under examination. Furthermore, the addition of the Prognosability Algorithm improves the model's capabilities by assisting in the detection of anomalies while offering prognostic insights. This deep learning strategy has shown to be quite efficient in dealing with the various data relationships that exist in complex systems.

LSTM Architecture with Data Fusion: To forecast RUL, the data fusion approach based on LSTM includes both long-term and short-term memory layers, as well as principal component analysis (PCA) is very efficient. LSTM layers are used in processing ordinal data, whereas PCA is used to reduce data dimensionality and improve feature extraction. By combining the benefits of these two methodologies, this methodology helps in capturing critical information while improving data representation. In multidimensional datasets, this deep learning architecture is an effective method for overcoming common prediction modeling pitfalls.

LSTM Architecture with Moving Median Filter: A hybrid technique is proposed for predicting remaining useful life (RUL), combining long short-term memory (LSTM) layers and a Moving Median Filter. By adding LSTM layers, the model optimally captures temporal correlations in the data, whereas the Moving Median Filter is used to reduce noise and improve data quality. Furthermore, the use of correlation analysis enhances the model's capacity to find meaningful patterns. This technique strikes an effective balance between LSTM modeling and data preprocessing, yielding more reliable RUL estimate models.

CNN-LSTM Architecture with 3 dense and 2 dropout layers with Genetic Optimization: The CNN-LSTM model uses convolutional neural network (CNN) and long short-term memory (LSTM) layers, as well as dense and dropout layers, to develop efficient deep learning model for predicting remaining useful life (RUL). This fusion includes not only the structural and sequential features, but also the detailed feature processing achieved through the integration of CNN and LSTM layers. The integration of these deep layers increases the model's complexity and depth, allowing for a deeper evaluation of input features. Dropouts are used to reduce overfitting and the use of Genetic Optimization for tweaking the model's hyperparameters improves the overall performance. This technique performs well with complicated datasets as optimal RUL forecasting requires robust feature extraction and hyperparameter optimization.

CNN-LSTM Architecture: The use of different activation functions, such as sigmoid, tanh, and relu, in the convolutional and long short-term memory (LSTM) layers improves the flexibility of this CNN-LSTM methodology. These activation functions are critical for the model to effectively assess and handle data. This approach has

the potential to increase the performance metrics of remaining useful life (RUL) forecasting by tolerating differing data features in diverse scenarios.

Sparse Autoencoders (SAE) + Temporal Convolutional networks (TCN) Architecture: This technique uses stacked sparse autoencoders (SAE) and temporal convolutional networks (TCN) to predict remaining usable life (RUL). SAEs improve the method's performance by reducing data dimensionality and eliminating noise, whereas TCN extracts relevant features using extended and residual convolutions. The layered nature of SAEs considerably improves data representation, and allows the model to effectively retrieve information at several scales. This deep learning approach is specifically developed to increase feature extraction, making it optimal for complex data sets like the NASA C-mapps dataset.

Hyper Tuned LSTM model: The improved hyper-tuned LSTM model predicts RUL using integration of long-term and short-term memory (LSTM) layers. This approach uses exponential smoothing to effectively decrease data noise. Furthermore, hyperparameter tuning is done to further increase the effectiveness of the model and adapt to different input kinds, resulting in higher accuracy.

Bayes-optimized Multi-Layer Perceptron model: The Bayes-optimized Multi-Layer Perceptron model uses Single Exponent Smoothing (SES) for data filtering and Random Forest Feature Selection. Data noise is significantly decreased by integrating SES, and relevant variables are identified and prioritized using random forest features. The optimization procedure utilizes a Bayesian approach, which improves efficiency and interpretability. This method is particularly ideal for data sets with varied signal-to-noise ratios, giving it a reliable and adaptable method for predicting RUL.

4 Discussion

Apostolopoulos & Tzani adopted a Transfer Learning approach, utilizing a custom VGG-19 model (MVGG-19). VGG-19 is renowned for its deep architecture, consisting of 19 layers, primarily composed of 3×3 convolutional layers. To enhance the model's robustness, the methodology integrated various data augmentation techniques such as random Gaussian noise, rotations, flips, and shifts in height and width. These augmentations were coupled with 2D convolutions, leading to an accuracy of 87.39%, a precision of 78.01%, a recall of 97.72%, an F-1 score of 87.37%, and an impressive AUC of 97.88%. Oh et al. ventured into Deep Learning by employing a Convolutional Autoencoder (CAE). A CAE comprises both encoder and decoder networks, emphasizing the extraction of essential features. The methodology was further enriched with data augmentation and convolutional operations. This resulted in a substantial accuracy of 98.87%, and an F-1 score of 98.58%. Habibpour et al. took a distinctive approach, amalgamating Transfer Learning with Feature Extraction using well-established architectures including VGG16, ResNet50, DenseNet121,

and InceptionResNetV2. These architectures are known for their depth and intricate design, allowing them to capture complex features. This ensemble methodology led to remarkable performance metrics; an accuracy of 99.6%, sensitivity of 99.4%, specificity of 99.9%, an AUC of 99.9%, and an uncertainty accuracy of 99%.

In their investigation, Remadna et al. [10] used a convolutional neural network (CNN) and a bidirectional long short-term memory (LSTM) model. The CNN layers extracted structural features, whereas the Bidirectional LSTM layers caught temporal dependencies. The model performed well, with comparatively small Root Mean Square Error (RMSE) values, especially in the FD001 and FD003 datasets. Maschler et al. [11] used an Autoencoder with LSTM to forecast remaining useful life (RUL). The Autoencoder's principal function was to minimize the dimensionality of the data. Their model produced competitive RMSE findings in the FD001 and FD003 datasets, but showed greater values in the FD002 and FD004 datasets. Amin and Kumar used a CNN-RNN technique for RUL prediction, with Convolutional Layers for the feature extraction. This technique performed rather well in the FD001 and FD003 datasets, but not equally well in the FD002 and FD004 datasets.

Muneer et al. [13] used an Attention-based DCNN model in conjunction with a Prognosability Algorithm. The results demonstrated that the model had superior RMSE values in FD001 and FD003, suggesting its ability to reliably capture fault patterns. Similarly, Soualhi et al. [14] used LSTM layers and Principal Component Analysis (PCA) for data fusion, yielding moderate RMSE values that were equivalent to other models. Asif et al. [15] used LSTM layers coupled with a Moving Median Filter to enhance data quality, resulting in impressive predictive accuracy, particularly in FD001.

Sabry and El-Latif [16] proposed a CNN-LSTM model with deeper dense and dropout layers, which was optimized with Genetic Optimization. The model had superior performance metrics, with continually low RMSE values throughout all datasets, demonstrating the model's durability. Karnik et al. [17] investigated a CNN-LSTM model using several activation functions. While their model performed quite well in FD001 and FD003, it had increased RMSE values in FD002 and FD004, indicating that its performance may differ based on the dataset's properties.

Liu et al. [18] used a model that integrated stacked sparse autoencoders (SAE) with temporal convolutional networks (TCN). The deep learning model produced considerably higher RMSE values, possibly due to the model's emphasis on complicated data representation. Kadre and Paspula [19] used a hyper-tuned LSTM model with exponential smoothing, yielding significant RMSE values, especially for FD002. Wang et al. [20] used a Bayesian-optimized Multi-Layer Perceptron with Single Exponent Smoothing (SES) and Random Forest Feature Selection.

In summary, the CNN-LSTM model refined with Genetic Optimization in 'Sabry and El-Latif [16]' consistently produced the lowest RMSE in FD002 and FD004, but 'Asif et al. [15]' with the LSTM + Moving Median Filter performed exceptionally well in FD001 and FD003. These models exhibit their adaptability in dealing with various dataset characteristics, as well as their ability to forecast RUL accurately in a variety of operational circumstances. Among the models tested, Asif et al.'s model performed best in capturing the complicated RUL patterns in FD001, with a score

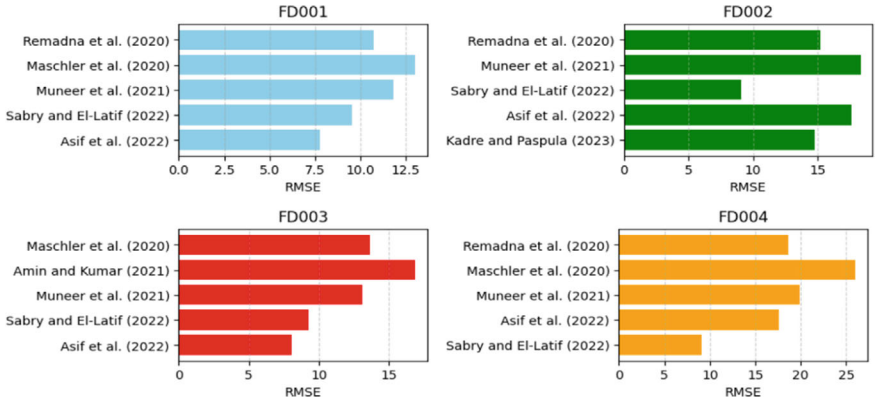


Fig. 8 Best performing models with respect to different C-MAPSS datasets

of 7.78. It also performed well in FD003, attaining an RMSE of 8.03, demonstrating that it can adapt to a wide range of data properties. However, in FD002 and FD004, the model had significantly greater RMSE values of 17.64 and 17.63, accordingly. This shows that the model may have had difficulty reliably forecasting RUL for these particular datasets, maybe due to differences in fault situations or operational settings. Sabry and El-Latif's CNN-LSTM model, which had three dense and two dropout layers and was fine-tuned via Genetic Optimization, produced competitive RMSE values throughout all four datasets. It achieved RMSE values of 9.54, 9.10, 9.27, and 9.07 in FD001, FD002, FD003, and FD004, demonstrating its robustness and adaptability across a wide range of datasets. The integration with further dense and dropout layers, as well as fine-tuning via Genetic Optimization, proved critical in increasing predicted accuracy. Overall, this model is comparatively more suited for RUL prediction, performing well under different operational scenarios.

The comparative analysis of different architectures for RUL prediction is tabulated in Table 2 and the best performing models are visualized in Fig. 8.

Notably, Remadna et al. [10] CNN+ Bidirectional LSTM model outperforms convolutional neural networks in handling sequential dependence. Sabri and El-Latif [16] CNN-LSTM model optimized by genetic algorithms benefits from synergistic combination of convolutional and LSTM layers, dropout methods, and genetic optimization. Muneer et al.'s [13] Attention-based DCNN model, with Convolutional layers and Prognosability Algorithm, exhibits skill in capturing important features by focusing on relevant features of input sequence the so the so. Maschler et al.'s [11] Autoencoder + LSTM model exploits autoencoders' constrained representation learning capabilities, followed by LSTM, without elastic weight integration, emphasizing the preservation of known positions over time. Together, these examples demonstrate the importance of considering architectural choices, preprocessing techniques, and optimization strategies to achieve better performance under the specific conditions of turbofan jet engine RUL prediction.

Table 2 Summary of methods and RMSE values

Reference	Methodology	Feature extraction	RMSE
Remadna et al. [10]	CNN + Bidirectional LSTM	Convolutional layers + Concatenated LSTM layers	FD001: 10.74 FD002: 15.20 FD003: 13.85 FD004: 18.60
Maschler et al. [11]	Autoencoder + LSTM without Elastic Weight Consolidation	Autoencoder	FD001: 13.03 FD002: 24.28 FD003: 13.60 FD004: 26.04
Amin and Kumar [12]	CNN- RNN	Convolutional Layers	FD001: 15.68 FD002: 22.26 FD003: 16.89 FD004: 26.32
Muneer et al. [13]	Attention-based DCNN model	Convolutional layers + Prognosability Algorithm	FD001: 11.81 FD002: 18.34 FD003: 13.08 FD004: 19.88
Soualhi et al. [14]	LSTM + Data Fusion	LSTM Layers + PCA	FD001: 15.32
Asif et al. [15]	LSTM + Moving Median Filter	LSTM Layers + Correlation	FD001: 07.78 FD002: 17.64 FD003: 08.03 FD004: 17.63
Sabry and El-Latif [16]	CNN-LSTM with 3 dense and 2 dropout layers + Genetic Optimization	Convolutional + LSTM Layers	FD001: 09.54 FD002: 09.10 FD003: 09.27 FD004: 09.07
Karnik et al. [17]	CNN-LSTM (sigmoid/tanh/relu)	Convolutional + LSTM Layers	FD001: 14.28 FD003: 16.07
Liu et al. [18]	Sparse Autoencoders (SAE) + Temporal Convolutional networks (TCN)	Stacked Sparse Autoencoder + Dilated/Residual convolutions	FD001: 18.01
Kadre and Paspula [19]	Hyper Tuned LSTM model	LSTM Layers with exponential smoothing	FD002: 14.76
Wang et al. [20]	Bayes-optimized Multi-Layer Perceptron	Single exponent smoothing (SES) algorithm + Random Forest Feature Selection	FD001: 17.14 FD002: 37.38 FD003: 18.08 FD004: 41.09

5 Conclusion

In conclusion, this review emphasizes the need of Remaining Useful Life (RUL) prediction and utilized the C-MAPSS NASA dataset for analyzing different deep learning models. The paper explored various architectural configurations, ranging from models with convolutional layers and concatenated LSTM layers to autoencoder models without Elastic Weight Consolidation. Further investigations encompassed models that integrated LSTM layers with PCA, the Moving Median Filter, and fine-tuned LSTM models with dense layers and dropout, optimized through Genetic Optimization.

Different activation functions, such as sigmoid, tanh, and relu, were also examined. Other investigations applied models that featured Sparse Autoencoders (SAE) combined with Temporal Convolutional Networks (TCN), Hyper Tuned LSTM models with exponential smoothing, and Bayes-optimized Multi-Layer Perceptrons with Single Exponent Smoothing (SES) and Random Forest Feature Selection as parts of the wider study. The model accuracy is assessed by means of Root Mean Square Error (RMSE), which are lower when the model has a better prediction. Among them, Sabry and El-Latif's [16] CNN-LSTM model with Genetic Optimization consistently achieved the lowest RMSE values in diverse datasets, proving its reliability across operational scenarios. Additionally, lightweight edge computing solutions could be explored. Furthermore, ensemble methods can help make interpretable models. Also for instance one can do an analysis on explainability and the augmentation of datasets to enhance model robustness.

References

1. Maulana F, Starr A, Ompusunggu AP (2023) Explainable data-driven method combined with bayesian filtering for remaining useful lifetime prediction of aircraft engines using NASA CMAPSS datasets. Machines 11(2):163
2. Chatterjee A, Kar I, Das R, Mukhopadhyay S, Chahar J, Kulbhushan P (2023) Comparative study of generative adversarial networks for sensor data generation-based remaining useful life classification. In: Hybrid computational intelligent systems (pp. 329-343). CRC Press
3. Ruan D, Wu Y, Yan J, Gühmann C (2022) Fuzzy-membership-based framework for task transfer learning between fault diagnosis and RUL prediction. IEEE Trans. Reliab
4. de Pater I, Mitici M (2023) Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder. Eng Appl Artif Intell 117:105582
5. https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xaut-bemq
6. Srinivasan A, Andresen JC, Holst A (2023) Ensemble neural networks for remaining useful life (RUL) prediction. arXiv preprint arXiv:2309.12445
7. Li Y, Chen Y, Hu Z, Zhang H (2023) Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. Reliab Eng Syst Saf 229:108869
8. Leal AS, Berton L, de Castro Santos LC (2022) Learning methods for remaining useful life prediction in a turbofan engine. In: Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional (pp. 556-566). SBC

9. Agrawal S, Sarkar S, Srivastava G, Maddikunta PKR, Gadekallu TR (2021) Genetically optimized prediction of remaining useful life. Sustain Comput Inform Syst 31:100565
10. Remadna I, Terrissa SL, Zemouri R, Ayad S, Zerhouni N (2020) Leveraging the power of the combination of CNN and bi-directional LSTM networks for aircraft engine RUL estimation. In: 2020 prognostics and health management conference (PHM-Besançon) (pp. 116-121). IEEE
11. Maschler B, Vietz H, Jazdi N, Weyrich M (2020) Continual learning of fault prediction for turbofan engines using deep learning with elastic weight consolidation. In: 2020 25th IEEE international conference on emerging technologies and factory automation (ETFA) (Vol. 1, pp. 959-966). IEEE
12. Kumar KD (2021) Remaining useful life prediction of aircraft engines using hybrid model based on artificial intelligence techniques. In: 2021 IEEE International conference on prognostics and health management (ICPHM) (pp. 1-10). IEEE
13. Muneer A, Taib SM, Fati SM, Alhussian H (2021) Deep-learning based prognosis approach for remaining useful life prediction of turbofan engine. Symmetry 13(10):1861
14. Soualhi M, Nguyen K, Medjaher K, Zerhouni N (2022) Remaining useful life estimation of turbofan engines using adaptive fault detection learning. In: Annual conference of the PHM society (Vol. 14, No. 1)
15. Asif O, Haider SA, Naqvi SR, Zaki JF, Kwak KS, Islam SR (2022) A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset. IEEE Access 10:95425–95440
16. Sabry HM, El-Latif YMA (2022) Regression analysis for remaining useful life prediction of aircraft engines. In: International conference on advanced machine learning technologies and applications (pp. 157-168). Cham: Springer International Publishing
17. Karnik R, Pandya K (2023) Long-short-term memory model for remaining usable life estimation
18. Liu X, Xiong L, Zhang Y, Luo C (2023) Remaining useful life prediction for turbofan engine Using SAE-TCN model. Aerospace 10(8):715
19. Kadre SS, Paspula RG (2023) Improved RUL predictions of aero-engines by hyper-parameter optimization of LSTM neural network models
20. Wang H, Li D, Li D, Liu C, Yang X, Zhu G (2023) Remaining useful life prediction of aircraft turbofan engine based on random forest feature selection and multi-layer perceptron. Appl Sci 13(12):718

Sentiment Analysis of Marathi Texts Using Deep Learning Models

Deepak Mane, Sarthak Pithe, Hrishikesh Potnis, Soham Nimale, and Madhur Vaidya

Abstract The process of understanding the underlying subjective information and opinion in a given sentence is sentiment analysis. Accurate Sentiment analysis is possible with the use of Deep-Learning techniques. We have proposed three Deep-Learning based fine-tuned models for the three-class sentiment analysis task. It is complemented with an end-to-end pipeline that includes classical approaches like stemming, lemmatization in the data preprocessing, parameter tuning, and so forth for Marathi texts analysis. The proposed models are based on ANN, IndicBERT, and BiLSTM. The "L3CubeMahaSent" dataset was used for both training and testing of the proposed models. The IndicBERT based model achieved a superior F1-score of 85%, Precision of 85% and a recall score of 86% on the dataset. The overall weighted accuracy of this model was 81%. Additionally, all the proposed models achieved higher F1-scores and accuracy in predicting the negative sentiments compared to neutral and positive ones even in the class-balanced dataset.

Keywords Marathi NLP · Sentiment analysis · Deep learning · IndicBERT · BiLSTM

D. Mane · S. Pithe · H. Potnis · S. Nimale (✉) · M. Vaidya
Vishwakarma Institute of Technology, Pune, India
e-mail: soham.nimale22@vit.edu

D. Mane
e-mail: deepak.mane@vit.edu

S. Pithe
e-mail: sarthak.pithe22@vit.edu

H. Potnis
e-mail: hrishikesh.potnis221@vit.edu

M. Vaidya
e-mail: madhur.vaidya221@vit.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_57

1 Introduction

Sentiment analysis is a technique used to predict the emotion in a given sentence into various classes. It is used to analyze the emotion behind the sentences and to classify them either in two-class (positive or negative sentences), three-class (positive, neutral or negative sentences) or multi-class [1, 2]. Thus, it is commonly used for applications like customer feedback analysis, and social media monitoring [3]. The two main methods for performing sentiment analysis are—dictionary-based and machine learning-based. Sentiment analysis has been extensively researched in the context of the English language. However, there is less research done on this topic, compared to other languages. Marathi is a language spoken by many people. It is the third-most spoken language in India and 13th in the world in terms of total native speakers, and has received little attention in this regard. This paper presents a comparative study of various methods for three-class sentiment-analysis of sentences in Marathi language. Discussion about the implementations, advantages and disadvantages and how well they suit the environment of Marathi-language structure is also done. We used various deep learning methods to classify Marathi texts into positive, neutral or negative categories. We also propose an end-to-end pipeline for the Marathi Sentiment analysis that includes methods in Pre-processing, Model selection, Fine-tuning, getting Classification reports and revising the models.

The objectives of this paper are as follows:

- To perform accurate sentiment analysis using various classification models and preprocessing techniques on Marathi texts.
- To compare and evaluate the results obtained from each of them and with the existing models.
- To develop a preprocessing pipeline for the Marathi texts to obtain trainable features from raw dataset without much loss of sentiment.
- To understand the sentiments behind Marathi tweets using the models developed and its practical importance in various fields and in decision making.

The overview of this paper is as follows. The Sects. 1 and 2 includes the introduction and the related work respectively. It discusses the limitations of existing systems, and thus proves the novelty of our proposed solution. Section 3 discusses the general pipeline for sentiment analysis which is not language specific, along with the models proposed. Section 4 talks about the comparative analysis of results given by our proposed pipeline and previous results from papers cited in Sect. 2. Section 5 concludes the paper, followed by our future scope in Sect. 6.

2 Literature Review

In their paper 'L3CubeMahaSent', Kulkarni et al. [4] presented a new dataset of over 16,000 labelled Marathi tweets and applied various models, such as Convolutional Neural Network (CNN), Bidirectional long short-term memory

(BiLSTM), and Bidirectional Encoder Representations from Transformers (BERT), for three-class and two-class sentiment analysis.

Divate [5] modified the long short-term memory (LSTM) method for Marathi sentences and used a BiLSTM model on a Marathi news dataset, achieving an F1-score of 0.72. Ansari and Govilkar [6] proposed a novel approach for Marathi and Hindi sentences using the Support Vector Machine (SVM) method, but they relied on Romanized translations instead of Devanagari scripts.

Kale, Sunil D., et al. [7] discussed various Indian languages and the corresponding Sentiment Analysis work done in that area. About the Marathi language, they compared a few previously done research works with their methods and obtained their results.

One of the pioneers in this field was Balamurali et al. [8]. They suggested a cross-lingual approach and a 2-class classification method (positive and negative). Linked-WordNets were used by them, with both Hindi and Marathi texts in their dataset.

Deshmukh, Sujata, et al. [9] used WordNet in their approach and calculated the sentiment using a dictionary-based method. Keywords from the given sentence are detected and mapped to a previously assigned polarity. The average polarity for each sentence was then computed using the individual polarity of each keyword in a sentence. The sentiment of a sentence is decided based on this generated polarity.

In [10], Snehal and Swati used the Naïve Bayes (NB) probabilistic model and SVM for lexicon-based sentiment analysis. The classifier uses a positive and negative word set for reference. Their approach is optimized for the analysis of customer reviews and feedback.

M. K. Patil et al. [11] present various problems faced by researchers in Marathi sentiment analysis and discuss a semantic-corpus based model for the Marathi sentiment analysis and opinion-mining.

Catelli et al. [12] and Pota, Marco, et al. [13] suggested various methods for BERT-based analysis, which are useful for training the BERT machine learning model in the case of non-English languages. Pota, Marco, et al. [14] proposed a BERT-based pipeline for sentiment analysis for both English and Italian sentences.

3 Methodology

In the first section, the input dataset is pre-processed using the proposed pipeline as depicted in the Fig. 1. A detailed description of the same is provided in Sect. 3.2. The pre-processed texts were then used to generate word and sentence embeddings. Embeddings are needed to extract numerical features from the input Marathi texts. The following techniques were used to generate the embeddings for the proposed models: fastText for BiLSTM based model, IndicBERT for IndicBERT based model and Keras word embeddings for Artificial Neural Network (ANN) based model.

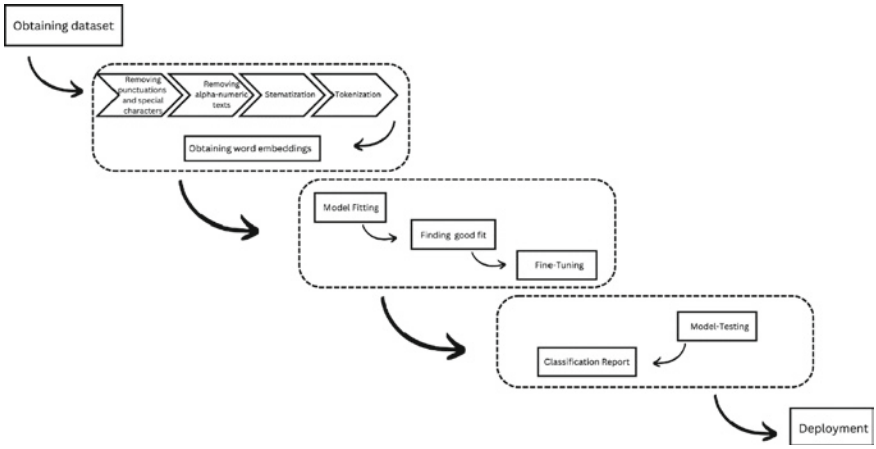


Fig. 1 Proposed Marathi sentiment analysis pipeline

The proposed models are then trained on the extracted word or sentence embeddings. Each model is associated with hidden fully-connected layers before the final output layer, to further enhance feature extraction. The IndicBERT based model was run for 100 epochs, while the ANN and BiLSTM based models were run for less than 10 epochs to obtain the optimal accuracy and avoid overfitting. The optimal learning rate for the IndicBERT based model was 0.0001%. Dropout rates of 10% were used for the fully connected layers. Validation dataset was used to validate the trained models and to avoid overfitting. After the validation, models were individually tested on the test dataset. Classification report was then generated from the results obtained.

3.1 Preprocessing

We have used the dataset from [4] for the sentiment analysis. It is a tweet-based, class-balanced dataset with over 16,000 tweets in the Marathi language, each corresponding to one of these three sentiment output classes: positive, negative, and neutral. The dataset was available in training, testing, and validating splits. Each split consists of a tweet column with the texts in the Marathi language and a sentiment column with the appropriate class identity for the corresponding sentiment. Some of the features of the dataset include:

- The training, testing, and validation datasets consist of 12,114, 2250, and 1500 rows. Each split has unique, class-balanced Marathi tweets in it.
- The maximum length of a tweet ignoring Uniform Resource Locators (URLs) and stop words is rounded off to 250 words. We used this length to pad the strings to ensure a consistent input shape while training.

- The number of unique words after removing the URLs and stop words is approximately equal to 60,000. This data is helpful during tokenization.
- The tweets also consisted of English words in the Marathi texts. English words were observed the most in the hashtags part of tweets.

The available dataset had not undergone any prior preprocessing; hence, we performed a series of steps on the training and testing dataset independently before the model training for the sentiment classification task [15, 16]. These steps include:

- We conducted a comprehensive analysis of the dataset before starting with the preprocessing to gain insights, some of which are presented at the start of this section.
- After analyzing the data, the first step in the preprocessing involved using regular expressions to eliminate punctuation marks and special characters. Examples include: "@", "#", and "?". This step also involved the removal of excessive spaces, tabs, or newline characters with the use of the same.
- We also removed the alpha-numeric English letters from the Marathi texts. Examples include: "100", "#politics", and "https". This step, together with the previous one ensures the removal of the URLs from the text.
- The lists of conjunctions, stop-words, pronouns, and numbers in the Marathi language were manually cumulated from various sources available on the internet by native speakers. We used these lists to remove parts of the sentences with less sentiment significance. Some examples from each list are provided in Table 1 along with their English translation.
- Because the Marathi language does not consist of lower-case and upper-case letters as in the English Language and all the English letters were removed from the texts in step three, the case conversion step in the English text preprocessing was skipped [17].
- Finally, we stem the words in the Marathi language into their root form. We used a dictionary-based suffix removal algorithm for the same.

3.2 Classification Models

All the described models make use of dense hidden layers for the classification purpose including the output layer integrated with the Softmax activation function. The mathematical equation representing the dense layer is as follows.

Table 1 Examples of words with less sentiment significance

S. No.	Example of numbers	Examples of pronouns	Examples of stop-words
1	११—Eleven	आम्ही—We	त्यामुळे—Therefore
2	१२—Twelve	तुझे—Yours	झाली—Done
3	१३—Thirteen	तो—He	होता—Was

$$\alpha_j = \beta_j + \sum_{i=0}^{n} (\alpha_i \times w_{ij})$$

"j" is the index of node under consideration that is present in the current dense layer. "i" is the index of node from the previous dense layer. "i" varies from the 0th index to the nth index, n being the number of nodes present in the previous layer. "ß" is the bias associated with the jth node of current layer, and w_{ij} is the weight of the edge from i^{th} node of previous layer to the jth node of the current layer.

ANN. We used tokenization to create a vocabulary set of 60,000 words. Each word in the sentence was replaced with the corresponding token number in the vocabulary set; Padding with zeros was done for consistent sentence length. Next, a Keras Word Embedding Layer was used to transform these numerical values into 32-dimensional vectors. These vectors were then passed through a Dense Layer with 256 units, using the Gaussian Error Linear Units (GELU) activation function for pattern learning. Finally, another Dense Layer with 3 units and a Softmax activation function is used to classify the text into one of 3 categories. The architecture of the ANN-based model is presented in the Fig. 2.

IndicBERT. For BERT based models like IndicBERT, the input should consist of tokens with special labels. This input format was processed through IndicBERT that assigned a set of 768 numbers to each sentence. These vectors were passed through a dense layer with the GELU activation and output 256 numbers. Lastly, another dense layer with Softmax activation classifies the sentence among 3 categories. The architecture of the IndicBERT-based model is presented in the Fig. 3.

FastText + BiLSTM. Similar to the ANN model we have used a tokenization that creates a vocabulary of 60,000 distinct words. We used different word embedding—fastText. It assigns 300 unique numerical values to each word and encodes it in a 300-dimensional space to capture the semantic relationships between the words. Then, we have passed these vectors through the BiLSTM neural network of 128 units that generates an output of 256-dimensional space (each BiLSTM cell has two sets of LSTM layers). Then through a dense layer of 64 units with

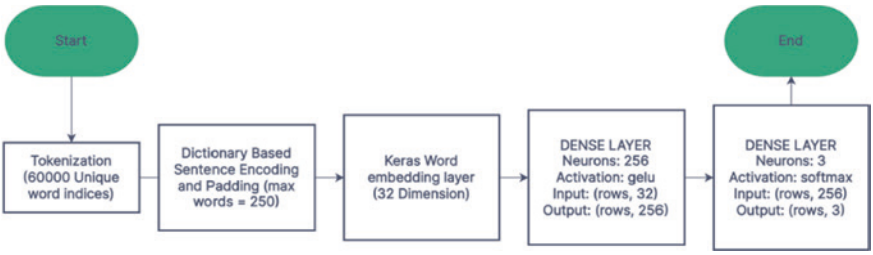


Fig. 2 Block diagram of the model using an ANN

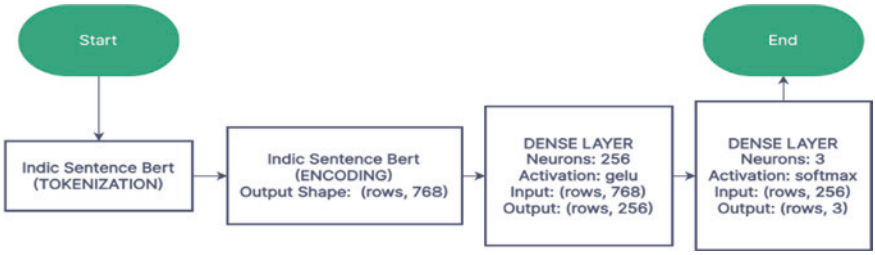


Fig. 3 Block diagram of the model using IndicBERT

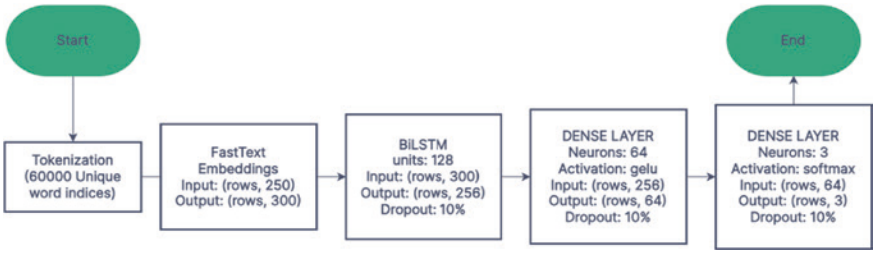


Fig. 4 Block diagram of the model using FastText and BiLSTM

the GELU activation function. The last 3-Node dense layer is the same as the IndicBERT based model. The architecture of the BiLSTM-based model is presented in the Fig. 4.

4 Experimental Results

We discuss the outcomes of each of the three models used. Given in Table 2 is the classification report for each of the models using our proposed pipeline. A graphical comparison of performance of each model is given in Fig. 5. It is evident that our second model—which combines IndicBERT with dense layers—performs better than the other two models in terms of accuracy, with an accuracy rate of 81%.

Figure 6a, b, c depict the confusion matrix for the Artificial Neural Network (ANN), IndicBERT. BiLSTM neural network with fastText word embeddings respectively. The IndicBERT model, which uses dense layers and IndicBert phrase encodings, achieved the maximum accuracy of 81%. Both ANN and IndicBERT based models showed higher accuracies in predicting negative and positive sentiments as compared to neutral sentiment. Finally, the BiLSTM based model that uses the aforementioned characteristics to predict sentiments, achieved an accuracy of 77%.

Table 2 Results obtained using all proposed pipelines

Models	Positive	Negative	Neutral	Weighted average (%)	Macro average (%)	Overall accuracy (%)
ANN	Precision:84%	Precision: 84%	Precision: 74%	81	81	80
	Recall: 75%	Recall: 86%	Recall: 81%	80	80	
	F1-score: 79%	F1-score: 85%	F1-score: 77%	80	80	
IndicBert + Dense Layers	Precision: 81%	Precision: 85%	Precision: 76%	81	81	
	Recall: 78%	Recall: 86%	Recall: 77%	81	81	81
	F1-score:80%	F1-score:85%	F1-score:77%	81	81	
FastText + BiLSTM	Precision: 81%	Precision: 79%	Precision: 74%	78	78	
	Recall: 73%	Recall: 84%	Recall: 75%	77	77	77
	F1-score: 77%	F1-score:81%	F1-score:75%	77	77	

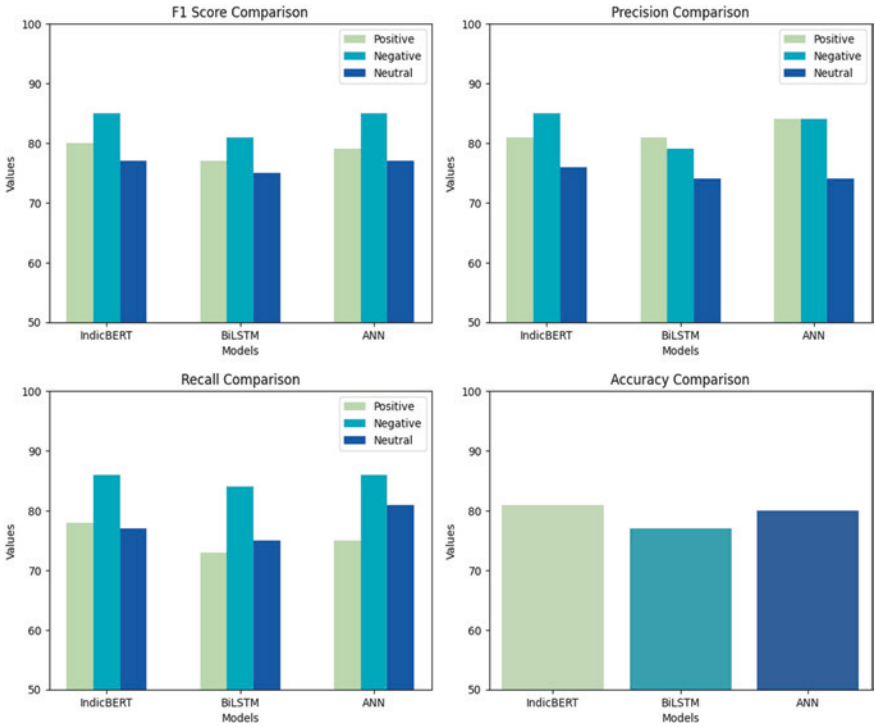


Fig. 5 Comparison of f1-score, precision, recall, and accuracy for IndicBERT, BILSTM, and ANN model for each of the three sentiment classes

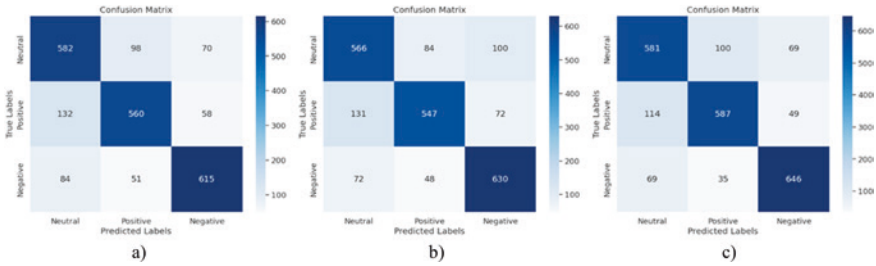


Fig. 6 Confusion matrices after model testing. **a** Confusion matrix for ANN model, **b** Confusion matrix for IndicBERT model, **c** Confusion matrix for BiLSTM model

4.1 Overall Performance

Figure 7 depicts the percentage of tweets in each of the predicted classes, along with the actual distribution in the test dataset. The red slice represents the negative sentiment, the green slice represents the positive sentiment and the yellow

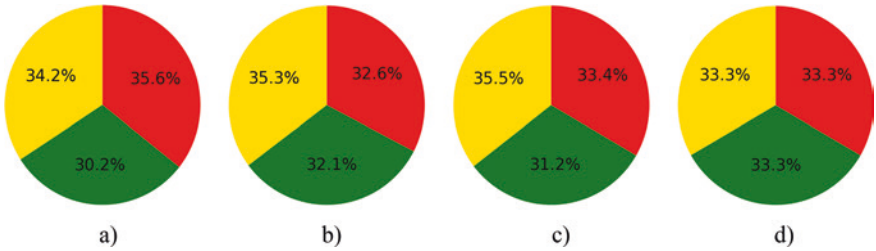


Fig. 7 Distribution of predicted and actual sentiments of the tweets in positive, negative and neutral classes. **a** BiLSTM based distribution, **b** IndicBERT based distribution, **c** Simple ANN based distribution, and **d** Actual distribution in the test dataset

slice represents the neutral sentiment. We can observe from the confusion matrices that the sentences associated with negative sentiments are predicted more accurately compared to the positive and the neutral ones. Examples of sentiments of sentences in each of the three classes along with the predictions made by our models is given in Table 3. A sentiment analysis model must be resilient to unknown datasets in order to accurately forecast the sentiment of a given input text, and our model meets this requirement.

4.2 Comparison with Existing Models

We provide a thorough comparison of models using our pipeline and other existing ones in Table 4. It is important to note that depending on the type of comparison parameter (example: accuracy/F1 score) and the model used, the results of some papers outperform our/others models and vice versa [18, 19]. For instance, L3CubeMahaSen has superior accuracy for IndicBERT that is around 84%, while MahaEmoSen has better accuracy for its Multilingual Representations for Indian Languages (MuRIL) model that is around 82%.

In [20] authors proposed a methodology that translates Devnagari Marathi texts into English using google translate to perform sentiment analysis on the English texts and compare the accuracy with the models trained for Devnagari texts. Our methodology outperforms the proposed models in the paper with accuracy of 65.41% and 64.16% for Random Forest and SVM based models respectively.

In [21], authors proposed the use of data augmentation of the dataset using back translation along with the extraction of visual features using visual encoders from the tweets in order to improve the overall accuracy of models. Despite this, our models present a higher accuracy than the proposed models in [21].

The authors in [4] introduced the 'L3CubeMahaSent' dataset for 3-class sentiment analysis of Marathi sentences. The following approaches/models were used—CNN, BiLSTM + GlobalMaxPool, ULMFiT (Based on LSTM) and BERT (mBERT and IndicBERT). Their results concluded that IndicBERT was best-performing model for 3-class sentiment analysis, which is also true in our case.

Table 3 Comparing few sentiment predictions with the actual sentiment

S. No.	Original tweet	English translation (Translated using google translate)	Sentiment predicted by our model	Actual senti-ment
1	सत्तेत येण्यापुर्वी तरुणांना ठोस रोजगार देण्याचं स्वप्न दाखवलंत, अजूनही नोकर भरतीची गाजरं दाखवतच आहात आणि आता मृहणताय सरकारी नोकरी हवी कशाला?	Before coming to power, you showed the dream of providing concrete employment to the youth, you are still showing the carrots of recruitment and now you are saying why do you need a government job?	Negative	Negative
2	Met Hon Union HM @AmitShah ji in Parliament in New Delhi केंद्रीय गृहमंत्री मा. अमति शाहजी यांची आज संसद भवन येथे सदिच्छा भेट घेतली. pic.twitter.com/92jX8Ox5vG	yMet Hon Union HM @AmitShah ji in Parliament in New Delhi Union Home Minister Hon. Satichha met Amit Shahji today at Sansad Bhavan. pic.twitter.com/92jX8Ox5vG	Neutral	Neutral
3	@INCMaharashtra च्या माजी अध्यक्ष स्व. प्रभाताई राव यांना जयंतीनमित्ति वनिम्र अभविादन! pic.twitter.com/Ztvgy1nvxt	Former president of @INCMaharashtra. Greetings to Prabhatai Rao on his birth anniversary! pic.twitter.com/Ztvgy1nvxt	Positive	Positive

Table 4 Comparison with existing models

Year and Ref	Method	Comparison parameters with values (%)
2023 [19]	MahaEmoSen—Indicbert	Accuracy: 75.67
2023 [19]	MahaEmoSen—BiLSTM	Accuracy: 77
2021 [4]	L3CubeMahaSent-Indicbert	F1-Score: 78.9
2021 [22]	IndicFed—mBert	Accuracy: 69.98 F1-Score: 69.97
–	**Our proposed pipeline**	**Accuracy: 81** **F1-Score: 81**

5 Future Work

The future scope for this research includes various directions. One of them being the development of an accurate stemmer for Marathi language. The primary objective of this Marathi stemmer would be to produce root words without losing the contextual meaning of the text. Unlike English, Marathi words undergo intricate

morphological changes based on various factors such as tense, gender, number, and case. Therefore, a stemmer must be accurate enough in recognizing and handling such linguistic details effectively.

MuRIL is a state-of-the-art multilingual model developed by Google AI that is specifically designed to handle the linguistic complexities of Indian language. Therefore, fine-tuning MuRIL for sentiment analysis could help achieve better results.

Expansion of the dataset using techniques like scraping and annotation [22] could also lead to the enhancement of the results obtained using the models. Scraping involves extraction of Marathi text data from online sources like social media platforms. Whereas, annotation requires human annotators to label Marathi text samples. Application of Institution-level Federated Learning for Marathi sentiment analysis could help expand the dataset without the need to make the datasets available publicly [23].

Another potential direction is the development of a large-scale stop-words dictionary for the removal of appropriate words from the Marathi texts that do not carry any sentimental meaning. Machine learning based models could be developed for the same purpose [24], making the removal of such stop-words easy and help sustain the sentimental and contextual meaning of the sentence.

6 Conclusion

This paper focuses on sentiment analysis of Marathi-language sentences. From experimentation, we conclude that the optimal model for performing sentiment analysis was IndicBERT (combined with dense hidden layers) with the highest accuracy of 81%. All the models have performed very well in predicting negative sentiments which is a key factor in sentiment analysis. The accuracy of predicting the neutral sentiment was generally the lowest. The development of an end-to-end pipeline for Marathi text sentiment analysis is the novelty of our work.

References

1. Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J 5(4):1093–1113
2. Prabowo R, Thelwall M (2009) Sentiment analysis: a combined approach. J Informet 3(2):143–157
3. Alessia D, Ferri F, Grifoni P, Guzzo T (2015) Approaches, tools and applications for sentiment analysis implementation. International J Comput Appl 125(3).
4. Kulkarni A, Mandhane M, Likhitkar M, Kshirsagar G, Joshi R (2021) L3cubemahasent: a marathi tweet-based sentiment analysis dataset. arXiv preprint arXiv:2103.11408.
5. Divate MS (2021) Sentiment analysis of Marathi news using LSTM. Int J Inf Technol 13(5):2069–2074

6. Ansari MA, Govilkar S. Sentiment analysis of mixed code for the transliterated hindi and marathi texts. Int J Nat Lang Comput (IJNLC) Vol 2018;7.
7. Kale SD, Prasad R, Potdar GP, Mahalle PN, Mane DT, Upadhye GD (2023) A comprehensive review of sentiment analysis on Indian regional languages: techniques, challenges, and trends. Int J Recent Innov Trends Comput Commun 11(9s):93–110
8. Balamurali AR, Joshi A, Bhattacharyya P (2012) Cross-lingual sentiment analysis for indian languages using linked wordnets. In: Proceedings of COLING 2012: Posters (pp 73–82).
9. Deshmukh S, Patil N, Rotiwar S, Nunes J (2017) Sentiment analysis of Marathi language. Int J Res Publ En-Gineering Technol IJRPET. 3:93–97
10. Snehal VP, Swati M (2017) Sentiment analysis in Marathi language. Int J Recent Innov Trends Comput Commun 5(8)
11. Patil MK, Chaudhari N, Pawar BV, Bhavsar R. (2021) Exploring various emotion-shades for Marathi sentiment analysis, 2021 Asian conference on innovation in technology (ASIANCON), PUNE, India, pp 1–5 https://doi.org/10.1109/ASIANCON51346.2021.9544961.
12. Catelli R, Pelosi S, Esposito M (2022) Lexicon-based versus bert-based sentiment analysis: a comparative study in Italian. Electronics 11(3):374
13. Pota M, Ventura M, Fujita H, Esposito M (2021) Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets. Expert Syst Appl 1(181):115119
14. Pota M, Ventura M, Catelli R, Esposito M (2020) An effective BERT-based pipeline for twitter sentiment analysis: a case study in Italian. Sensors 21(1):133
15. De Clercq O, Lefever E, Jacobs G, Carpels T, Hoste V (2017) Towards an integrated pipeline for aspect-based sentiment analysis in various domains. In: Proceedings of the 8th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp 136–142)
16. Agarwal A, Sabharwal J (2012) End-to-end sentiment analysis of witter data. In: Proceedings of the workshop on information extraction and entity analytics on social media data (pp 39–44)
17. Camacho-Collados J, Pilehvar MT (2017) On the role of text preprocessing in neural network architectures: an evaluation study on text categorization and sentiment analysis. arXiv preprint arXiv:1707.01780.
18. Naukarkar RA, Thakare AN (2021) A review on recognition of sentiment analysis of Marathi tweets using machine learning concept. Int J Scienti Resear Sci Eng Tech (IJSRSET). 8(2):190–3
19. Bhagat C, Mane D (2019) Survey on text categorization using sentiment analysis. Int J Sci Technol Res 8(8):1189–1195
20. Gavali, H (2020) Text sentiment analysis of Marathi language in English And Devanagari script. Diss. Dublin Business School
21. Chaudhari P, Nandeshwar P, Bansal S, Kumar N (2023) MahaEmoSen: towards emotion-aware multimodal marathi sentiment analysis. ACM Trans Asian Low-Resour LanGuage Inf Proc 22(9):1–24
22. Pingle A, Vyawahare A, Joshi I, Tangsali R, Joshi R (2023) L3Cube-MahaSent-MD: a multi-domain Marathi sentiment analysis dataset and transformer models. arXiv preprint arXiv:2306.13888
23. Mehta J, Gandhi D, Rathod N, Bagul S (2021) IndicFed: a federated approach for sentiment analysis in indic languages. In: Proceedings of the 18th international conference on natural language processing (ICON) Dec (pp 487–492).
24. Vidyavihar M (2017) Sentiment analysis in Marathi language. Int J Recent Innov Trends Comput Commun 5(8):21–25

Neuroscientific Study with Hybrid Models Validation for Deception Detection

Dharmendra Mishra, Rohit Mishra, Rahee Walambe, Ketan Kotecha, Priyanka Jain, N. K. Jain, Rahul Neiwal, and Manoj Jain

Abstract This research delves into the neuroscientific dimensions of deception detection, providing deeper insights into the cognitive and physiological processes associated with concealing, hiding, or deceiving. The study focuses on the use of the P300 ERP-based odd-ball paradigm to investigate cognitive aspects such as attention, memory, and decision-making in deception detection. The development and validation of a new hybrid paradigm, supported by a Machine-Learning model, demonstrate promising outcomes. The paper discusses the significance of Neuro-ERP methods, their cognitive processing implications, objective indicators, and cross-examination utility, along with ethical considerations and limitations. The study involved 10 subjects, employing AEP and VEP techniques for processing textual, auditory, and visual stimuli. The proposed hybrid paradigms, tested alongside traditional methods, are explored in detail, and its post-implementation effects are validated through canonical correlation analysis and a Machine-Learning model, providing evidence-backed insights. The results and implications are thoroughly discussed in the context of real-world applications.

Keywords P300 · Oddball paradigm · ERP · Design paradigm · Deception detection · EEG · Cognitive analysis · Nonlinear interdependence

D. Mishra (✉) · R. Mishra · P. Jain · N. K. Jain
C-DAC, Delhi, India
e-mail: darmendram@cdac.in

R. Walambe · K. Kotecha
SIT, Pune, India

R. Neiwal · M. Jain
MeitY, Delhi, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

A. Noor et al. (eds.), *Emerging Trends and Technologies on Intelligent Systems*, Lecture Notes in Networks and Systems 1073, https://doi.org/10.1007/978-981-97-5703-9_58

1 Introduction

Deception detection or concealed information test has been experimented in many studies on mock-thefts. An effort is needed to investigate various responsible components contributing in any ERP based paradigm. To identify a moral behavior (positive or negative) with ERPs, a comparative study and subsequently a design implementation is targeted. This work investigates the application of P300 detection in the context of truth and lie inference using the oddball paradigm. The P300 event-related potential (ERP) is a well-studied neurophysiological response associated with cognitive processes like attention and memory retrieval. The oddball paradigm, involving the presentation of rare and frequent stimuli, was extensively utilized to elicit the P300 response. Few paradigms are worked upon to investigate the contribution of textual information, auditory cue, auditory information and visual information in oddball paradigm-based stimuli. Backed up with this research outcome, a hybrid design based on Bootstrapped Amplitude Difference Analysis (BADA) [1], is proposed augmenting the impact of many more supportive parameters of EEG data analysis in time and frequency domain.

2 Related Work

For more than half a century, extensive research has been conducted on the Concealed Information Test (CIT) [2, 3], to detect relevant information and discriminate between knowledgeable (guilty) and unknowledgeable (innocent) individuals [4, 5]. Also known as Guilty Knowledge Test (GKT), it has been popular in the forensic domain due to its reliability on proper control questions in comparison to Control Questions Test (CQT). It was reported by [6–8] the P300 ERP evoked between 300 and 800 ms from the stimuli onset.

In a standard method [1], the comparison of the amplitude during non-target and target stimuli of P300 response recorded at the scalp sites along the midline of the head (Pz, Cz and Fz). Research on EEG-based lie detection has prominently employed the P3 oddball paradigm, as noted in [16]. Though several modifications of this technique have been suggested, as seen in [17, 18], and [19], all these methods face challenges or complexities due to potential conscious countermeasures. The primary comparison to determine guilt in these methods juxtaposes the P3 response during probe trials to that during irrelevant trials. While earlier ERP lie detection studies have utilized bootstrapping techniques, such as in [16], our approach adopts randomization. With adequate resampling, our method closely mirrors exhaustive permutation tests, which hold precise statistical accuracy as discussed in pages 15 and 16 of [20]. In the late 1980s, the use of ERPs in the Concealed Information Test (CIT) was explored, with methods utilizing the P300 waveform to identify crime-related recognition [12, 13]. Farwell and Donchin as noted in [14] technique, termed "brain fingerprinting," is based on the P300 component, evident between 300 and

800 ms post-stimulus. Using the oddball paradigm [15], where infrequent events induce a prominent P300, the approach posits that crime-linked stimuli will trigger a stronger P300 in informed (likely guilty) individuals.

For instance [9], explored frontal EEG asymmetry as a potential biomarker for anxiety. It is found that greater right frontal activity was associated with higher anxiety levels, providing valuable insights into the neural basis of anxiety during deception. Recently, machine learning techniques have gained prominence in EEG-based deception classification. Researchers have employed various AI algorithms such as Support Vector Machines (SVM), Random Forest (RF), and Convolutional Neural Networks (CNN) to differentiate between anxious and non-anxious states. Emotion-inducing stimuli can range from visual cues to audiovisual content. A study by [10] utilized affective pictures to provoke anxiety and investigated EEG responses. A recent study by [11] presented a real-time EEG-based anxiety monitoring system that showed promise in providing timely feedback to individuals experiencing anxiety in real-time.

3 Methodology & Implementation

3.1 Data Acquisition

Apparatus: The Emotiv EPOC X is a 14-channel EEG (electroencephalogram) device used for this research. It is set to sampling frequency at 250 Hz, allowing for the accurate capture of a wide range of neural activity. This high sampling rate ensures that the device detects even rapid changes in brainwave patterns with precision.

Data Collection: Each participant underwent three runs for each of the three paradigms. In each run, participants were presented with 100 trials of the corresponding paradigm, resulting in a total of 300 trials for each participant per paradigm for EEG data collection.

Participants: 10 right-handed healthy participants age ranged from 22–30 (M = 29.6, SD = 7.4) were recruited. All participants had normal or corrected vision and did not have any history of neurological diseases. All of them were naive to the experimental design and signed the informed consent.

3.2 Paradigm Description

Before proposing a new hybrid paradigm, this research work initially employs a comparative Paradigm study design to investigate the P300 event-related potential across three distinct oddball paradigms: Reading familiar-nonfamiliar name-based paradigm, listening spoken names in auditory and watching face-based paradigms.

The P300 component, a positive deflection in the event-related potential associated with cognitive processes, was examined across these paradigms to explore potential variations in its characteristics.

Procedure and tasks: The study included three types of oddball paradigms:

- **Paradigm I—Reading familiar-nonfamiliar name-based paradigm:** Participants were presented with written names in text on the computer screen in front of them. This task involved a selection of three names known to each subject. Out of these three names, one was randomly designated as the target stimulus for each trial. The rationale behind this design was to investigate the brain's response in reading the familiar and non-familiar names, simulating a real-world scenario where individuals encounter both familiar and unfamiliar information regularly.
- **Paradigm II—Listening the spoken names in auditory paradigm:** The stimuli design is similar to Paradigm I but with an auditory modality in auditory paradigm. Participants heard a series of names through auditory stimuli with one of the three names being spoken as the target stimulus in each run. The use of auditory stimuli added a dimension of sensory modality to the investigation, allowing us to explore whether the mode of stimulus presentation influenced neural responses.
- **Paradigm III—Watching familiar-nonfamiliar face-based paradigm:** In the familiar face-based paradigm, participants were presented with a sequence of images. These images included both familiar faces (specifically, their own faces) and non-familiar faces. Participants were instructed to focus on the screen on which faces were being presented. This paradigm aimed to explore how the brain processes self-referential familiar stimuli compared to unfamiliar faces, tapping into self-recognition processes. (Figure 1)

3.3 Signal Processing

Preprocessing Steps: The collected EEG data underwent several processing steps, as outlined below:

- Preprocessing: EEG data were imported and preprocessed using Python libraries, including MNE, Pandas, NumPy, and Matplotlib. Raw EEG data files for each run were loaded using the MNE library, resulting in a collection of raw objects.
- Filtering: A band-pass filter was applied to the combined raw EEG data, retaining frequencies within the range of 1 to 10 Hz. This filtering step aimed to isolate the relevant neural signals associated with the P300 component.
- Re-referencing: EEG data were re-referenced to the average reference. This step aimed to enhance the comparability of EEG signals across electrodes.
- Baseline correction: Epochs were baseline-corrected by subtracting the mean amplitude during the baseline period from each time point within the epochs.
- Data Analysis: The grand averaged ERP waveforms were calculated separately for target and non-target events. This involves averaging the individual epochs'

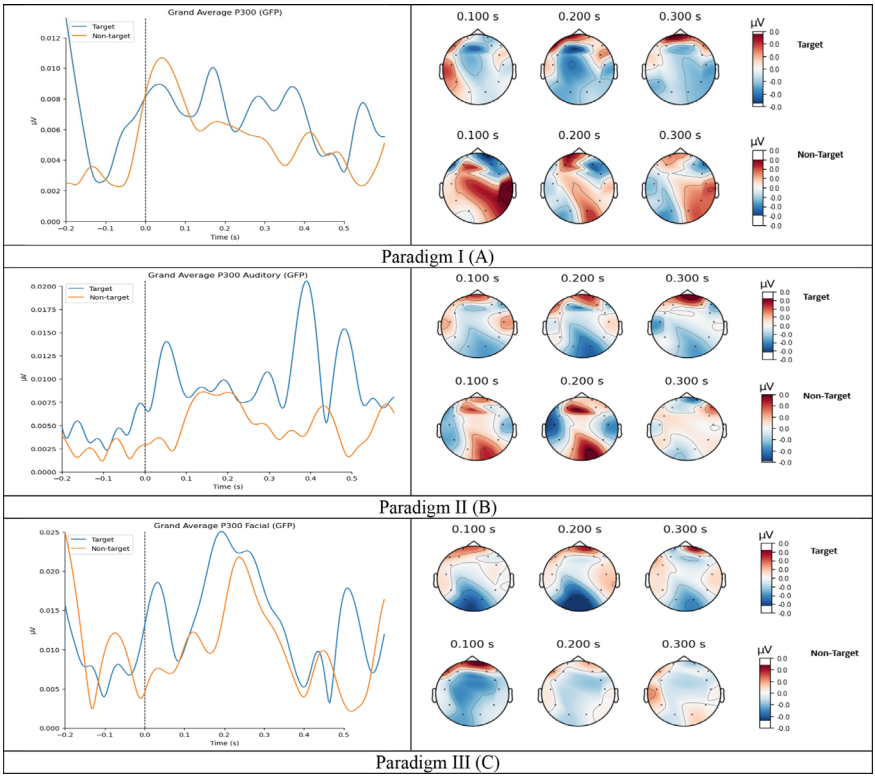


Fig. 1 Grand-averaged signal and topo-plots on three paradigms (I, II&III) respectively

signal and plotting the resulting averaged signals to obtain an average waveform, which provides insights into the brain's response to specific events (Figs. 2 and 3).

The EEG shape analysis performed with the assumptions that (i) each subject produces P300 signals with consistent waveform patterns that can be represented using template curves and (ii) these template curves from a subject exhibit greater similarity to P300-related curves than to curves generated by EEG background activity [21]. Many techniques that align with these principles fall into categories such as Cross Correlation Alignment (e.g., Woody's [22] and Maximum Likelihood (ML) [23] methods), Dynamic Time Warping (DTW) alignment [24, 25], and linear methods like coherent averaging [26]. Despite being the most debated, coherent (or grand) averaging is the fastest and most commonly employed averaging method due to a compelling argument: a P300 can be viewed as a well-defined component, and its peak alignment "tends to be linear even in the presence of nonlinear distortion" [27]. Consequently, it is customary to repeat the stimulation process to enhance the signal-to-noise ratio (SNR) by coherently averaging several segments of filtered EEG signals generated after stimulation (referred to as trials). Here we have taken 3 runs with 100 trials each i.e. 300 trials, the specific number of simulations required may

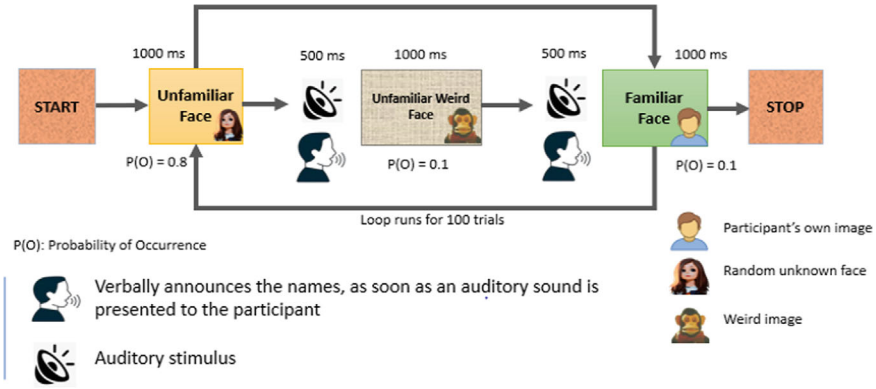


Fig. 2 Watching familiar-weird faces with an auditory cue paradigm. (Paradigm IV)

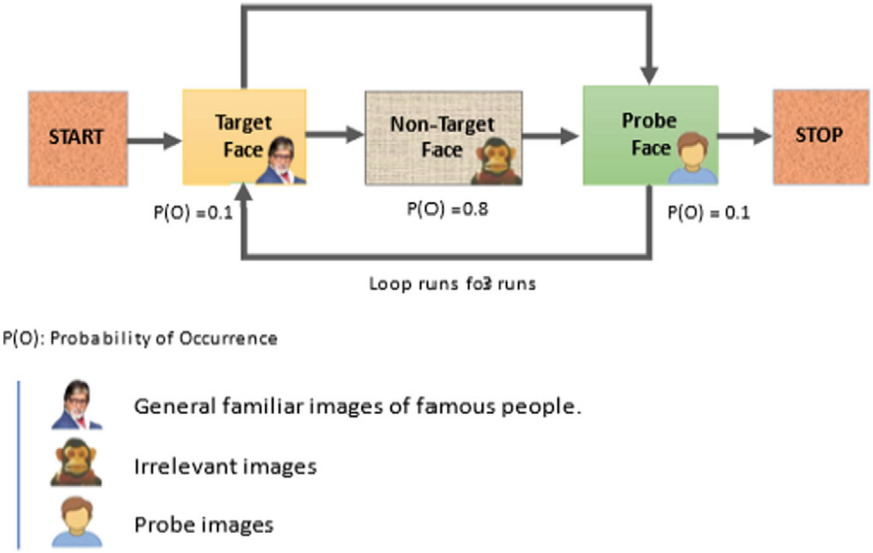


Fig. 3 Watching familiar-weird faces CCA-based paradigm. (Paradigm V)

vary among subjects, as discussed in [28]. Figure 4 presents the grand average plot and topolots of epochs of EEG filtered at frequency 1 to 10 pertaining to all three experimented paradigms.

The EEG signal was preprocessed with band-pass filtration from 1 to 10 Hz, average re-referencing and baseline correction. To steer the experiment in the direction of lie detection, it is proposed that familiar non-familiar name-based paradigms be used to elicit the P300. In the familiar-nonfamiliar name-based paradigm, three distinct names were asked (of people familiar to him) from the subject on each trial (total 3 trials) and one of which was randomly selected to be used as an oddball. To

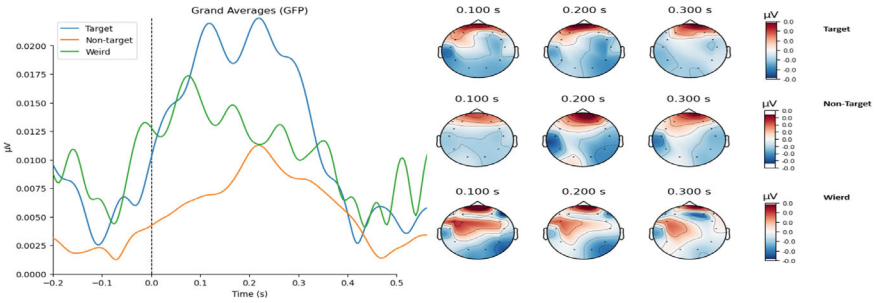


Fig. 4 Grand-averaged signal plot and topo-plots of the EEG signal on proposed Paradigm IV

assess the difference of ERP averaged waveforms both when the subject read the name (Paradigm I) and when the subject was dictated the names (Paradigm II), we conducted another experiment taking 10 subjects with 3 runs consisting of 100 trials each. (Fig. 1) Upon analysis of the plots of averaged epochs of when the subject heard the names and when the subject read the name, it was observed that there is significant difference. For our study we plotted and visualized the distribution of EEG activity at specific time points of 100, 200 and 300 ms after the onset of stimuli.

3.4 Proposed Paradigm IV and V

Building upon insights gained from previous paradigms, we introduce two novel approaches to further our comprehension of attentional focus in the brain and its ability to inhibit distractions from irrelevant stimuli. Proposed Paradigm IV maintains our exploration of attention dynamics by incorporating unusual or unsettling images amidst familiar faces. This paradigm aims to unravel the brain's reactions to both the familiar (participants' own faces) and the unfamiliar (peculiar images), considering the impact of aural cues on this cognitive dynamic. An auditory cue, signaling the imminent display of either a familiar target (participants' own face) or an unconventional image (a non-familiar stimulus), precedes the image presentation. This innovative paradigm allows us to scrutinize how the brain responds to unexpected deviations from familiar stimuli and examines the influence of auditory cues on these responses.

Proposed Paradigm V, termed the CCA-based paradigm, introduces a unique methodology post-preprocessing. Canonical components between target, non-target, and probe grand-averaged epochs are computed. Subsequently, Pearson correlation coefficients between target and irrelevant as well as between target and probe epochs are determined. Subjects are deemed 'guilty' if the correlation between target and probe epochs exceeds that between target and irrelevant epochs. This approach provides an additional avenue for investigating cognitive processes related to attention and deception.

ML Classification with Support Vector Classification (SVC): In P300 waveform research, the linear support vector machine (SVM) is considered one of the more accurate classifiers [29, 30]. The primary concept behind a linear SVM is to locate a hyperplane that effectively separates two classes, ensuring that the distance between the hyperplane and the nearest points from both classes is maximized. In simpler terms, the goal is to maximize the gap or margin between these two classes [31]. However, it's important to note that in real-world scenarios, the two classes are not always completely separable by a straight line. To address this, the concept of the linear SVM has been extended to situations where data points are allowed to exist within the margin (and sometimes even on the wrong side of the decision boundary) by introducing a regularization term.

- Canonical Correlation analysis (CCA) [32] is applied in online-preprocessing to calculate canonical components between target, non-target, and probe grand-averaged epochs. This statistical technique explores the relationships between two sets of variables by identifying linear combinations that maximize the correlation between the sets. In the context of Paradigm V, CCA is utilized to unveil the connections between the brain's responses to familiar (target), unfamiliar (non-target), and potentially deceptive (probe) stimuli.

- Calculation of Canonical Components: CCA enables the extraction of canonical components that represent the shared variability between the different sets of epochs. This provides valuable insights into how the brain processes and integrates information during the detection of deceptive stimuli.
- The objective in CCA is to maximize the correlation between linear combinations of variables X and Y

$$max_{a,b} Corr\left(a^T X, b^T Y\right)$$

- Pearson Correlation Coefficients: Following the determination of canonical components, Pearson correlation coefficients are computed between target and irrelevant as well as between target and probe epochs. This step serves as a critical measure to identify patterns and correlations indicative of deceptive cognitive processes. The Pearson correlation coefficient between two variables X and Y is given by:

$$Corr(X, Y) = \frac{Corr(X, Y)}{\sigma_x \sigma_y}$$

In our offline analysis, we employed a method based on the linear least squares SVM [31] to solve the minimization problem. This problem aims to minimize the expression: $\left(\frac{1}{2}\right) w^T w + \gamma \sum_i^n e_i^2$, while simultaneously satisfying the conditions:
$y_i(w^T f_i + b) = 1 - e_i, I = 1, ..., n$

Here, fi represents the training data points in the feature space, and yi denotes the associated output, with a value of + 1 indicating responses to the target stimulus

and -1 indicating responses to the nontarget stimulus. The regularization parameter is estimated through a line search based on cross-validation results. Key features and concepts of SVC include:

- Hyperplane: In a binary classification (where data points are divided into two classes), SVC aims to find a hyperplane that best separates the data points of one class from the other. This hyperplane is positioned to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class.
- Support Vectors: These are the data points closest to the hyperplane and have the smallest margin. They play a critical role in determining the position and orientation of the hyperplane.
- Kernel: SVC handles both linearly separable and non-linearly separable data by transforming the input data into a higher-dimensional space using a kernel function. This transformation enables the algorithm to find a hyperplane that can effectively separate non-linearly distributed data.
- C Parameter: The C parameter in SVC is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error. A smaller C value encourages a wider margin but may lead to misclassification, while a larger C value aims for correct classification but may result in a narrower margin.

For P300 component analysis in EEG data, Vanilla SVC of sklearn is used with default parameters. The accuracy rates obtained for each paradigm, as presented in Table 1, indicate how effectively the SVC model distinguishes P300 waveforms in the EEG data collected during different experimental paradigms. The analysis aimed to determine whether there were significant differences in the characteristics of the P300 component across the four oddball paradigms. The latency of the P300 peak and activation regions were of particular interest, as these measures can provide insights into the cognitive processing associated with familiar and non-familiar stimuli.

Also, in our machine learning classification (Table 1), the Support Vector Classifier (SVC) model demonstrated a greater ability to detect the P300 waveform with higher accuracy in the Paradigm IV paradigm compared to the preceding paradigms. This observation suggests that the Paradigm IV, which combines auditory cues with familiar and weird face stimuli, holds promise as an effective method for eliciting

Table 1 P300 latency and ML detection accuracy of ERP waveforms in respective paradigms

Paradigm	P300 Latency (ms)	Accuracy (%)
Paradigm I: Reading familiar-nonfamiliar name-based	206	79
Paradigm II: Listening the spoken names in auditory	390	78
Paradigm III: Watching familiar-nonfamiliar faces	207	83
Paradigm IV: Watching familiar-weird faces with an auditory cue	265	92
Paradigm IV: Watching familiar-weird faces CCA-based	100	55

and detecting the P300 component. Paradigm IV explores the nuances of familiarity, the influence of auditory cues, and the brain's capacity to distinguish self from non-self in the midst of unexpected stimuli, building a bootstrap paradigm based on the groundwork set by Paradigm I, Paradigm II, and Paradigm III.

4 Results and Discussion

The analysis aimed to determine whether there were significant differences in the characteristics of the P300 component across the three oddball paradigms (Fig. 5).

The latency and amplitude of the P300 peak were of particular interest, as these measures can provide insights into the cognitive processing associated with familiar and non-familiar stimuli. Here, we present and discuss the findings derived from all these plots and, which helped us better understand the distinct responses to various stimuli and paradigms.

- Paradigm I: Grand averaged and topo-plots for the familiar-non-familiar name-based paradigm revealed a pronounced positive peak at frontal electrode sites. This peak, corresponding to the P300 component, indicated enhanced neural activity during the processing of familiar names. The distribution of the brain response provided evidence of the brain's sensitivity to familiar stimuli, with a clear focus on frontal cortex regions. This finding aligns with existing literature on P300 responses in the context of memory and familiarity.
- Paradigm II: Similar to the topoplot of familiar non-familiar name-based paradigm, topographic plot displayed a similar positive activation at frontal region electrodes, albeit with a slightly more concentrated distribution in the frontal region compared to the other name and face based paradigm (Fig. 1). Interestingly, the latency of this P300 component was notably longer in the auditory paradigm, suggesting that auditory processing of familiar names may involve distinct neural mechanisms or additional processing stages.

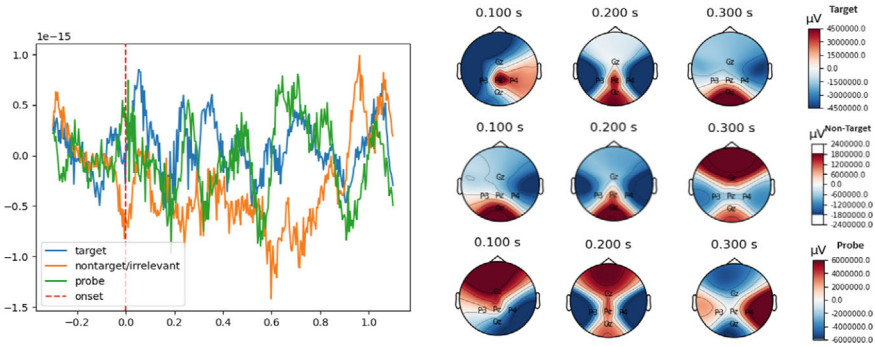


Fig. 5 Grand-averaged signal plot and topo-plots of the EEG signal on proposed Paradigm V

- Paradigm III: The topographic plots for the familiar face-based paradigm demonstrated a unique neural response pattern. Instead of a prominent P300 peak, we observed a more diffuse and bilateral distribution of neural activity across frontal and temporal regions. This finding indicates that the processing of familiar faces may engage a broader network of brain regions compared to the verbal stimuli used in the other paradigms.
- Paradigm IV: Due to the inclusion of auditory cues, topographic plots revealed noteworthy differences in the distribution and greater brain responses compared to the other paradigms. Our analysis revealed that the P300 peak latency in the Paradigm IV was notably longer compared to the other paradigms except the Paradigm II which also included auditory stimuli. This duration was closer to that of the P300 response which indicates a more robust and sustained cognitive processing of the stimuli, which could have important implications for applications such as deception detection.
- Paradigm V: This is another novel paradigm similar to paradigm IV which utilizes correlation between the canonical components of grand averaged epochs of target, nontarget and probe. Our inference results with this paradigm on 10 subjects for 3 runs showed around 55% accuracy.

While significant progress has been made in deception classification using EEG signals, challenges remain. These include the need for larger and more diverse datasets, improved feature selection methods, and the integration of multimodal data sources. Future research is to be focused on enhancing the interpretability of machine learning models and ensuring their applicability.

5 Conclusion

The current research aimed to systematically explore this potential by contrasting auditory and visual stimuli within the framework of the same stimulus sequence pattern, administered under both passive and active task circumstances. The methodology outlined in above sections facilitated the investigation of the P300 event-related potential across different oddball paradigms. By collecting and preprocessing EEG data, generating ERPs, and conducting grand average and peak detection analyses, this study aimed to contribute to the understanding of how the P300 component varies across distinct cognitive tasks involving familiar and non-familiar stimuli. The subsequent studies will present and discuss the findings derived from the application of this methodology in the context of lie detection. The use of EEG signals for deception classification is a promising area of research with potential implications for improving the diagnosis of deception. Recent studies have shown the effectiveness of machine learning techniques and the exploration of EEG biomarkers in this context.

References

1. Rosenfeld JP, Soskins M, Bosh G, Ryan AH (2004) Simple, effective countermeasures to P300-based tests of detection of concealed information. Psychophysiology 41(2):205–219. https://doi.org/10.1111/j.1469-8986.2004.00158.x
2. Lykken DT (1959) The GSR in the detection of guilt. J Appl Psychol 43(6):385–388. https://doi.org/10.1037/h0046060
3. Lykken DT (1960) The validity of the guilty knowledge technique: the effects of faking. J Appl Psychol 44:258–262
4. Ben-Shakhar G, Furedy JJ (1990) Theories and applications in the detection of deception: a psychophysiological and international perspective. Springer-Verlag, New York
5. Elaad E (1998) The challenge of the concealed knowledge polygraph test. Expert Evid 6:161–187
6. Winograd MR, Rosenfeld JP (2011) Mock crime application of the complex trial protocol (CTP) P300-based concealed information test. Psychophysiology 48:155–161
7. Cutmore TRH, Djakovic T, Kebbell MR, Shum DHK (2009) An object cue is more effective than a word in ERP-based detection of deception. Int J Psychophysiol 71:185–192
8. Jung EK, Kang KY, Kim YY (2013) Frontoparietal activity during deceptive responses in the P300-based guilty knowledge test: an sLORETA study. Neuroimage 78:305–315
9. Knyazev GG (2004) EEG correlates of self-referential processing. Int J Psychophysiol 51(2):91–97
10. Schell AM, Dawson ME, Marinkovic K, Böhmelt AH (2018) Brain–body fear circuit: attention–orienting vs. threat–assessment. Brain Cogn 127:1–12
11. Zhang J, Han J, Li Z, Zhang T (2021) A real-time EEG-based anxiety monitoring system. IEEE Access 9:7313–7320
12. Farwell LA, Donchin E (1986) The '"brain detector"': P300 in the detection of deception. Psychophysiology 23:434
13. Rosenfeld JP, Cantwell B, Nasman VT, Wojdac V, Ivanov S, Mazzeri L (1988) A modified, event-related potential-based guilty knowledge test. Int J Neurosci 42:157–161
14. Farwell LA, Donchin E (1991) The truth will out: interrogative polygraphy ('"lie detection"') with event-related brain potentials. Psychophysiology 28:531–547
15. Donchin E (1981) Surprise!… surprise? Psychophysiology 18:493–513
16. Farwell LA, Donchin E (1991) The truth will out: interrogative polygraphy ("Lie Detection") with event-related brain potentials. Psychophysiology 28:531–547
17. Rosenfeld J, Shue E, Singer E (2007) Single versus multiple probe blocks of P300-based concealed information tests for self-referring versus incidentally obtained information. Biol Psychol 74:396–404
18. Farwell LA, Smith SS (2001) Using brain MERMER testing to detect knowledge despite efforts to conceal. J Forensic Sci 46:135
19. Labkovsky E, Rosenfeld JP (2012) The P300-based, complex trial protocol for concealed information detection resists any number of sequential countermeasures against up to five irrelevant stimuli. Appl Psychophysiol Biofeedback 37:1–10
20. Manly BFJ (2007) Randomization, bootstrap and Monte Carlo methods in biology: Chapman & Hall.
21. Liang N, Bougrain L (2008) Averaging techniques for single-trial analysis of oddball event-related potentials, In: Proceedings of the 4th international brain-computer interface workshop, Graz, Austria.
22. Woody CD (1967) Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals. Med Biol Eng 5(6):539–554
23. Jaskowski P, Verleger R (1999) Amplitudes and latencies of single-trial ERP's estimated by a maximum-likelihood method. IEEE Trans Biomed Eng 46(8):987–993
24. Picton T, Hunt M, Mowrey R, Rodriguez R, Maru J (1988) Evaluation of brain-stem auditory evoked potentials using dynamic time warping. Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section 71(3):212–225

25. Casarotto S, Bianchi AM, Cerutti S, Chiarenza GA (2005) Dynamic time warping in the analysis of event-related potentials. IEEE Eng Med Biol Mag 24(1):68–77
26. Wang K, Begleiter H, Porjesz B (2001) Warp-averaging event-related potentials. Clin Neurophysiol 112(10):1917–1924
27. Polich J, Kok A (1995) Cognitive and biological determinants of P300: an integrative review. Biol Psychol 41(2):103–146
28. Krusienski DJ, Sellers EW, Cabestaing F et al (2006) A comparison of classification techniques for the P300 Speller. J Neural Eng 3(4):299–305
29. Thulasidas M, Guan C, Wu J (2006) Robust classification of EEG signal for brain-computer interface. IEEE Trans Neural Syst Rehabil Eng 14(1):24–29
30. Vapnik V (1995) The nature of statistical learning theory. Springer, NY, USA
31. Suykens J, Van Gestel T, De Brabanter J, De Moor B, Vanderwalle J (2002) Least Square Support Vector Machines. Singapore: World Scientific
32. "Sklearn.cross_decomposition.CCA."Scikit-Learn,scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.CCA.html. Accessed 29 Jan. 2024